NASA TECHNICAL NOTE

NASA TN D-4525

C. 1

NASA TN D-4525

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

by Theodore Katsanis

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1968

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

By Theodore Katsanis

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

CONTENTS

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

by Theodore Katsanis

Lewis Research Center

SUMMARY

A FORTRAN IV computer program was written that gives the solution of the two-dimensional, subsonic, compressible (or incompressible), nonviscous flow problem for a rotating or stationary circular cascade of blades on a blade-to-blade surface of revolution. The flow may be axial, radial, or mixed. There may be a change in stream channel thickness in the through-flow direction.

The computer program requires the basic cascade geometry, the meridional stream channel coordinates, fluid total conditions, weight flow, and inlet and outlet flow angles. The output includes streamline coordinates, velocity magnitude and direction throughout the passage, and the blade surface velocities.

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

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

INTRODUCTION

In the design of blade rows for turbines or compressors, it is desirable to obtain the velocity distribution through the passage and particularly over the blade surfaces. The trend to highly loaded blading results in more widely spaced blades with less of the passage being within a guided channel between the blades. The velocity distribution is readily obtained within the guided channel by stream filament techniques.

For the unguided portion of the passage, finite-difference methods have been used. Stanitz (refs. 1 and 2) obtained finite-difference solutions for compressible flow through turbomachines, without the use of a computer. Kramer (refs. 3 and 4) has obtained finite-difference solutions for incompressible flow through centrifugal pumps, using a computer for the solution of the finite-difference equations for the stream function. More recently a program has been written to perform in addition to the solution of the finite-difference equations, the calculation of the coefficients, and the differentiation of the stream function to obtain the velocities for incompressible flow through an axial blade row (ref. 5).

To extend this technique, a computer program has been written to obtain a numerical solution for ideal, subsonic, compressible (or incompressible) flow for either an axial, radial, or mixed flow turbomachine blade row which may be fixed or rotating. The stream function used here is a function of meridional streamline distance and angular coordinate, whereas the previously mentioned references all used either radius or axial coordinates instead of the meridional streamline distance. Also, the finite-difference equation has been given in a simpler form. The input required consists of the basic geometry coordinates, fluid total conditions, weight flow, and inlet and outlet flow angles. The output includes velocity magnitude and direction through the passage, blade surface velocities, and streamline coordinates.

This report includes the FORTRAN IV computer program that was developed, with an explanation of the equations involved and the method of solution. A radial gas turbine rotor and an axial turbine stator have been analyzed to illustrate the use of the program, and these results are compared with results obtained by other methods.

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

SYMBOLS

A	coefficient matrix, eq. (A7)
$\left.\begin{matrix} a_0, a_1, a_2, a_3 \\ a_4, a_{12}, a_{34} \end{matrix}\right\}$	coefficients in eq. (A2)
a_{ij}	typical element of matrix A

b	normal stream channel thickness, m
b_{12}, b_{34}	quantities in eq. (A2)
c_p	specific heat at constant pressure, J/(kg)($^{\circ}$K)
h_1, h_2, h_3, h_4	spacing between adjacent points, eq. (A1), see fig. 18
$\underline{k}$	constant vector, $\begin{pmatrix} k_1 \\ \cdot \\ \cdot \\ \cdot \\ k_n \end{pmatrix}$, eq. (A7)
L_1	coefficient matrix of eq. (A8) when $\Omega = 1$
m	meridional streamline distance, see fig. 2
n	number of unknown mesh points
R	gas constant, J/(kg)($^{\circ}$K)
r	radius from axis of rotation, m
s	angular blade spacing, rad
T	temperature, $^{\circ}$K
u	stream function
$\underline{u}$	discrete approximation to stream function at n mesh points, $\begin{pmatrix} u_1 \\ \cdot \\ \cdot \\ \cdot \\ u_n \end{pmatrix}$
$\underline{u}^m$	m^{th} iterate of $\underline{u}$, $\begin{pmatrix} u^m \\ \cdot \\ \cdot \\ \cdot \\ u^{m+1} \end{pmatrix}$
V	absolute fluid velocity, m/sec
W	fluid velocity relative to blade, m/sec
w	mass flow per blade flowing through stream sheet, kg/sec
z	axial coordinate, m
α	angle between meridional streamline and axis, rad, see fig. 1
β	angle between relative velocity vector and meridional plane, rad, see fig. 1

γ	specific heat ratio
η	outer normal to region
θ	relative angular coordinate, rad, see fig. 1
λ	prerotation $(rV_\theta)_{in}$, m^2/sec
ρ	density, kg/m^3
$\rho(\ )$	spectral radius of matrix
Ω	overrelaxation factor, eq. (A8)
ω	rotational speed, rad/sec

Subscripts:

cr	critical velocity
giv	given
i	dummy variable
in	inlet or upstream
j	dummy variable
l	lower surface of blade
m	component in direction of meridional streamline
out	outlet or downstream
r	radial component
u	upper surface of blade
z	axial component
θ	tangential component
0, 1, 2, 3, 4	quantities at these locations in finite difference expression, fig. 18

Superscripts:

T	transpose of vector or matrix
'	absolute stagnation condition
''	relative stagnation condition

MATHEMATICAL ANALYSIS

It is desired to determine the flow distribution through a stationary or rotating cascade of blades on a blade-to-blade surface. The following simplifying assumptions are used in deriving the equations and in obtaining a solution:

(1) The flow is steady relative to the blade.

(2) The fluid is a perfect gas or is incompressible.

(3) The fluid is nonviscous.

(4) There is no loss of energy.

(5) The flow is absolutely irrotational.

(6) The blade-to-blade surface is a surface of revolution.

(7) The velocity component normal to the blade-to-blade surface is zero.

(8) The stagnation temperature is uniform across the inlet.

(9) The velocity magnitude and direction is uniform across the upstream and across the downstream boundaries.

(10) The relative velocity is subsonic everywhere.

The flow may be axial, radial, or mixed and there may be a variation in the stream channel thickness b in the through-flow direction.

The coordinate system is shown in figure 1. Since the variables r and z are not

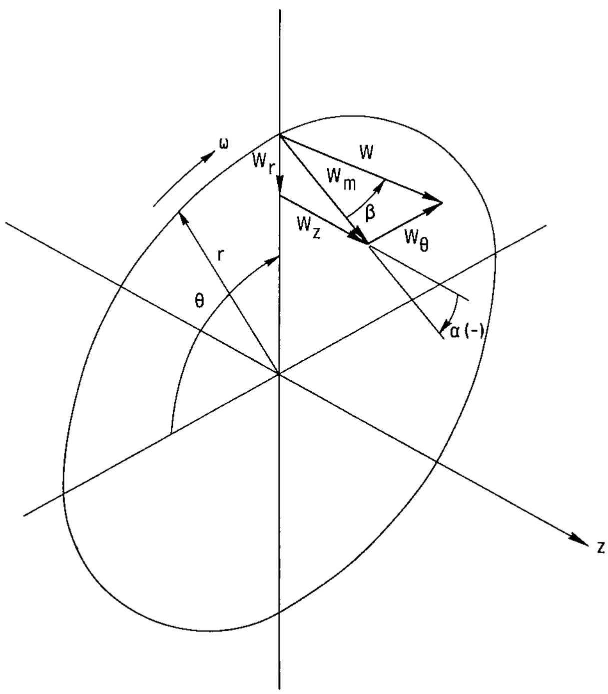


Figure 1. - Coordinate system and velocity components.

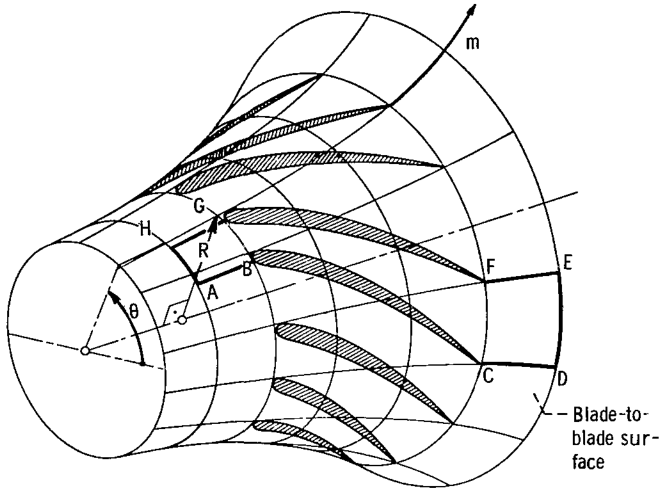


Figure 2. - Blade-to-blade surface of revolution.

independent on the stream surface, one variable can be eliminated. It is better, however, to use the meridional streamline distance m as an independent variable (see fig. 2). Then, m and θ are the two basic independent variables. A stream channel is defined by specifying a stream channel thickness b as shown in figure 3.

For the mathematical formulation of the problem the stream function is used. The stream function u satisfies

$$\frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} + \frac{\partial^2 u}{\partial m^2} - \frac{1}{r^2}\frac{1}{\rho}\frac{\partial \rho}{\partial \theta}\frac{\partial u}{\partial \theta} + \left[\frac{\sin \alpha}{r} - \frac{1}{b\rho}\frac{\partial(b\rho)}{\partial m}\right]\frac{\partial u}{\partial m} = \frac{2b\rho\omega}{w}\sin \alpha \qquad (1)$$

This may be obtained from equation 12(9) of reference 6 by letting $u = -\psi/w$, where ψ is the stream function as defined in reference 6. The stream function u has the value 0 on one blade and 1 on the other. Also, the derivatives of the stream function satisfy

$$\frac{\partial u}{\partial m} = -\frac{b\rho}{w} W_\theta \qquad (2)$$

$$\frac{\partial u}{\partial \theta} = \frac{b\rho r}{w} W_m \qquad (3)$$

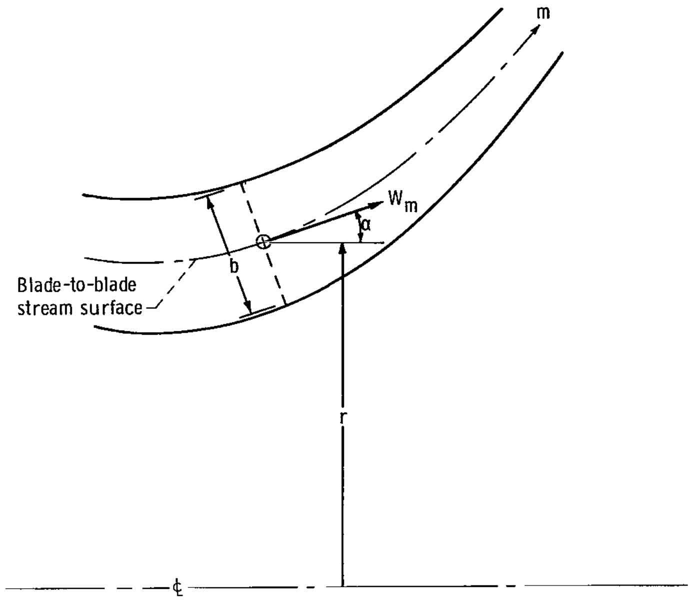


Figure 3. - Flow in a mixed flow stream channel.

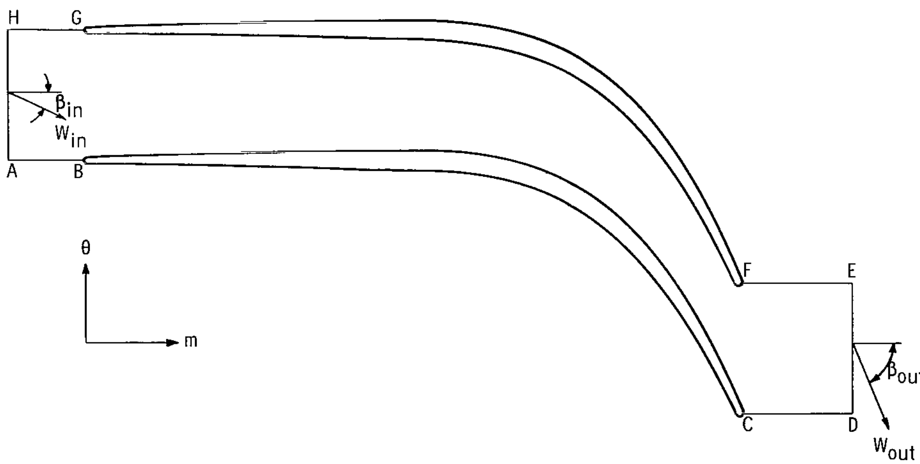


Figure 4. - Finite flow region for a radial turbine.

For the solution of equation (1), a finite region is considered (as indicated in fig. 4) with the condition that the flow along AB is the same as along HG, and the flow along CD is the same as along FE. Also, it is assumed that AH is sufficiently far upstream so that the flow is uniform along this boundary, and that the flow angle β_{in} is known. Similarly, it is assumed that the flow is uniform along DE, and that the flow angle β_{out} is known. For an actual blade row, β_{out} may usually be determined by means of experimentally determined rules. Specifying β_{out} along DE is mathematically equivalent to specifying the location of the stagnation point on the trailing edge of the blade.

Since equation (1) is elliptic for subsonic flow, boundary conditions for the entire boundary ABCDEFGH are required. Along BC, u = 0; along FG. u = 1. Along AB, GH, CD, and EF, a periodic condition exists; that is, the value of u along HG and FE is exactly 1 greater than it is along AB and CD. Along AH and DE, $\partial u/\partial \eta$ is known, where η is in the direction of the outer normal. From equations (2) and (3), since $W_\theta/W_m = \tan\beta$,

$$\frac{\partial u}{\partial m} = -\frac{\partial u}{r\partial\theta} \tan \beta \tag{4}$$

Along AH and DE,

$$\frac{\partial u}{\partial \theta} = \frac{u(H) - u(A)}{s} = \frac{1}{s}$$

so that

$$\left(\frac{\partial u}{\partial \eta}\right)_{in} = \frac{\tan \beta_{in}}{sr_{in}} \quad \text{along AH} \tag{5}$$

$$\left(\frac{\partial u}{\partial \eta}\right)_{out} = -\frac{\tan \beta_{out}}{sr_{out}} \quad \text{along DE} \tag{6}$$

These are the boundary conditions required to determine a solution to equation (1). The method used for the numerical solution of equation (1) is described in appendix A.

After computing a numerical solution to equation (1) in a given flow region, the velocity at any point can be computed from equations (2) and (3) by using numerical differentiation. The streamlines are located by the contours of equal stream-function values.

NUMERICAL EXAMPLES

To illustrate the use of the program and to show the type of results which can be obtained, two numerical examples are given. The first example is a radial inflow turbine, and the other is an axial turbine stator.

Radial Inflow Turbine Rotor

The turbine profile is shown in figure 5 with the mean streamlines and mean stream-sheet thickness as calculated by the quasi-orthogonal method (ref. 7). The blade-to-blade shape in θ and m coordinates is shown in figure 4. This particular rotor had splitter blades as indicated in figure 5. There were 11 complete blades and 11 splitter blades. The program cannot handle the case where the blades are not all identical. However, two solutions can be obtained, one based on 11 blades and one based on 22 blades. The solution with 11 blades should be reasonable for the region beyond the splitter blades, and the solution with 22 blades should be reasonable for the region with the splitter blades. Note that this technique would not work for a compressor with splitter blades, since the percentage of flow on each side of the splitter blade would not be known.

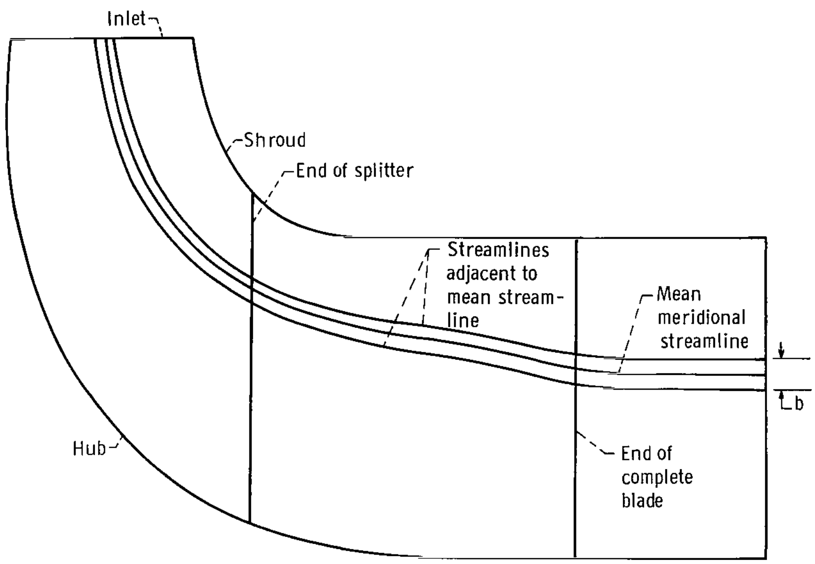


Figure 5. - Hub-shroud profile with streamlines used for blade-to-blade analysis of a radial turbine.

The input for the case with 11 blades is given in table I. Because of a high local velocity near the trailing-edge radius, the program did not converge at every point (see error condition (6) (p. 36) for further discussion). However, the solution failed to converge at only one point, so that the rest of the velocities should be valid.

The results are plotted in figure 6. There is also shown in this figure a solution obtained by the quasi-orthogonal method of reference 8. To make the results comparable, the quasi-orthogonal solution was obtained for zero loss. There is close agreement on

TABLE I. - ELEVEN-BLADE RADIAL INFLOW TURBINE ROTOR AND COMPUTER INPUTS

```
   GAM              AR              TIP             RHOIP            WTFL            OMEGA            W
 1.6667000       208.20000       1083.0000        0.3556600      0.1258000E-02    4030.0000       -0
  CHORD            STGR            BETAI           BETAO
 0.6844000E-01  -0.5390000      -54.200000      -61.500000
    RI             ALLI            ALLI              RC              ALUO             ALLO
 0.6480000E-03   2.0000000      -2.0000000      0.7530000E-03   -63.800000       -63.800000
MXBI MXBC   MX NRPI NLSF NLSF NRSP  NBL NINT
 5   39    50   15   11   11   16   11   5
      MU ARRAY
-0              0.8600000E-02   0.1600000E-01   0.2350000E-01   0.2903000E-01   0.3428000E-01   0.3954000E-01   0.4623000E-01
 0.5394000E-01  0.6197000E-01  -0
      XSPU ARRAY
-0              0.1220000E-01   0.1560000E-01   0.1890000E-01   0.2090000E-01   0.2150000E-01   0.1600000E-01  -0.1160000E-01
-0.9620000E-01 -0.2791000      -0.5427000
      ML ARRAY
-0              0.8600000E-02   0.1600000E-01   0.2350000E-01   0.2903000E-01   0.3428000E-01   0.3954000E-01   0.4623000E-01
 0.5394000E-01  0.6197000E-01  -0
      XSPL ARRAY
-0             -0.1220000E-01  -0.1560000E-01  -0.1900000E-01  -0.2070000E-01  -0.2220000E-01  -0.2840000E-01  -0.5580000E-01
-0.1452000     -0.3347000      -0.5842000
      MR ARRAY
-0.7620000E-02  0               0.8600000E-02   0.1600000E-01   0.2350000E-01   0.2903000E-01   0.3428000E-01   0.3954000E-01
 0.4623000E-01  0.5394000E-01   0.6197000E-01   0.6844000E-01   0.7354000E-01   0.8116000E-01   0.8878000E-01   0.9700000E-01
      RMSP ARRAY
 0.8407000E-01  0.7645000E-01   0.6800000E-01   0.6103000E-01   0.5471000E-01   0.5089000E-01   0.4808000E-01   0.4602000E-01
 0.4435000E-01  0.4295000E-01   0.4131000E-01   0.4005000E-01   0.3964000E-01   0.3944000E-01   0.3940000E-01   0.3940000E-01
      BESP ARRAY
 0.9700000E-03  0.9600000E-03   0.1030000E-02   0.1090000E-02   0.1140000E-02   0.1160000E-02   0.1160000E-02   0.1240000E-02
 0.1330000E-02  0.1430000E-02   0.1530000E-02   0.1620000E-02   0.1670000E-02   0.1690000E-02   0.1700000E-02   0.1700000E-02
BLDATA NULAKI  ERPRT  STRFN  SLCRD  ARPRT INTVEL SURVEL
   1      0      3      2      2      0     1      3
```

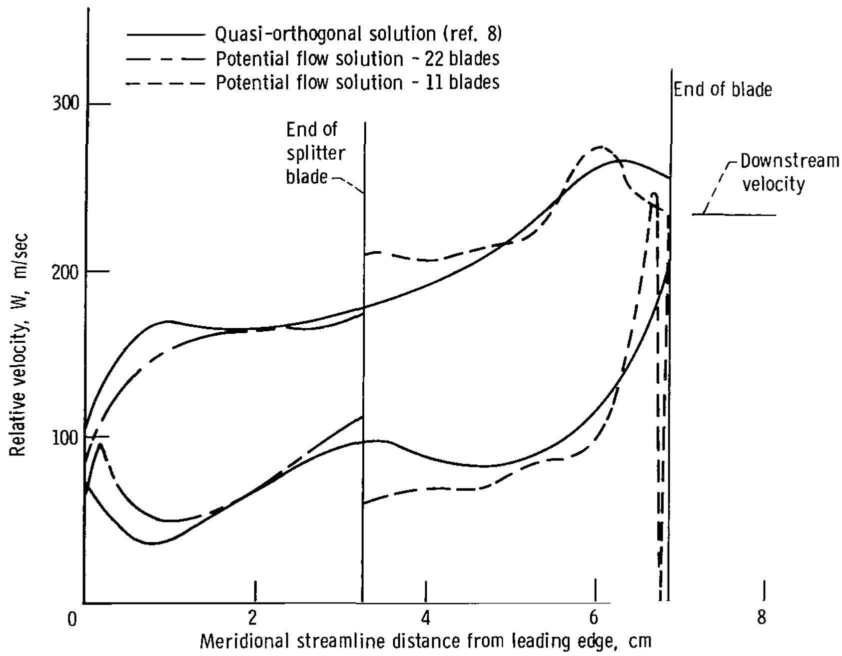


Figure 6. - Blade surface velocities for a radial turbine with splitter blade.

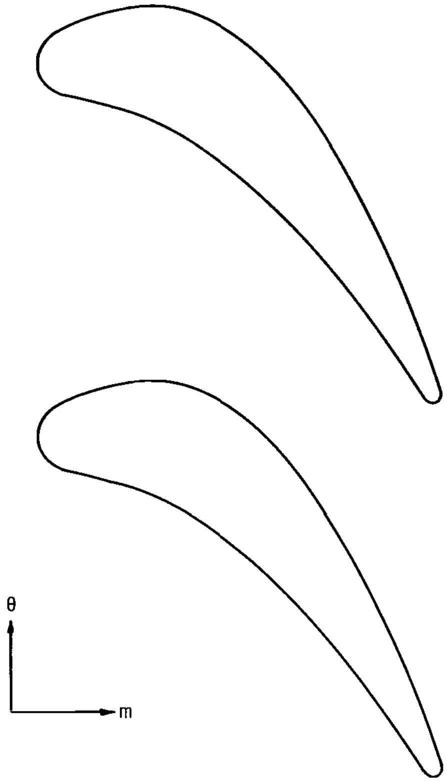


Figure 7. - Axial stator blade for numerical example.

TABLE II. - AXIAL STATOR COMPUTER INPUT

```
     GAM              AR              TIP             RHOIP          WTFL          OMEGA             W
  1.4000000       287.05300       288.15000       1.2250000      0.3146000      0               1.8500000
    CHORD             STGR            BETAI           BETAC
  0.4265000E-01  -0.1116150       0               -67.000000
     RI              ALUI             ALLI             RO            ALUO           ALLO
  0.3810000E-02   28.300000       -14.200000      0.8890000E-03  -72.400000     -56.100000
MXBI MXBO   MX NBBI NUSP NLSP NRSP  NBL NINT
 15   32    47   20    7    6    2   50    5
        MU  ARRAY
 -0                0.8575000E-02   0.1715000E-01   0.2572500E-01   0.3430000E-01   0.3858800E-01  -0
        XSPU ARRAY
 -0                0.1769000E-01   0.1538000E-01  -0.5310000E-02  -0.4654000E-01  -0.7400000E-01  -0
        ML  ARRAY
 -0                0.8575000E-02   0.1715000E-01   0.2572500E-01   0.3430000E-01  -0
        XSPL 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
 BLDATA NULAKI  ERPRT  STRFN  SLCRD  ARPRT INTVEL SURVEL
     1      0      3      1      2      0      1      3
```

TABLE III. - AXIAL STATOR COMPUTED SURFACE VELOCITIES

(a) Surface velocities based on axial components

	UPPER SURFACE				LOWER SURFACE			
M	VELOCITY	ANGLE (DEG)	SURF. LENGTH	RHC*W	VELOCITY	ANGLE (DEG)	SURF. LENGTH	RHC*W
0	0	90.00	0	-6.3313	0	-90.00	0	-8.9717
0.2509E-02	97.625	27.31	0.4405E-02	114.73	76.541	-20.67	0.4372E-02	91.409
0.5018E-02	111.01	21.66	0.7165E-02	128.86	64.625	-13.80	0.6959E-02	77.745
0.7526E-02	128.90	14.63	0.9807E-02	146.82	60.592	-16.21	0.9547E-02	73.054
0.1004E-01	148.82	6.20	0.1236E-01	165.37	59.431	-21.52	0.1219E-01	71.698
0.1254E-01	168.62	-3.39	0.1487E-01	182.13	60.967	-26.75	0.1492E-01	73.491
0.1505E-01	185.87	-13.61	0.1741E-01	195.23	64.056	-31.55	0.1777E-01	77.085
0.1756E-01	199.22	-23.78	0.2006E-01	204.35	68.661	-35.91	0.2075E-01	82.408
0.2007E-01	206.91	-33.36	0.2292E-01	209.18	74.600	-39.87	0.2387E-01	89.205
0.2258E-01	211.94	-41.92	0.2609E-01	212.17	82.039	-43.47	0.2716E-01	97.602
0.2509E-01	211.49	-49.21	0.2969E-01	211.90	91.119	-46.74	0.3063E-01	107.66
0.2760E-01	205.15	-54.84	0.3379E-01	208.10	101.90	-49.74	0.3428E-01	119.31
0.3011E-01	198.59	-58.36	0.3837E-01	203.94	114.60	-52.57	0.3814E-01	132.55
0.3261E-01	195.85	-60.50	0.4331E-01	202.14	129.33	-55.22	0.4223E-01	147.24
0.3512E-01	202.03	-61.99	0.4852E-01	206.15	146.40	-57.62	0.4656E-01	163.20
0.3763E-01	209.68	-66.76	0.5428E-01	210.84	164.98	-58.75	0.5108E-01	179.18
0.4014E-01	192.93	-71.68	0.6150E-01	200.16	170.02	-58.49	0.5566E-01	183.24
0.4265E-01	0	-90.00	0.6989E-01	51.236	0	90.00	0.5830E-01	-47.293

(b) Surface velocities based on tangential components

	UPPER SURFACE		
M	VELOCITY	ANGLE (DEG)	RHC*W
-0	10.858	90.00	13.294
0.6157E-03	78.376	56.97	93.484
0.3586E-02	95.898	25.04	112.87
0.1906E-01	204.51	-29.59	207.71
0.2201E-01	209.76	-40.09	210.89
0.2419E-01	212.11	-46.76	212.26
0.2598E-01	209.87	-51.50	210.95
0.2753E-01	205.28	-54.73	208.18
0.2894E-01	200.81	-56.93	205.37
0.3025E-01	197.72	-58.52	203.37
0.3149E-01	196.38	-59.68	202.49
0.3268E-01	195.76	-60.54	202.08
0.3384E-01	197.02	-61.16	202.91
0.3497E-01	201.14	-61.82	205.59
0.3605E-01	206.82	-63.34	209.12
0.3704E-01	208.97	-65.38	210.42
0.3795E-01	208.33	-67.52	210.04
0.3876E-01	204.59	-69.51	207.76
0.3951E-01	199.01	-70.89	204.21
0.4021E-01	192.51	-71.74	199.88
0.4088E-01	187.24	-72.25	196.21
0.4154E-01	182.58	-72.50	192.84
0.4219E-01	184.58	-72.51	194.30
0.4265E-01	167.51	-90.00	181.23

	LOWER SURFACE		
M	VELOCITY	ANGLE (DEG)	RHC*W
0.6158E-03	51.650	-56.97	62.545
0.4764E-02	38.457	-13.29	46.810
0.1139E-01	60.251	-23.59	72.656
0.1539E-01	64.970	-31.07	78.146
0.1850E-01	70.903	-36.17	84.983
0.2114E-01	77.737	-40.05	92.762
0.2347E-01	85.262	-43.17	101.20
0.2558E-01	93.344	-45.76	110.09
0.2753E-01	101.76	-47.99	119.16
0.2933E-01	110.64	-49.97	128.49
0.3102E-01	119.90	-51.75	137.93
0.3261E-01	129.46	-53.35	147.36
0.3411E-01	139.13	-54.81	156.55
0.3554E-01	149.59	-55.96	166.05
0.3692E-01	159.80	-56.60	174.88
0.3828E-01	169.65	-56.83	182.95
0.3964E-01	178.75	-56.67	190.00
0.4102E-01	199.67	-56.10	204.64
0.4250E-01	80.078	56.49	95.402

the splitter blade. On the complete blade, there is a short distance after the end of the splitter blade before the complete blade assumes the blade loading without a splitter blade.

Axial Stator

This example is a stator nozzle mean blade section (fig. 7) for a turbine built at Lewis Research Center (ref. 9). This blade section has also been analyzed by using the incompressible flow program of reference 5. Downstream of the blade V/V_{cr} is 0.58. The input for this case is given in table II. The surface velocities obtained by the program are given in table III. The velocities are plotted against blade surface length in figure 8. Also shown in figure 8 are experimental data obtained from the investigation described in reference 9. There is fairly good agreement with the computed values for this example.

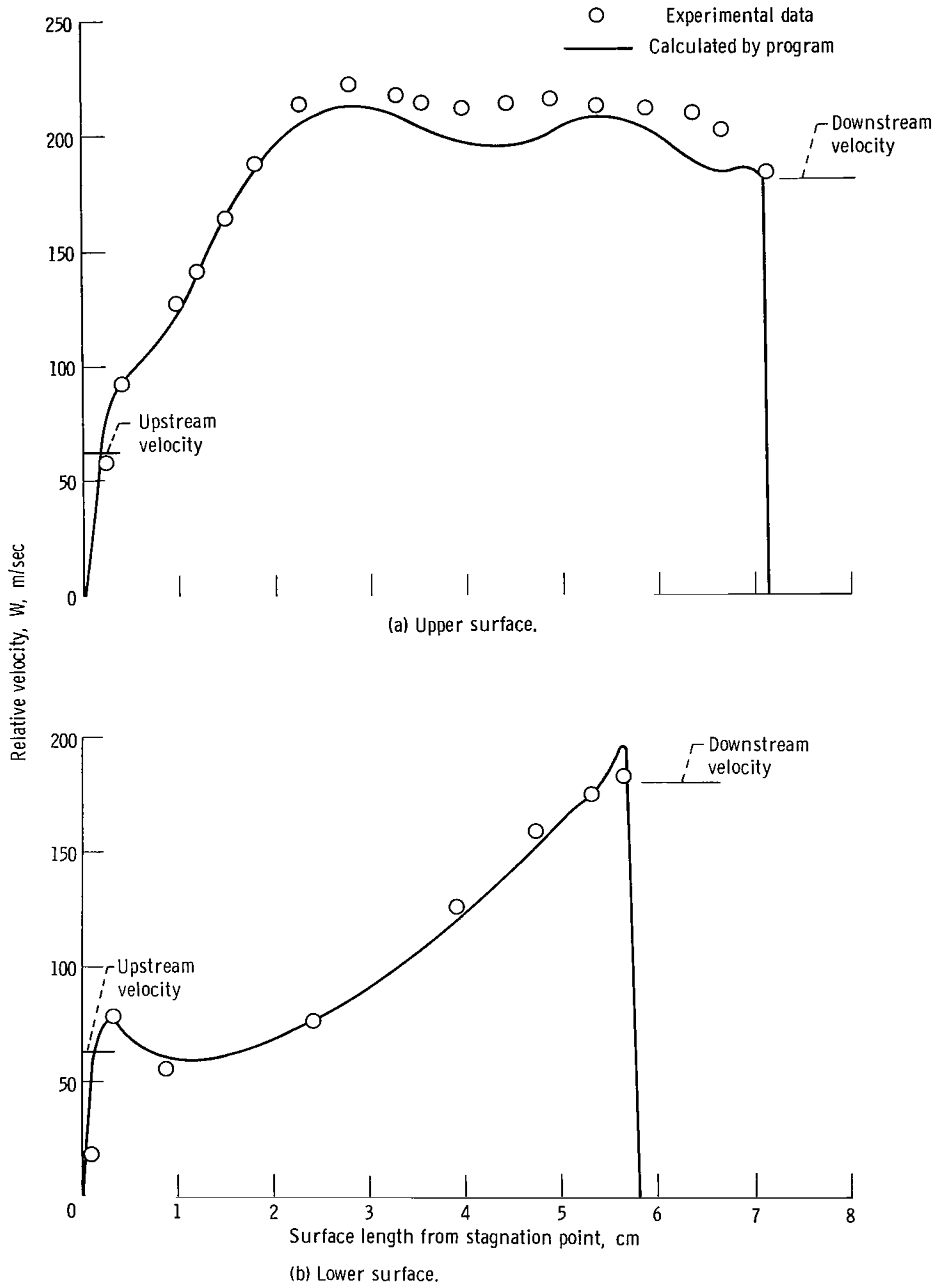


Figure 8. - Blade surface velocity for numerical example compared with experimental data.

DESCRIPTION OF INPUT AND OUTPUT

The computer program requires as input sufficient information to describe the blade shape accurately, the inlet and outlet angles, the extent of the region to be considered, the mesh size to be used, the appropriate gas constants, and operating conditions such as inlet temperature, density, weight flow, and rotational speed. Output obtained from the program includes velocity magnitude and direction at all interior points, blade surface velocities, stream function values, and streamline locations if there is no reverse flow. If there is reverse flow (as may occur with radial flow), streamline locations may be obtained by plotting contours of equal stream function values.

Instructions for Preparing Input

Figure 9 shows the input variables as they are to be punched on the data cards. There are two types of input variables, geometric and nongeometric. The geometric input variables are shown in figures 10 and 11. Units are always meters for length and radians for θ coordinates.

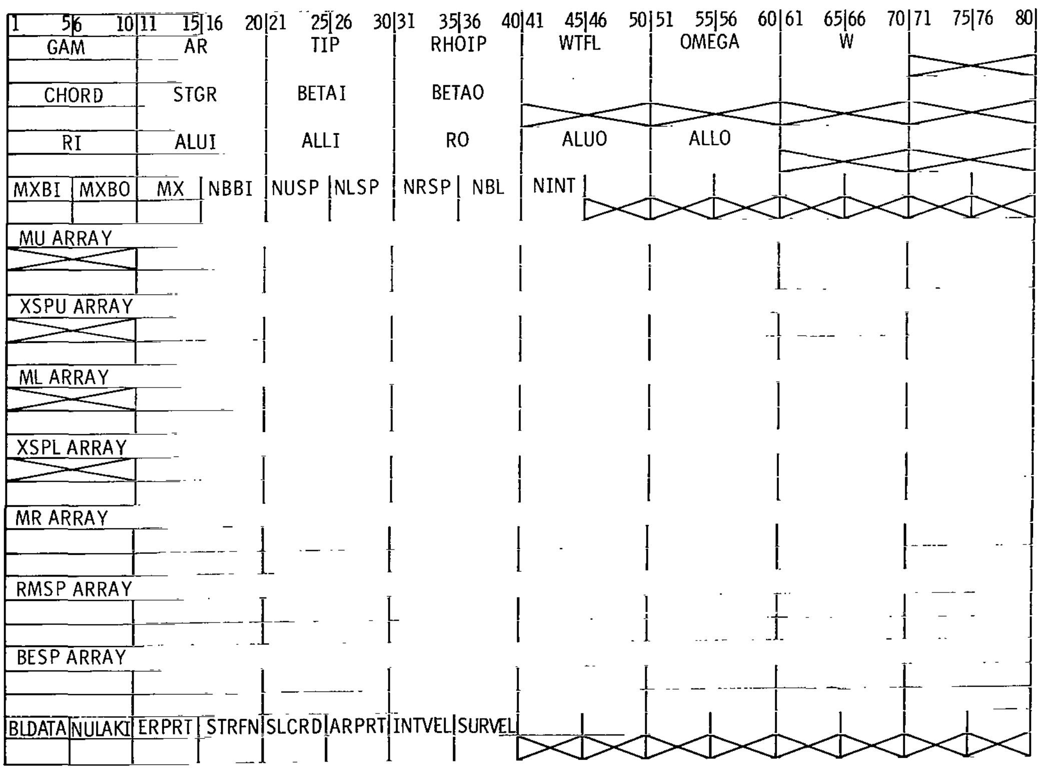


Figure 9. - Input form. (Card column numbers appear at top.)

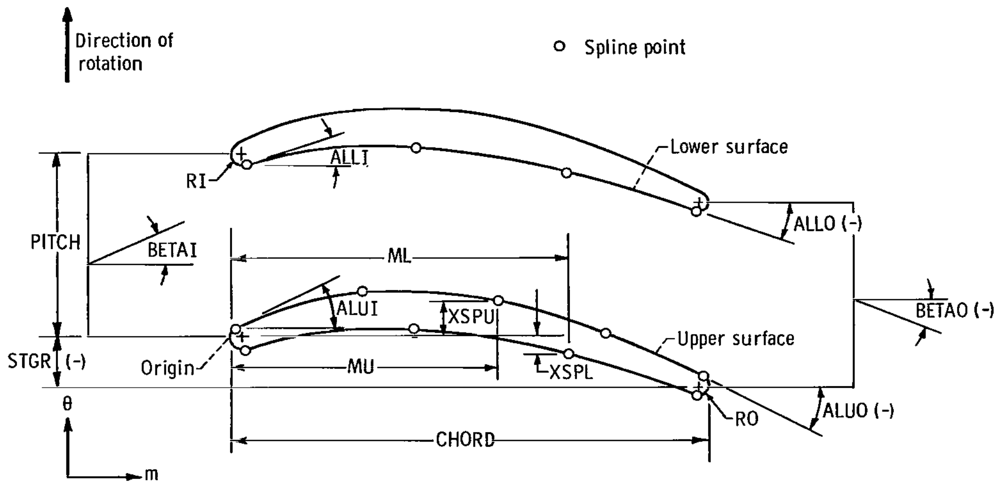


Figure 10. - Geometric variables required as input. Blade-to-blade coordinates on stream surface The variables BETAI, BETAO, ALUI, ALLI, ALUO, and ALLO are to be given as a true angle β, not the angle as measured in the m, θ plane. (Use $\tan \beta = r(d\theta/dm)$ to obtain the value of β.)

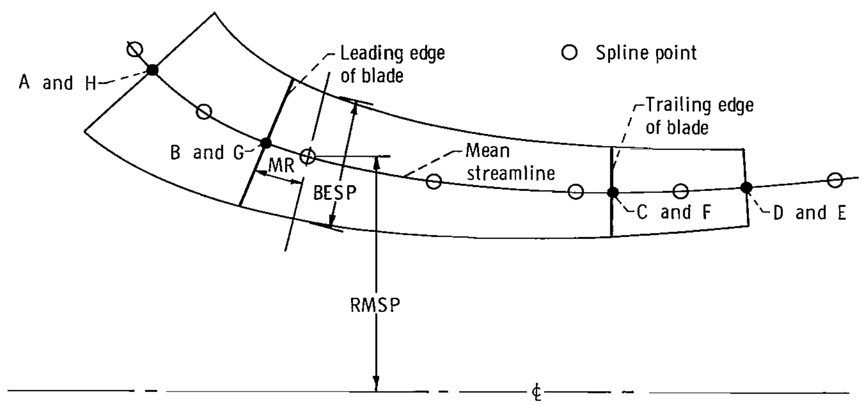


Figure 11. - Geometric variables required as input. Meridional plane.

The blade shape is defined by specifying the leading and trailing edge radii and a number of blade surface m and θ coordinates. These coordinates are used to define a cubic spline curve (refs. 5 and 10). The coordinates are given with respect to the leading edge of the lower blade, as shown in figure 10. The standard sign convention is used for angles, as indicated in figure 10. The blade should be oriented with the concave side down.

The mean stream surface of revolution and normal stream channel thickness are also defined by cubic spline curves as indicated in figure 11. The m coordinates for the mean stream surface are independent of the blade shape coordinates.

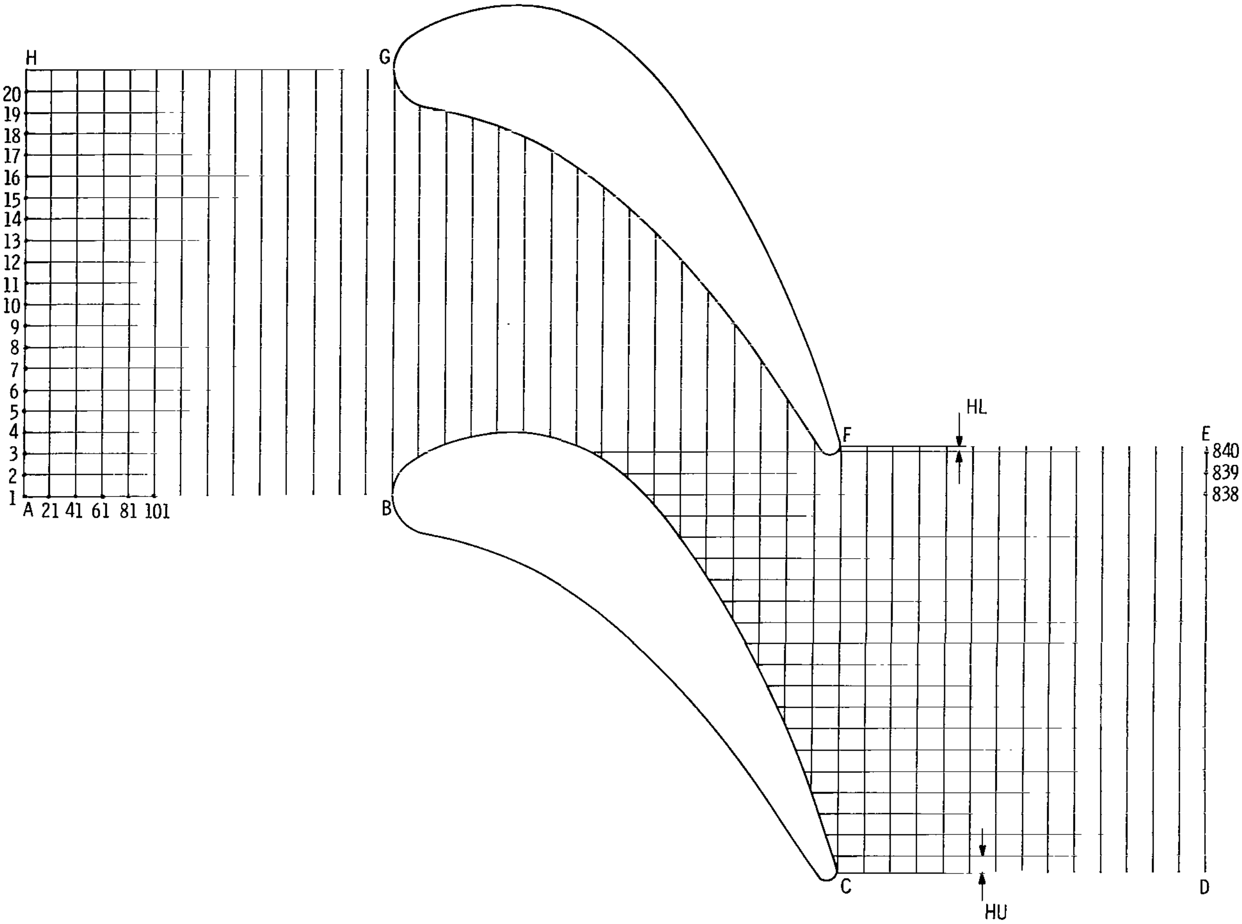


Figure 12. - Mesh used for axial stator numerical example. Numbers are mesh point indexes (I in program). There are 840 unknown mesh points.

A mesh must be used for the finite-difference solution of equation (1). A typical mesh pattern is shown in figure 12. The mesh spacing and the extent of the upstream and downstream regions are determined by the values of MXBI, MXBO, MX, and NBBI. The mesh spacing must be chosen so that there are not more than 2500 unknown mesh points.

The values of β_{in} and β_{out} must be given on AH and DE, respectively. However, it may be that the average values along BG and CF are what is known. In this case the input values, β_{in} and β_{out}, must be calculated by equation (B13) or (B15).

While the program was written for compressible flow, it can be used equally well for incompressible flow. To use the program for incompressible flow specify GAM = 1.5, AR = 1000, and TIP = 10^6 as input. Of course, RHOIP is simply the density in kilograms per cubic meter. This results in one iteration only.

The International System of Units (ref. 11) is used throughout. However, the program does not use any constants which depend on the system of units being used (other than degrees or radians where specified) so that any other consistent set of units may be used. For example, force, length, temperature, and time units may be chosen independently (mass units defined by $f = ma$); 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). Output then gives the velocity in the chosen units of length per unit time, and, of course, the output is not labeled with the correct units.

Input

All the numbers on the card beginning with MXBI and on the card beginning with BLDATA are integers (no decimal point) in a 5-column field (see fig. 9). Integers must be right adjusted. The remaining input variables are real numbers (punch decimal point) in a 10-column field.

The input variables are as follows:

GAM	specific heat ratio, γ
AR	gas constant, R, J/(kg)($^{\circ}$K)
TIP	inlet total temperature, T'_{in}, $^{\circ}$K
RHOIP	inlet total density, ρ'_{in}, kg/m^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. 10.)
W	value for overrelaxation factor Ω to be used in eq. (A8) (If W = 0, the program calculates an estimated value for the overrelaxation factor, see appendix A for discussion.)
CHORD	overall length of blade in m direction, meters, see fig. 10
STGR	angular coordinate θ for center of trailing-edge radius, radians, see fig. 10
BETAI	flow angle β_{in} along AH, deg, see fig. 10
BETAO	flow angle β_{out} along DE, deg, see fig. 10
RI	leading-edge radius, m, see fig. 10
ALUI	angle at tangent point of leading-edge radius on upper surface, deg, see fig. 10
ALLI	angle at tangent point of leading-edge radius on lower surface, deg, see fig. 10
RO	trailing-edge radius, m, see fig. 10
ALUO	angle at tangent point of trailing-edge radius on upper surface, deg, see fig. 10

ALLO angle at tangent point of trailing-edge radius on lower surface, deg, see fig. 10

MXBI number of mesh lines from AH to BG inclusive (fig. 12)

MXBO number of mesh lines from AH to CF inclusive (fig. 12)

MX total number of mesh lines in m direction from AH to DE, maximum of 100 (fig. 12)

NBBI number of mesh spaces in θ direction between AB and HG, maximum of 50 (fig. 12)

NUSP number of blade spline points including end points that are tangent to leading- and trailing-edge radii for upper surface (BC) of blade (figs. 4 and 10); maximum of 50

NLSP same as NUSP, but for the lower surface (GF) of the blade

NRSP number of spline points for streamsheet radius (RMSP) and thickness (BESP) coordinates (see fig. 11), maximum of 50

NBL number of blades

NINT number of streamlines desired as output, maximum of 10

MU array of m-coordinates of spline points for upper surface measured from leading edge, m, fig. 10 (The first and last points must be left blank, since these points are calculated by the program. If the last point is on a new card, a blank card must be used. The total number of points is NUSP.)

XSPU array of θ-coordinates corresponding to the MU array, rad

ML same as MU but for lower surface (The total number of points is NLSP.)

XSPL array of θ-coordinates corresponding to the ML array, rad (Note that these coordinates are to the lower blade as shown in fig. 10.)

MR array of m-coordinates of spline points for stream surface radii and stream channel thickness measured from leading edge, m, see fig. 11 (These coordinates should include the entire distance from AH to DE, and may extend beyond these points if desired. The total number of points is NRSP.)

RMSP array of stream surface radii corresponding to the MR array, m

BESP array of stream channel thicknesses corresponding to the MR array, m

The remaining variables, starting with BLDATA, 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 iteration only; 2, for the first and final iteration; and 3, for all 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.

BLDATA radii and stream sheet thickness at each vertical mesh line; coordinates, first and second derivative of blade spline points; θ-coordinate and slope at each vertical mesh line for each blade surface; coordinates of intersection of horizontal mesh lines with blade; NU and NL arrays (internal variables) (This will be printed for the first iteration only since these values do not change.)

NULAKI coefficient array A, the vector K, and the value of I for the adjacent points I1, I2, I3, and I4 (This information is needed for debugging the program only.)

ERPRT the maximum change in the stream function for each iteration of the SOR equation, eq. (A8)

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

SLCRD streamline coordinates at each vertical mesh line and streamline plot

ARPRT values for (ρW_m) and (ρW_θ) at all interior points and along blade surfaces, value of ρW at all interior points (This information is needed for debugging the program only.)

INTVEL velocity and flow angle at all interior mesh points

SURVEL m-coordinate, surface velocity, flow angle, distance along surface, and ρW based on meridional velocity components; m-coordinate, surface velocity, flow angle, and ρW based on tangential components; plot of blade surface velocities against meridional streamline distance m (It is suggested that SURVEL = 3 be used. This will give surface velocities on every iteration, so that satisfactory velocities may be obtained even when final convergence is not reached in the alloted time.)

Output

Sample output is given for the radial inflow turbine numerical example, but with an outlet angle of -66.5°. Since the complete output would be lengthy, only the first few lines of each type of output are reproduced here. Most of the output is optional and is controlled by the last input card as already described. The output labels are either internal variable names or else are spelled out (e.g., THETA for θ).

Each section of the sample output (table IV) has been numbered to correspond to the following descriptions:

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

(2) The calculated value of λ is given followed by W (free-stream velocity) and the maximum value of the mass flow parameter ρW (corresponding to $W = W_{cr}$) along AH (inlet) and DE (outlet). As a check, the free-stream values of β at leading edge (BG) and trailing edge (CF) corresponding to the input values of β_{in} and β_{out}, respectively, are calculated and printed out under the heading "BETA CORRECTED TO BLADE LE OR TE". The relative critical velocity W_{cr} at BG (inlet) and CF (outlet) is printed out. Also some internal program constants are printed out at this point.

(3) This is the output corresponding to BLDATA (see the list of input variables).

(4) This is the number of mesh points at which the stream function is unknown.

(5) This is the output corresponding to NULAKI.

(6) If the program calculates an optimum overrelaxation factor Ω (i.e., W = 0 for input), then upper and lower bounds for Ω (WMAX and WMIN) and upper and lower bounds for $\rho(L_1)$, (LMAX and LMIN) are printed out for each iteration (see appendix B of ref. 5 for details). The last printed value of WMAX is the value of Ω (W) used by the program.

(7) This is the output corresponding to ERPRT.

(8) This is the output corresponding to STRFN.

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

(10) This is the output corresponding to SLCRD.

(11) This is the output corresponding to ARPRT.

(12) This is the output corresponding to INTVEL.

(13) This gives the maximum relative change in the density ρ for each outer iteration.

(14) This is the output corresponding to SURVEL.

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

TABLE IV. - SAMPLE OUTPUT

Description[a]

1

```
     GAM               AR                TIP               RHOIP             WTFL              OMEGA             W
  1.6667000         208.20000         1083.0000         0.3956600         0.6290000E-03     4030.0000         0
     CHORD             STGR              BETAI             BETAC
  0.6844000E-01    -0.5350000        -54.200000        -66.500000
     RI                ALUI              ALLI              RO                ALUO              ALLO
  0.6480000E-03     2.0000000        -2.0000000         0.7530000E-03    -63.800000        -63.800000
MXBI MXBO  MX NBBI NUSP NLSP NRSP  NBL NINT
  5   39   45   8   11   11   16    22   5
        MU ARRAY
 -0                 0.8600000E-02     0.1600000E-01     0.2350000E-01     0.2903000E-01     0.3428000E-01     0.3954000E-01     0.4623000E-01
  0.5394000E-01     0.6197000E-01    -0
        XSPL ARRAY
 -0                 0.1220000E-01     0.1560000E-01     0.1890000E-01     0.2090000E-01     0.2150000E-01     0.1600000E-01    -0.1160000E-01
 -0.9620000E-01    -0.2791000        -0.5427000
        ML ARRAY
 -0                 0.8600000E-02     0.1600000E-01     0.2350000E-01     0.2903000E-01     0.3428000E-01     0.3954000E-01     0.4623000E-01
  0.5394000E-01     0.6197000E-01    -0
        XSPL ARRAY
 -0                -0.1220000E-01    -0.1560000E-01    -0.1900000E-01    -0.2070000E-01    -0.2220000E-01    -0.2840000E-01    -0.5580000E-01
 -0.1452000        -0.3347000        -0.5842000
        MR ARRAY
 -0.7620000E-02     0                 0.8600000E-02     0.1600000E-01     0.2350000E-01     0.2503000E-01     0.3428000E-01     0.3954000E-01
  0.4623000E-01     0.5394000E-01     0.6197000E-01     0.6844000E-01     0.7354000E-01     0.8116000E-01     0.8878000E-01     0.9700000E-01
        RMSP ARRAY
  0.8407000E-01     0.7645000E-01     0.6800000E-01     0.6103000E-01     0.5471000E-01     0.5085000E-01     0.4808000E-01     0.4602000E-01
  0.4435000E-01     0.4295000E-01     0.4131000E-01     0.4005000E-01     0.3964000E-01     0.3944000E-01     0.3940000E-01     0.3940000E-01
        BESP ARRAY
  0.9700000E-03     0.5600000E-03     0.1030000E-02     0.1090000E-02     0.1140000E-02     0.1160000E-02     0.1160000E-02     0.1240000E-02
  0.1330000E-02     0.1430000E-02     0.1530000E-02     0.1620000E-02     0.1670000E-02     0.1650000E-02     0.1700000E-02     0.1700000E-02
BLDATA NULAKI  ERPRT  STRFN  SLCRD  ARPRT INTVEL SURVEL
    1      1      3      2      2      1      2      3
```

2

```
LAMBDA =   20.13600
             FREESTREAM       MAXIMUM VALUE     BETA CORRECTED TO    BLADE CRITICAL
              VELOCITY         FOR RHO*W         BLADE LE OR TE        VELOCITY
INLET         126.08875         125.45426         -28.427796          514.78984
OUTLET        250.69178         105.28652         -65.998662          497.78524


          CALCULATED PROGRAM CONSTANTS
     HA                HB                HU                HL                PITCH                 NBBO       NBUO
  0.2012941E-02     0.3569592E-01     0.3501244E-02     0.3219867E-01     0.2855993               -8         -16
```

3

```
                 STREAM SHEET COORDINATES AND THICKNESS TABLE
       M                R               SAL              B              CB/DM
  -0.80518E-02     0.84504E-01     -1.00521       0.97247E-03     -0.58887E-02
  -0.60388E-02     0.82482E-01     -1.00316       0.96235E-03     -0.40874E-02
  -0.40259E-02     0.80466E-01     -1.00060       0.95631E-03     -0.18435E-02
  -0.20129E-02     0.78454E-01     -0.99755       0.95523E-03      0.84291E-03
   0               0.76450E-01     -0.95398       0.96000E-03      0.39718E-02
```

[a]See p. 20.

TABLE IV. - Continued. SAMPLE OUTPUT

Description (a)

Description				
	BLADE DATA AT SPLINE POINTS			
		UPPER SURFACE		
	M	THETA	DERIVATIVE	2ND DERIV.
3	0.62754E-03	0.84719E-02	0.41324	24.7234
	0.86000E-02	0.12200E-01	0.47782	-8.52274
	0.16000E-01	0.15600E-01	0.45426	2.15530

Description				
		LOWER SURFACE		
	M	THETA	DERIVATIVE	2ND DERIV.
3	0.62754E-03	0.27713	-0.41324	-25.5529
	0.86000E-02	0.27340	-0.47452	10.1817
	0.16000E-01	0.27000	-0.46702	-8.15426

Description					
	BLADE COORDINATE TABLE				
	M	XU	DXDZU	XL	DXDZL
	0	0	50000000000	0.28560	50000000000
3	0.20129E-02	0.50663E-02	0.44349	0.27653	-0.44434
	0.40259E-02	0.99917E-02	0.47318	0.27560	-0.47420
	0.60388E-02	0.10960E-01	0.48598	0.27464	-0.48589
	0.80518E-02	0.11937E-01	0.48187	0.27366	-0.47943

Description				
	M	INU	INL	ITP
	0	1		
	0.4230E-01		1	1
3	0.4424E-01	2		2
	0.4814E-01		2	2
	0.4919E-01	3		3

4 NUMBER OF INTERIOR MESH POINTS = 328

Description		
	LIST OF NU AND NL	
	0	7
	0	7
3	0	7
	0	7
	1	7

[a]See p. 20.

TABLE IV. - Continued. SAMPLE OUTPUT

Description (a)

6

```
WMAX = 2.000000     WMIN = 1.000693     LMAX = 1.000000     LMIN = 0.002768
WMAX = 2.000000     WMIN = 1.125091     LMAX = 1.000000     LMIN = 0.395284
WMAX = 2.000000     WMIN = 1.200986     LMAX = 1.000000     LMIN = 0.557378
WMAX = 1.990991     WMIN = 1.234761     LMAX = 0.999980     LMIN = 0.615915
WMAX = 1.960779     WMIN = 1.246855     LMAX = 0.999600     LMIN = 0.635141
WMAX = 1.950149     WMIN = 1.250514     LMAX = 0.999347     LMIN = 0.640789
WMAX = 1.939927     WMIN = 1.261752     LMAX = 0.999041     LMIN = 0.657660
WMAX = 1.930263     WMIN = 1.274517     LMAX = 0.998695     LMIN = 0.675987
```

7

```
ERROR = 1.03856773
ERROR = 0.66880200
ERROR = 0.88835229
ERROR = 0.41945254
ERROR = 0.58963882
ERROR = 0.29354919
ERROR = 0.38924909
```

8

```
          STREAM FUNCTION VALUES
IA =  1
   -0.35951178  -0.23676679  -0.11125914   0.01617927   0.14364282   0.26979860   0.39430712   0.51764261
IA =  2
   -0.24386856  -0.12112157   0.00438607   0.13182450   0.25928804   0.38544382   0.50995234   0.63328782
IA =  3
   -0.14629163  -0.02489553   0.10100400   0.22934018   0.35743163   0.48374735   0.60803247   0.73104610
IA =  4
   -0.06585617   0.05129842   0.17862408   0.30920815   0.43870088   0.56530648   0.68898664   0.81121481
IA =  5
    0.10444354   0.23685549   0.37202834   0.50390199   0.63082559   0.75311223   0.87340546
```

9

```
TIME =  0.2836 MIN.
```

10

```
                         STREAMLINE COORDINATES

  M COORD.          STREAM FN.       THETA            STREAM FN.     THETA             STREAM FN.    THETA

-0.80517658E-02   -0.2000000      0.4624002E-01     0              0.1025786         0.2000000     0.1586906
                   0.4000000      0.2158414         0.6000000      0.2738242
-0.60388232E-02   -0.2000000      0.1283459E-01     0              0.70164188E-01    0.2000000     0.1261657
                   0.4000000      0.1826500         0.6000000      0.2402432
-0.40258882E-02   -0              0.4285818E-01     0.2000000      0.58955007E-01    0.4000000     0.1547645
                   0.6000000      0.2118780         0.8000000      0.2699632
-0.20129441E-02   -0              0.2040422E-01     0.2000000      0.77250711E-01    0.4000000     0.1320617
                   0.6000000      0.1884405         0.8000000      0.2466263
 0                 0              0                 0.2000000      0.61959541E-01    0.4000000     0.1146033
                   0.6000000      0.1696966         0.8000000      0.2281792         1.0000000     0.2855993
```

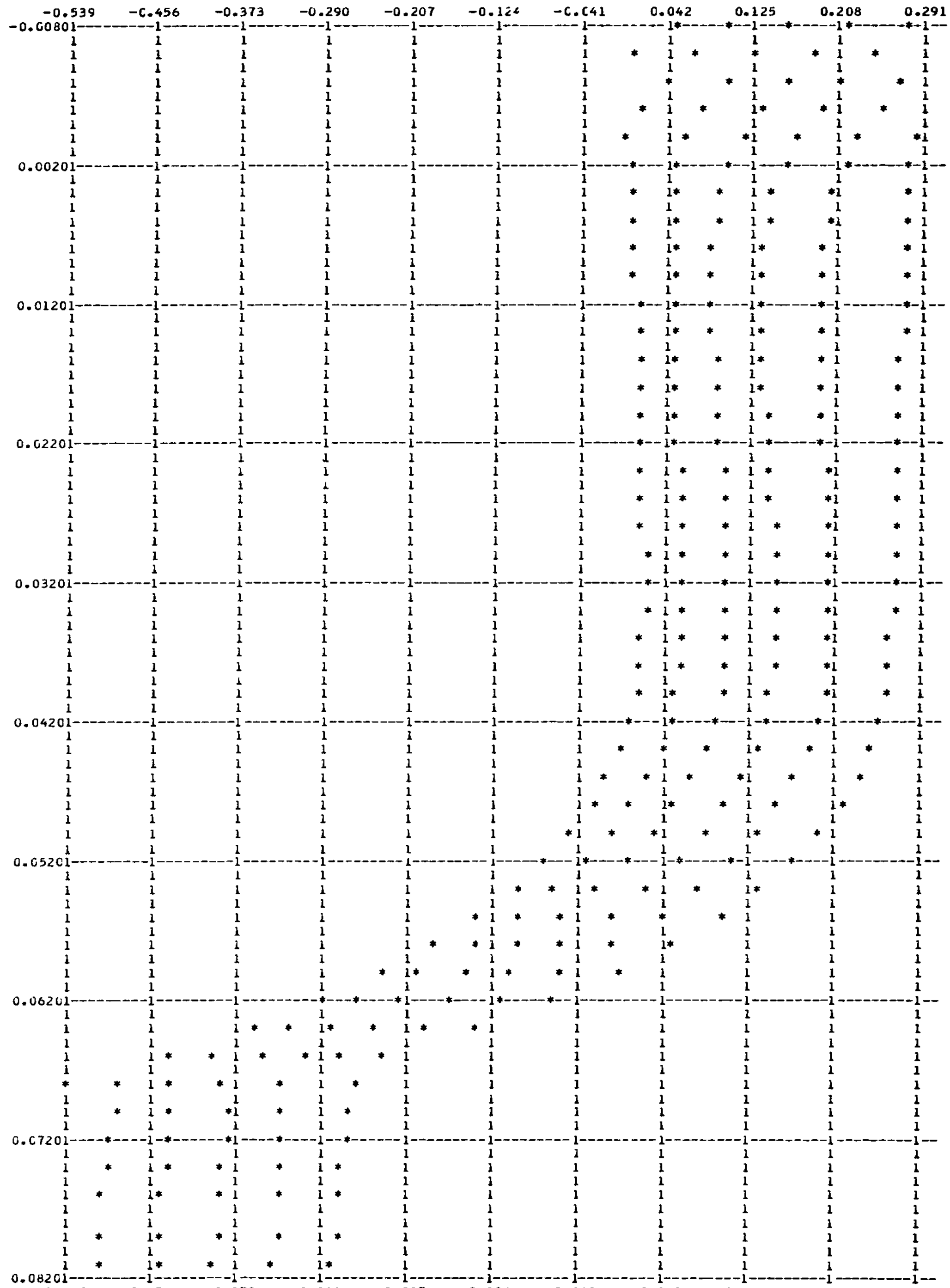
[a]See p. 20.

TABLE IV. - Continued. SAMPLE OUTPUT

Descrip-
tion
(a)

10

STREAMLINE PLOTS



STREAMLINES ARE PLOTTED WITH THETA ACROSS THE PAGE AND M DOWN THE PAGE

[a]See p. 20.

TABLE IV. - Continued. SAMPLE OUTPUT

Description (a)

12

```
                                   VELOCITIES AT INTERIOR MESH POINTS

IA= 1  VELOCITY    ANGLE(DEG)   VELOCITY    ANGLE(DEG)   VELOCITY    ANGLE(DEG)   VELOCITY    ANGLE(DEG)   VELOCITY    ANGLE(DEG)
        131.75      -57.02       133.12      -56.51       133.96      -55.96       134.01      -55.66       133.49      -55.76
        132.66      -56.09       132.41      -56.43       132.25      -56.63

IA= 2  VELOCITY    ANGLE(DEG)   VELOCITY    ANGLE(DEG)   VELOCITY    ANGLE(DEG)   VELOCITY    ANGLE(DEG)   VELOCITY    ANGLE(DEG)
        121.41      -52.11       122.18      -51.33       123.25      -50.78       123.96      -50.65       123.85      -50.91
        123.25      -51.30       122.57      -51.61       122.16      -51.75

IA= 3  VELOCITY    ANGLE(DEG)   VELOCITY    ANGLE(DEG)   VELOCITY    ANGLE(DEG)   VELOCITY    ANGLE(DEG)   VELOCITY    ANGLE(DEG)
        110.54      -46.77       111.07      -44.78       113.49      -44.26       114.55      -44.52       115.02      -45.11
        114.20      -45.66       113.13      -45.95       112.42      -45.94
```

13

```
ITERATION NO.  1   MAXIMUM RELATIVE CHANGE IN DENSITY = 0.8677
```

14

```
             *                    SURFACE VELOCITIES BASED ON AXIAL COMPONENTS                                                        *
             *               UPPER     SURFACE                                   *               LOWER     SURFACE                    *
     M       *   VELOCITY   ANGLE(DEG) SURF. LENGTH      RHO*W                   *     VELOCITY   ANGLE(DEG) SURF. LENGTH    RHO*W    *
 0           *  0              90.00  0                 20.954                  *     0            -90.00   0               31.393   *
 0.2013E-02  *  120.56          1.89  0.2126E-02        42.299                  *     79.725       -1.96   0.2126E-02      28.303   *
 0.4026E-02  *  138.98          1.96  0.4140E-02        48.095                  *     57.442       -2.04   0.4140E-02      20.343   *
 0.6039E-02  *  151.85          1.96  0.6154E-02        51.912                  *     46.589       -2.03   0.6154E-02      16.418   *
 0.8052E-02  *  161.04          1.89  0.8168E-02        54.465                  *     41.495       -1.55   0.8168E-02      14.539   *
 0.1006E-01  *  167.66          1.78  0.1018E-01        56.167                  *     39.893       -1.82   0.1018E-01      13.894   *
```

14

```
 SURFACE VELOCITIES BASED ON TANGENTIAL COMPONENTS
                         UPPER SURFACE
     M            VELOCITY    ANGLE(DEG)     RHO*W

*02*   VALUE OF RHO*W IS TOO LARGE$
           DECK NAME   IFN OF CALL   ABS. LOC.
           .ERROR        00025         16615
           DENSTY        00016         10132
           TASVEL        00524         26113
           ZCCP          00018         03056

  -0            53.341        90.00       19.164
  0.4424E-01   184.90        -12.61       57.795
  0.4919E-01   195.53        -23.06       60.602
  0.5227E-01   206.38        -30.50       63.453
  0.5460E-01   218.32        -36.16       66.525
  0.5652E-01   231.50        -40.57       69.822
```

14

```
                         LOWER SURFACE
     M            VELOCITY    ANGLE(DEG)     RHO*W
  0.4230E-01   129.59         -8.87       41.775
  0.4814E-01   110.54        -22.23       35.681
  0.5133E-01   119.89        -29.54       38.524
  0.5379E-01   128.55        -34.52       41.130
  0.5584E-01   135.69        -38.86       43.255
  0.5759E-01   143.79        -43.13       45.644
```

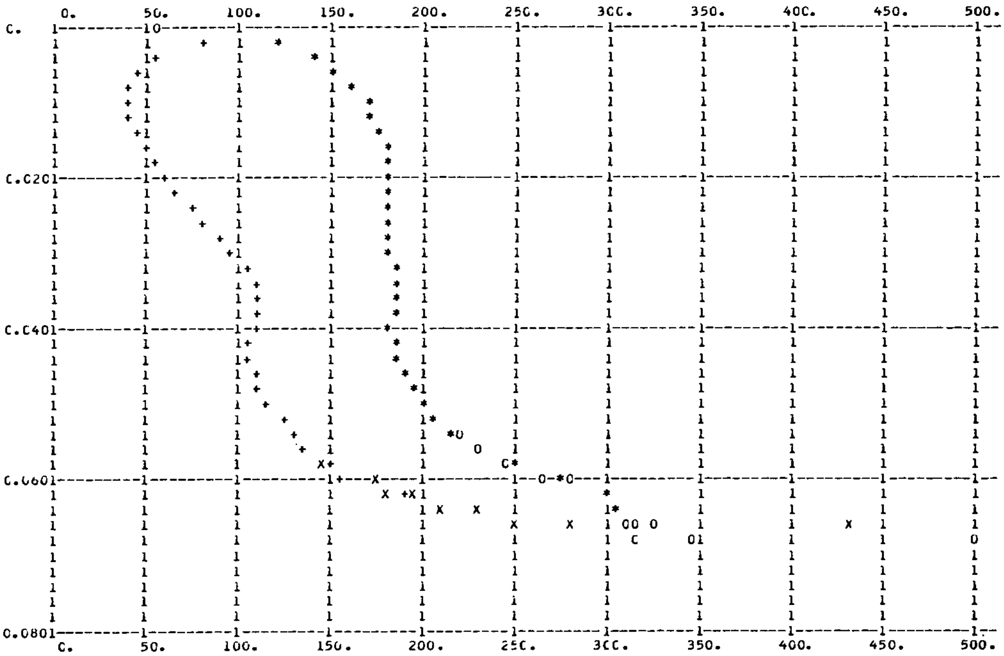
[a]See p. 20.

TABLE IV. - Continued. SAMPLE OUTPUT

Descrip-
tion
(a)

14

BLADE SURFACE VELOCITIES



VELOCITY(METERS/SECOND) VS. MERIDIONAL STREAMLINE DISTANCE(METERS) DOWN THE PAGE

* - UPPER SURFACE, BASED ON AXIAL COMPONENT
\+ - LOWER SURFACE, BASED ON AXIAL COMPONENT
0 - UPPER SURFACE, BASED ON TANGENTIAL COMPONENT
X - LOWER SURFACE, BASED ON TANGENTIAL COMPONENT

[a]See p. 20.

TABLE IV. - Continued. SAMPLE OUTPUT

Description (a)

15
```
TIME =  0.3906 MIN.
```

7
```
ERROR = 0.01259141
ERROR = 0.00940415
ERROR = 0.00755128
ERROR = 0.00603501
ERROR = 0.00576992
ERROR = 0.00585254
ERROR = 0.00609303
ERROR = 0.00538756
```

9
```
TIME =  0.5800 MIN.
```

13
```
ITERATION NO.  2   MAXIMUM RELATIVE CHANGE IN DENSITY = 0.3426E-01
```

14
```
                 *                      SURFACE VELOCITIES BASED ON AXIAL COMPONENTS                                                        *
                 *                 UPPER     SURFACE                              *                 LOWER     SURFACE                   *
      M          *   VELOCITY   ANGLE(DEG) SURF. LENGTH        RHO*W              *     VELOCITY   ANGLE(DEG) SURF. LENGTH        RHO*W   *
 0               *  0              90.00   0                   16.887             *    0             -90.00   0                   37.242  *
 0.2013E-02      *  106.83          1.89   0.2126E-02          37.651             *    94.839         -1.96   0.2126E-02          33.541  *
 0.4026E-02      *  125.61          1.96   0.4140E-02          43.692             *    69.861         -2.04   0.4140E-02          24.685  *
 0.6039E-02      *  138.13          1.96   0.6154E-02          47.498             *    58.054         -2.03   0.6154E-02          20.423  *
 0.8052E-02      *  146.80          1.89   0.8168E-02          49.972             *    52.628         -1.55   0.8168E-02          18.412  *
```

14
```
 SURFACE VELOCITIES BASED ON TANGENTIAL COMPONENTS
                              UPPER SURFACE
      M              VELOCITY    ANGLE(DEG)     RHO*W

*04*  VALUE OF RHO*W IS TOO LARGE$
          CECK NAME  IFN OF CALL  ABS. LCC.
          .ERROR        00025        16615
          DENSTY        00016        10132
          TASVEL        00524        26113
          ZLCP          00018        03056

  -0                 72.179          90.00        25.845
   0.4424E-01       178.49          -12.61        55.993
   0.4919E-01       189.38          -23.06        58.913
   0.5227E-01       200.12          -30.50        61.772
```

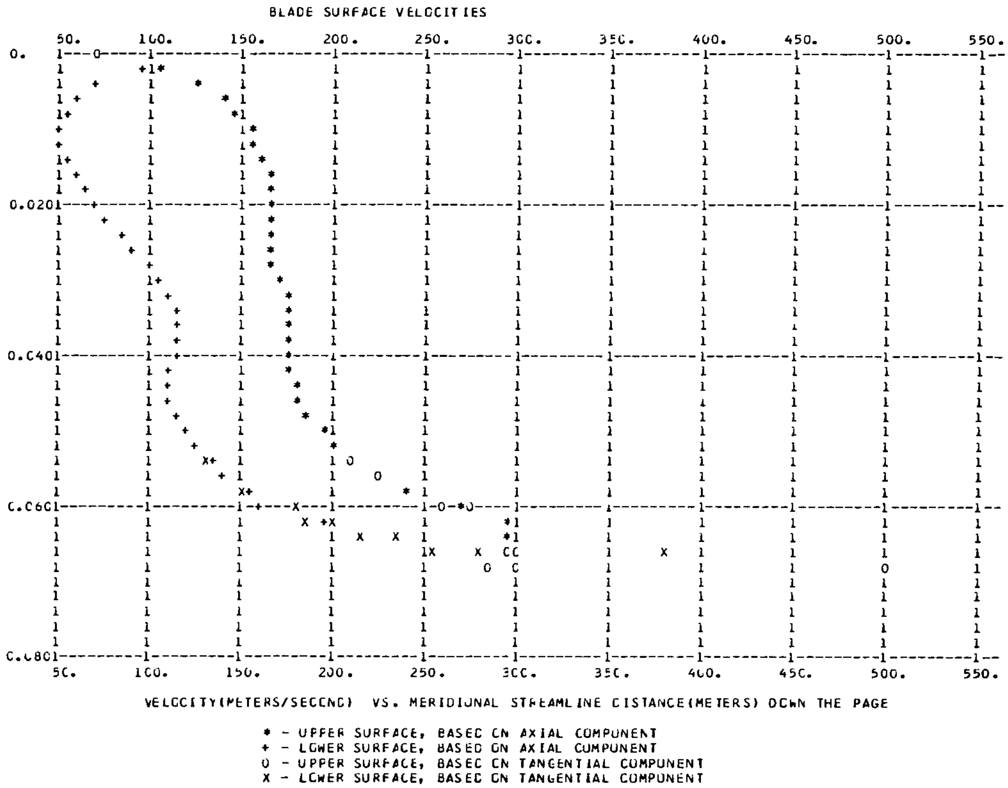
14
```
                               LOWER SURFACE
      M              VELOCITY    ANGLE(DEG)     RHO*W
   0.4230E-01       135.78           -8.87        43.533
   0.4814E-01       114.29          -22.23        36.842
   0.5133E-01       123.55          -29.54        39.647
   0.5379E-01       132.23          -34.52        42.246
```

[a]See p. 20.

TABLE IV. - Continued. SAMPLE OUTPUT

Descrip-
tion
(a)

14



[a]See p. 20.

TABLE IV. - Continued. SAMPLE OUTPUT

Description (a)

15
```
TIME =  0.6456 MIN.
```

7
```
ERROR = 0.00402374
ERROR = 0.00155175
ERROR = 0.00204642
ERROR = 0.00130510
ERROR = 0.00120662
ERROR = 0.00111038
```

9
```
TIME =  0.8267 MIN.
```

13
```
ITERATION NO.  3   MAXIMUM RELATIVE CHANGE IN DENSITY = 0.1940E-01
```

14
```
              *                   SURFACE VELOCITIES BASED ON AXIAL COMPONENTS                                            *
              *              UPPER     SURFACE                          *            LOWER     SURFACE                    *
      M       *  VELOCITY  ANGLE(DEG) SURF. LENGTH     RHO*W          *      VELOCITY  ANGLE(DEG) SURF. LENGTH     RHO*W   *
  0           *  0             90.00  0                16.975         *      0           -90.00  0                37.167   *
  0.2013E-02  *  107.31         1.89  0.2126E-02       37.814         *      94.578       -1.96  0.2126E-02       33.451   *
  0.4026E-02  *  126.22         1.96  0.4140E-02       43.893         *      69.651       -2.04  0.4140E-02       24.612   *
  0.6039E-02  *  138.86         1.96  0.6154E-02       47.734         *      57.863       -2.03  0.6154E-02       20.356   *
  0.8052E-02  *  147.64         1.89  0.8168E-02       50.240         *      52.439       -1.95  0.8168E-02       18.347   *
```

14
```
 SURFACE VELOCITIES BASED ON TANGENTIAL COMPONENTS
                             UPPER SURFACE
      M              VELOCITY   ANGLE(DEG)    RHO*W

*06*  VALUE OF RHO*W IS TOO LARGE$
             DECK NAME   IFN OF CALL   ABS. LOC.
             .ERROR        00025         16615
             DENSTY        00016         10132
             TASVEL        00524         26113
             ZCCP          00018         03056

  -0              71.826         90.00       25.721
   0.4424E-01    179.16         -12.61       56.183
   0.4919E-01    190.12         -23.06       59.116
   0.5227E-01    200.97         -30.50       62.003
   0.5460E-01    212.64         -36.16       65.044
```

14
```
                             LOWER SURFACE
      M              VELOCITY   ANGLE(DEG)    RHO*W
   0.4230E-01    135.46          -8.87       43.436
   0.4814E-01    114.07         -22.23       36.775
   0.5133E-01    123.32         -29.54       39.573
   0.5379E-01    131.96         -34.52       42.166
   0.5584E-01    139.20         -38.86       44.309
```

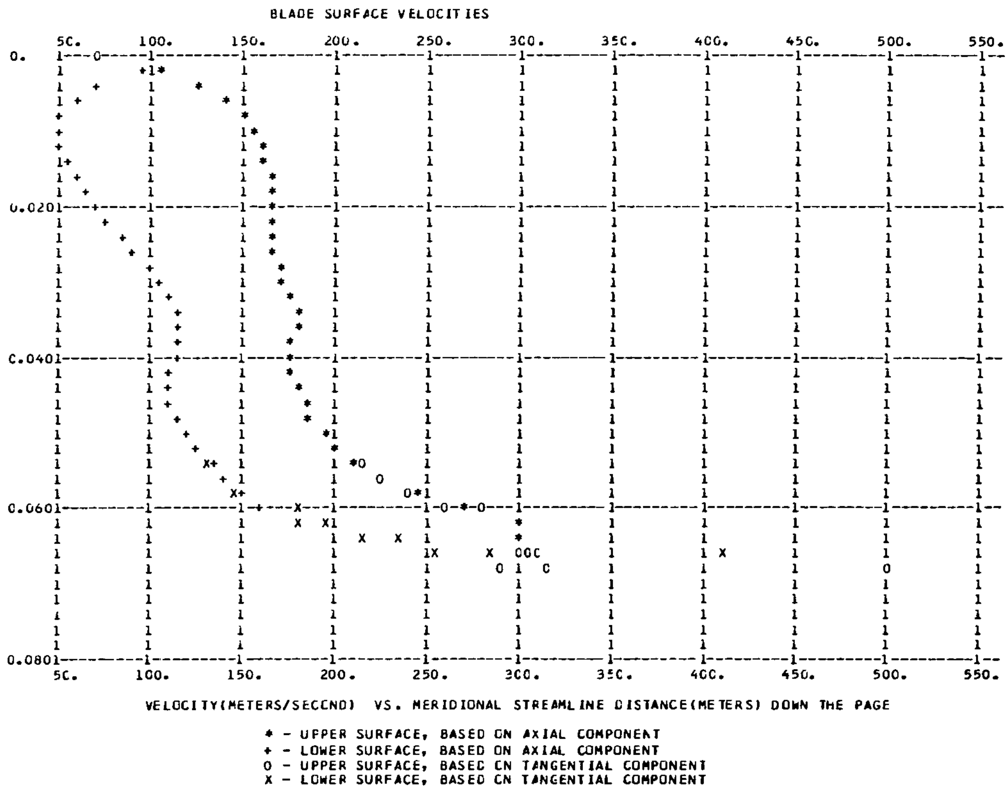
15
```
TIME =  0.8506 MIN.
```

[a] See p. 20.

TABLE IV. - Continued. SAMPLE OUTPUT

Description (a)

14



[a]See p. 20.

TABLE IV. - Continued. SAMPLE OUTPUT

Description (a)

5

IA	I	A(I,1)	A(I,2)	A(I,3)	A(I,4)	I1	I2	I3	I4	K(I)
1	1	C.	0.	0.	1.00000	8	2	0	9	-C.11565
1	2	C.	C.	C.	1.00000	1	3	0	10	-C.11565
1	3	C.	C.	0.	1.00000	2	4	0	11	-C.11565
1	4	C.	C.	C.	1.00000	3	5	0	12	-C.11565
1	5	C.	C.	C.	1.00000	4	6	0	13	-0.11565
1	6	C.	C.	C.	1.00000	5	7	0	14	-C.11565
1	7	C.	0.	C.	1.00000	6	8	C	15	-C.11565
1	8	C.	0.	C.	1.00000	7	1	0	16	-C.11565
2	9	C.15927	C.15921	0.34271	0.33881	16	10	1	17	-C.15310
2	10	C.15924	C.15924	C.34278	C.33874	9	11	2	18	C.CC617
2	11	C.15922	C.15926	C.34278	C.33874	10	12	3	19	C.CC617
2	12	C.15922	C.15926	C.34275	C.33877	11	13	4	20	C.0C616
2	13	C.15923	C.15925	C.34273	0.33879	12	14	5	21	C.CC616
2	14	C.15924	C.15924	C.34272	0.33880	13	15	6	22	0.0C616
2	15	C.15924	C.15924	C.34272	C.33880	14	16	7	23	C.CC616
2	16	0.15927	C.15922	C.34269	0.33883	15	9	8	24	C.16538
3	17	0.16472	C.16460	C.33815	C.33252	24	18	9	25	-0.15873
3	18	C.16466	C.16466	C.33820	0.33248	17	19	10	26	C.CC599
3	19	C.16462	C.16470	C.33815	0.33253	18	20	11	27	C.CC599
3	20	C.16463	C.16469	C.33811	C.33257	19	21	12	28	C.CC599

7

```
ERROR = 0.0CCC6517
ERROR = 0.0CCC3825
ERROR = C.0CCC3228
ERROR = 0.0CCC3865
ERROR = C.0CCC25C3
ERROR = 0.0CCC2364
ERROR = 0.0CCC1264
```

8

```
            STREAM FUNCTION VALUES
IA =  1
  -C.38500E45  -0.26370562  -0.14113821  -0.01622819   0.11014967   0.23700604   C.36375162   0.49012671
IA =  2
  -C.26936323  -0.1480606C  -0.C2549295   C.09941702   0.22579490   0.35265326   C.47942884   0.6C577193
IA =  3
  -C.1688972C  -0.C4583827   C.C7221C28   C.19751996   0.32436176   0.45148222   C.57844C39   C.7C525534
IA =  4
  -C.C8133291   C.C3C57560   0.1519C831   C.27847333   0.40642122   0.53403744   C.66116448   C.78868537
IA =  5
   C.C9010231   C.21272293   C.3426CC11   0.47260920   0.60083592   0.72757553   C.85547218
```

9

```
TIME =  2.7744 MIN.
```

10

STREAMLINE COORDINATES

M COORC.	STREAM FN.	THETA	STREAM FN.	THETA	STREAM FN.	THETA
-C.80517€5E-C2	-C.2CC0000	C.5432690E-01	0	0.1116599	C.2CC0000	0.1680872
	C.4CC0000	0.2244047	0.6000000	0.2812934		
-C.6C38823E-C2	-C.2CC0000	C.2044267E-C1	0	0.7873579E-C1	0.2CC0000	0.1355313
	C.40C00C0	C.1918262	0.6000000	0.2482602		
-C.4C258E2E-C2	-C	C.5039991E-01	0.2000000	C.1C78CC2	C.4CC0000	0.1640383
	C.6C000C00	C.2202608	0.8000000	C.2767515		
-C.2012941E-C2	-C	C.2620723E-01	0.2000000	C.85C6C24E-C1	C.4CCC00C	0.1410080
	0.60CC000	C.197C188	0.8000000	0.2529807		
C	C	C	0.2000000	C.6758166E-C1	0.40C0000	0.1228669

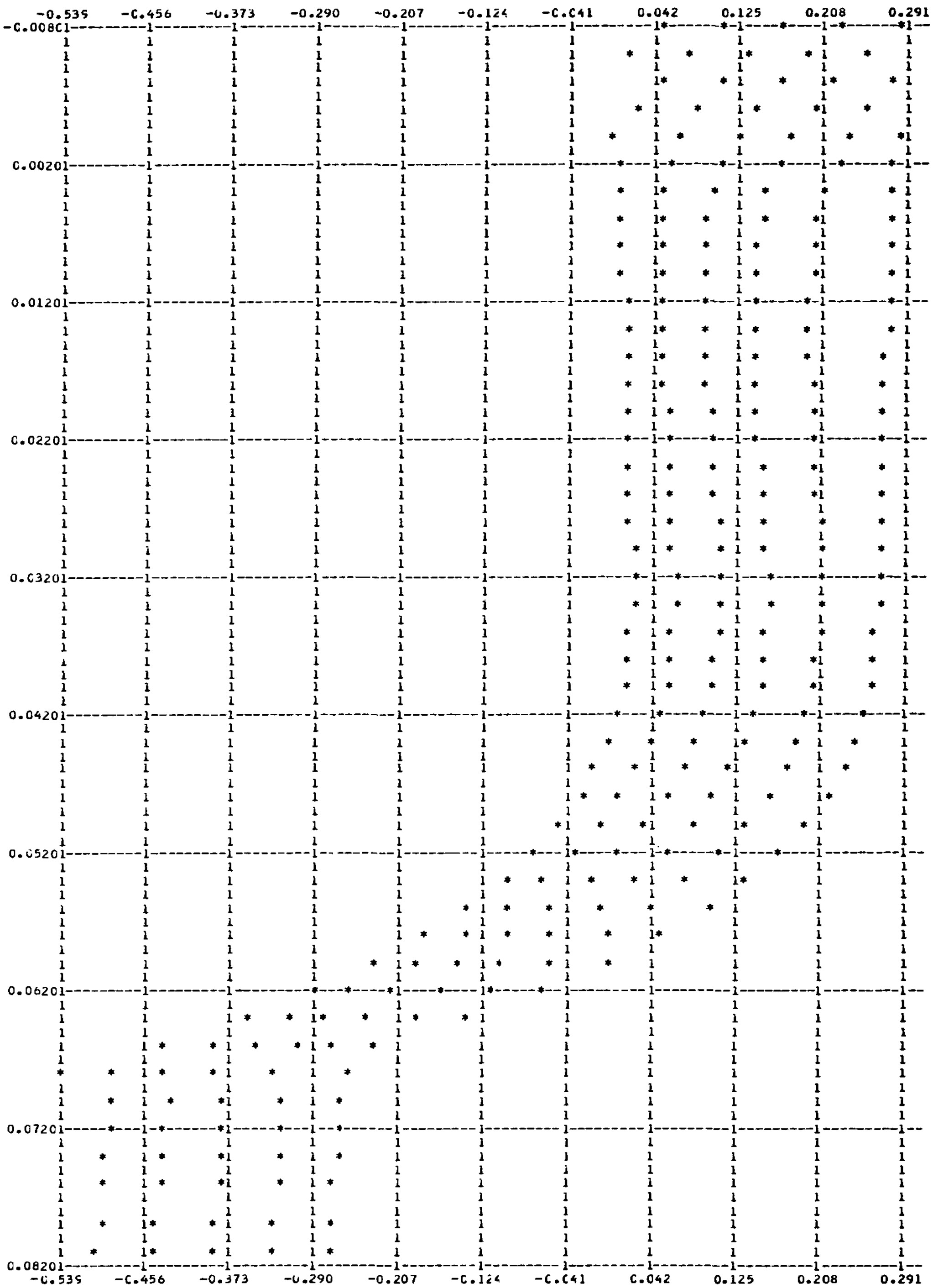
[a]See p. 20.

TABLE IV. - Continued. SAMPLE OUTPUT

Descrip-
tion
(a)

10

STREAMLINE PLCTS



STREAMLINES ARE PLCTTED WITH THETA ACROSS THE PAGE AND M DOWN THE PAGE

[a]See p. 20.

TABLE IV. - Continued. SAMPLE OUTPUT

Description (a)								
11	WM ARRAY (RHO*W-SUB-M)							
	25.907599	26.107347	26.522294	26.983462	27.174644	27.201939	27.162536	26.956273
	26.821924	27.028303	27.457889	27.935325	28.133250	28.161908	28.120715	27.907177
	26.938462	27.582843	28.347672	28.937641	29.105428	29.088145	29.069426	28.951727
	25.099002	27.521081	29.323885	30.027952	30.074642	29.925236	29.897213	30.232534
	26.289910	31.072922	31.268500	31.024394	30.616576	30.137543	32.223495	
	35.465356	33.768741	33.049494	31.902327	30.882637	29.963904	31.077170	
11	WMU ARRAY (RHO*W-SUB-M ON UPPER SURFACE)							
	16.971501	37.787557	43.857912	47.694059	50.197552	51.849903	52.922299	53.588732
	53.971274	54.158436	54.195787	54.121918	53.969568	53.835425	53.934666	54.563338
	55.519701	56.160765	56.063219	55.506120	55.010693	54.812426	54.848794	54.742220
	54.498377	54.068461	53.399481	52.632174	52.010118	51.683922	52.421168	52.255233
	47.281500	42.868861	7.3233204					
11	WML ARRAY (RHO*W-SUB-M ON LOWER SURFACE)							
	37.169581	33.435342	24.599400	20.346128	18.338659	17.639559	17.820372	18.639874
	19.943203	21.624109	23.575663	25.699009	27.898049	30.103014	32.292383	34.372876
	36.146715	37.319430	37.645979	37.362323	36.700486	35.015016	34.936288	34.317511
	34.134425	34.211736	34.474140	34.625603	34.322540	33.956013	32.302812	34.168130
	35.190435	34.166448	7.5800489					
11	WX ARRAY(RHO*W-SUB-THETA)							
	-39.624538	-39.905356	-40.041067	-39.991943	-39.922701	-39.880270	-39.848412	-39.774967
	-35.059493	-34.775723	-34.638582	-34.688226	-34.758195	-34.801071	-34.833264	-34.907481
	-30.427464	-29.084729	-28.989947	-29.231843	-29.475374	-29.602864	-29.676812	-29.923152
	-27.359055	-23.802847	-23.108671	-23.714139	-24.209203	-24.395206	-24.407707	-24.706454
	-12.940090	-16.356022	-18.165348	-19.045188	-19.242202	-18.921674	-18.543125	
11	MXU	WXU	MXU	WXU	MXU	WXU	MXU	WXU
	(M COORD. VS. RHO*W-SUB-THETA ON UPPER SURFACE)							
	-0	-25.724336	0.4423576E-01	-12.260539	0.4918970E-01	-23.145384	0.5226607E-01	-31.450157
	0.5459685E-01	-38.356696	0.5651698E-01	-44.427730	0.5817257E-01	-50.229419	0.5963962E-01	-55.820843
	0.6096762E-01	-61.084705	0.6218418E-01	-66.415494	0.6330672E-01	-68.043138	0.6434032E-01	-69.741194
	0.6529220E-01	-71.509989	0.6617128E-01	-73.904902	0.6698656E-01	-75.590648	0.6774543E-01	-77.330525
	0.6842666E-01	-113.30783	0.6844000E-01	-82.648045				
11	MXL	WXL	MXL	WXL	MXL	WXL	MXL	WXL
	(M COORD. VS. RHO*W-SUB-THETA ON LOWER SURFACE)							
	0.4229608E-01	-6.6972541	0.4613633E-01	-13.911637	0.5133148E-01	-19.516300	0.5379021E-01	-23.899858
	0.5584395E-01	-27.806569	0.5758686E-01	-31.960470	0.5908518E-01	-36.318843	0.6039599E-01	-42.909056
	0.6156307E-01	-45.404054	0.6261471E-01	-50.232773	0.6357468E-01	-55.119005	0.6445978E-01	-60.541574
	0.6528332E-01	-65.867313	0.6605394E-01	-72.164698	0.6677985E-01	-89.639418		
11	ARRAY OF RHO*W AT INTERIOR POINTS							
	47.342670	47.686801	48.028316	48.243784	48.293720	48.274024	48.225504	48.048815
	44.142764	44.044069	44.201437	44.538247	44.717020	44.768126	44.767521	44.691640
	40.638791	40.064095	40.546362	41.132562	41.423708	41.502407	41.542083	41.636492
	37.127857	36.386610	37.334982	38.262753	38.607895	38.611410	38.595071	39.043757
	29.301968	35.114754	36.162120	36.403739	36.161265	35.585126	37.177965	

[a] See p. 20.

TABLE IV. - Continued. SAMPLE OUTPUT

Description (a)

12

```
                                        VELCCITIES AT INTERICR MESH PCINTS

IA= 1  VELOCITY  ANGLE(DEG)  VELOCITY  ANGLE(DEG)  VELOCITY  ANGLE(DEG)  VELOCITY  ANGLE(DEG)  VELOCITY  ANGLE(DEG)
       130.5C    -56.82      131.5C    -56.81      132.49    -56.48      133.11    -55.99      133.26    -55.76
       133.2C    -55.7C      133.06    -55.72      132.55    -55.87

IA= 2  VELOCITY  ANGLE(DEG)  VELOCITY  ANGLE(DEG)  VELOCITY  ANGLE(DEG)  VELOCITY  ANGLE(DEG)  VELOCITY  ANGLE(DEG)
       122.26    -52.58      121.98    -52.15      122.43    -51.60      123.41    -51.15      123.93    -51.01
       124.C7    -51.02      124.07    -51.09      123.85    -51.36

IA= 3  VELOCITY  ANGLE(DEG)  VELOCITY  ANGLE(DEG)  VELOCITY  ANGLE(DEG)  VELOCITY  ANGLE(DEG)  VELOCITY  ANGLE(DEG)
       113.07    -48.48      111.47    -46.52      112.80    -45.64      114.45    -45.25      115.33    -45.36
       115.56    -45.5C      115.68    -45.59      115.95    -45.95
```

13

```
ITERATION NO. 12   MAXIMUM RELATIVE CHANGE IN DENSITY = 0.4274E-03
```

14

```
             *                    SURFACE VELCCITIES BASED ON AXIAL COMPONENTS                                             *
             *              UPPER     SURFACE                           *                  LOWER     SURFACE               *
    M        *   VELOCITY   ANGLE(DEG) SURF. LENGTH     RHO*W           *    VELOCITY   ANGLE(DEG) SURF. LENGTH     RHO*W  *
 0           *   0            90.0C     0               16.972          *    0            -90.C0    0               37.170 *
 0.2013E-02  *   1C7.29        1.89     0.2126E-02      37.808          *    94.585        -1.56    0.2126E-02      33.454 *
 0.4026E-02  *   126.19        1.96     0.4140E-02      43.884          *    69.657        -2.C4    0.4140E-02      24.614 *
 0.6039E-02  *   138.82        1.96     0.6154E-02      47.722          *    57.869        -2.C3    C.6154E-02      20.358 *
 0.8052E-02  *   147.59        1.89     C.8168E-02      50.225          *    52.445        -1.55    0.8168E-02      18.349 *
```

14

```
 SURFACE VELOCITIES BASED ON TANGENTIAL COMPCNENTS
                         UPPER SURFACE
     M             VELOCITY    ANGLE(DEG)    RHO*W

*24*  VALUE CF RHO*W IS TOO LARGE$
           DECK NAME   IFN OF CALL   ABS. LCC.
           .ERROR         00025          16615
           DENSTY         00016          1C132
           TASVEL         00524          26113
           2CCP           00018          C3056

  -0               71.836         90.CC       25.724
   0.4424E-01      179.C9        -12.61       56.165
   0.4919E-01      19C.04        -23.06       59.095
   0.5227E-01      20C.87        -30.50       61.975
   0.5460E-01      212.49        -36.16       65.006
```

14

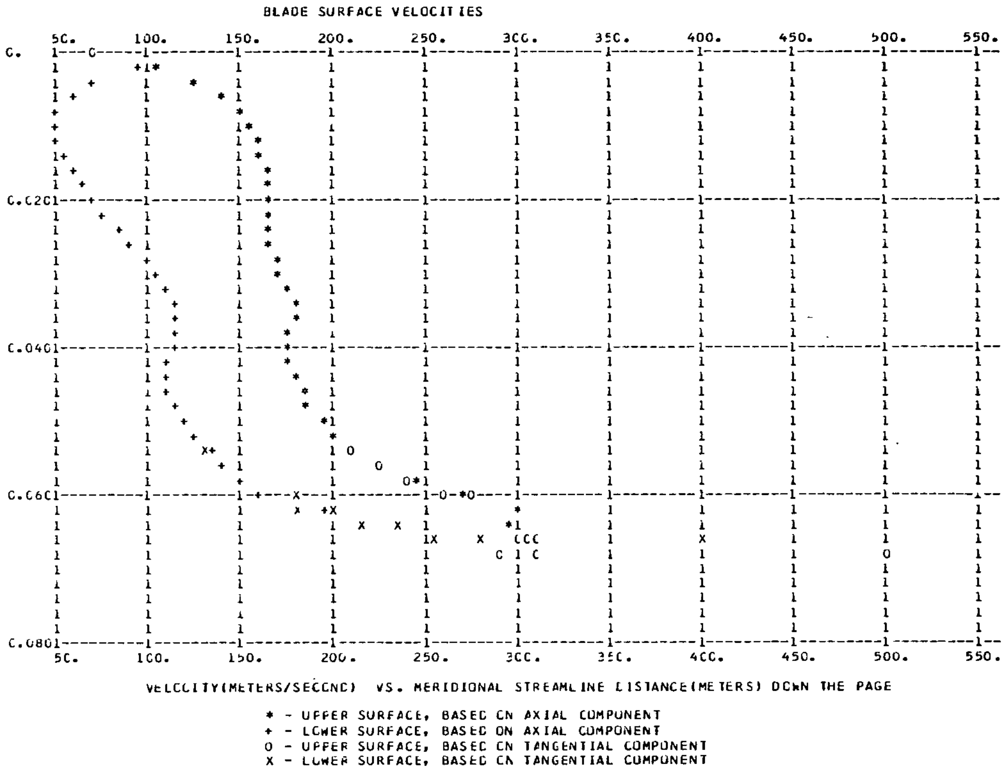
```
                         LOWER SURFACE
     M             VELOCITY    ANGLE(DEG)    RHO*W
   0.4230E-01      135.48         -8.87       43.442
   0.4814E-01      114.C8        -22.23       36.779
   0.5133E-01      123.33        -29.54       39.579
   0.5379E-01      131.98        -34.52       42.172
   0.5584E-01      139.23        -38.86       44.316
```

[a]See p. 20.

TABLE IV. - Concluded. SAMPLE OUTPUT

Descrip-
tion
(a)

14



15 { TIME = 2.5C5E MIN.

[a]See p. 20.

Error Conditions

The error message is given first for each error condition.

(1) MX, NBBI, NUSP, NLSP, NRSP, OR NINT IS TOO LARGE. If this message is printed, reduce the appropriate input values to the stated maximum value.

(2) WTFL IS TOO LARGE AT UPSTREAM BOUNDARY is printed out if WTFL (w) is greater than the choking mass flow for the upstream boundary AH. If the "continue" control card is used (see p. 86), WTFL will be cut in half, and calculations will proceed. This allows the calculation of further useful information.

(3) INU AND INL MUST BE LESS THAN 100 is printed if there are more than 100 intersections of horizontal mesh lines with either the upper or the lower blade surface. In this case NBBI should be decreased.

(4) INU NOT EQUAL TO JU OR INL NOT EQUAL TO JL. The number of intersections of horizontal mesh lines with either the upper or lower blade surface are counted by both COEF and TASVEL. If these counts do not agree, the above error message is printed out. This error is probably due to an error in the input.

(5) THE NUMBER OF UNKNOWN MESH POINTS EXCEEDS 2500, A COARSER MESH MUST BE USED is printed if there are more than 2500 interior mesh points. The actual number of interior mesh points is given. Either MX or NBBI must be reduced.

(6) VALUE OF RHO*W IS TOO LARGE is printed if the value of ρW at some point is so large that there is no solution for the value of ρ and W. Decreasing WTFL (w) sufficiently eliminates this condition. However, it may be desired to continue the calculations. If so, the "continue" control card (see p. 86) is used. This may permit an approximate solution to be obtained, which would be valid at other points. In some cases the value of ρW is reduced at the point in question during later iterations, resulting in a valid final solution.

(7) OUT OF RANGE Z = x.xxx is printed if SPLINT is used for extrapolation. Also, the input and output for SPLINT is printed. SPLINT is normally used for interpolation, but may be used for extrapolation in some cases. Calculations proceed normally after this print out.

(8) CAUTION-HB* RM(MXBO) LESS THAN RO LESS THAN HA MAY NOT GIVE CORRECT RESULTS is printed if the internally calculated values of HA and HB are such that

$$HB^* RM(MXBO) < RO < HA$$

Decreasing NBBI or increasing the difference MXBO - MXBI alleviates this condition.

PROGRAM PROCEDURE

The program is segmented into five main parts - the subroutines INPUT, COEF, SOR, SLAXVL, and TASVEL called by the main program 2DCP. In addition there are several other subroutines. All the subroutines and their relation are depicted in figure 13. All information which must be transmitted between the five main subroutines is placed in COMMON. The program can handle up to 2500 mesh points on the IBM 2-7094-7044 direct coupled system with a 32 768 word core.

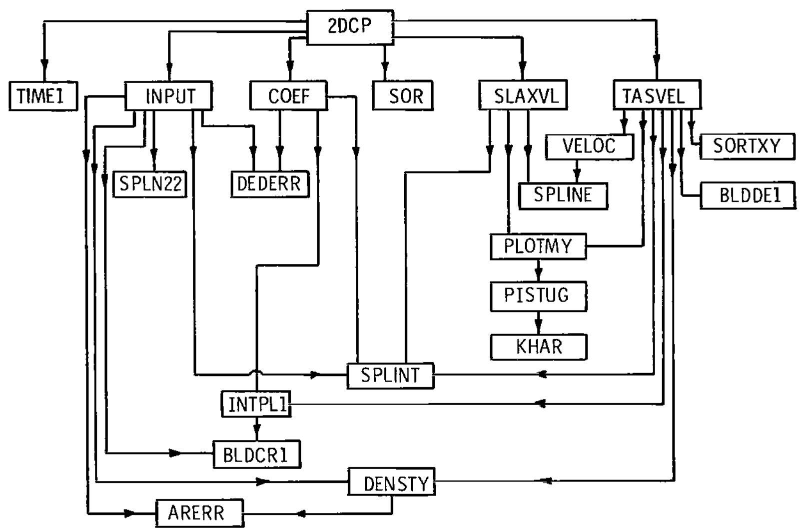


Figure 13. - Logical relation of subroutines.

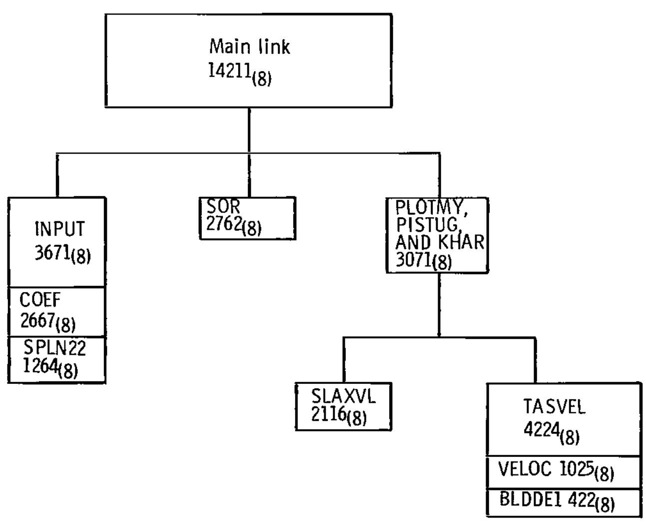


Figure 14. - Arrangement for overlay, showing octal storage requirements.

To be able to handle 2500 mesh points an overlay arrangement is used as indicated in figure 14. All subroutines not shown are in the main link. The total program storage requirements is $74303_{(8)}$ of which $47106_{(8)}$ is in COMMON. The system storage requirement for our computer is $2764_{(8)}$ and unused storage is $511_{(8)}$. If there is a storage problem on the user's computer, the maximum number of mesh points should be reduced.

The first segment of the program is INPUT. This subroutine reads all input data cards, calculates basic constants and useful information, and calculates the blade coordinates on mesh lines. INPUT is called only once for each case. The next subroutine is COEF which calculates the entries of the matrix A and the vector $\underline{k}$ of equation (A7). These coefficients must be recalculated for each outer iteration. The subroutine SOR estimates an optimum overrelaxation parameter Ω on the first call if it is not given as input. The same value of Ω is used for each outer iteration. SOR then finds the linear solution to equation (A7) with fixed coefficients by successive overrelaxation. Then subroutine SLAXVL calculates the streamline locations and ρW_m and plots the streamline locations if desired. Finally, the subroutine TASVEL calculates ρW_θ, velocity magnitudes and direction, densities for next outer iteration, the surface velocities based on axial velocities and on tangential velocities, and plots the surface velocities.

Conventions Used in Program

For convenience, a number of conventions are used in naming variables and assigning subscripts. First, several pairs of variables are spelled the same except for one letter, which is U in one case and L in the other. The U signifies the upper surface BC, and L the lower surface CF. Another practice is to use the letters I and O in a similar manner, where I refers to the inlet or the region ABGH, and O refers to the outlet or region CDEF. Thus, ALUO refers to the angle on the upper blade surface near the outlet, or near point C (fig. 10, p. 15).

The variable I is used to number all the mesh points starting with I = 1 at A and proceeding along the vertical mesh lines and moving to the right to the next line after the end of each vertical line and ending with I = NXN at the last mesh point near E. The mesh spacing in the m direction is labeled HA, and the spacing in the θ direction is HB.

The techniques used in the program and correspondence to the mathematical equations are described briefly. Each subroutine is described separately, first the five segments of the main program, followed by descriptions of each of the remaining subroutines. The various segments of the subroutines are labeled by comment cards, which generally correspond to the headings in the following descriptions.

Subroutine INPUT

Input. - The first step is to read all input cards for a particular case. A detailed description of the input required is given in the section Instructions for Preparing Input (p. 14). All input data are given as the first output.

Calculation of constants and initialization. - After all input has been read in, the various constants needed in the program are calculated and certain quantities are initialized. The input θ-coordinates for the lower blade surface (XSPL) are increased by PITCH to define the blade passage. Certain convergence tolerances are specified at this point. The arrays RM and BE of the quantities r and b at each vertical mesh line are calculated by subroutine SPLINT (using cubic spline interpolation). Prerotation λ is calculated by an interative procedure. Also, the useful quantities of upstream and downstream free-stream velocity, maximum values of the mass flow parameter ρW, relative critical velocity, and free-stream flow angle corrected to leading and trailing edges are all calculated and printed. All density arrays are initialized to ρ_{in}'(RHOIP).

Calculation of mesh coordinates along boundary. - The θ-coordinates of boundaries BC and GF at each vertical grid line are calculated by BLDCR1 and stored in the arrays XU and XL. BLDCR1 requires as input the first and second derivatives at each spline point of the cubic spline curves describing the blade surface. These values are calculated by SPLN22. The first and last points of the spline curves are determined by the angle of tangency to the leading- and trailing-edge radii. Therefore, these points are not specified as input, but are computed by this section of the program.

Subroutine COEF

The coefficients of $\underline{u}$ in equations (A2) to (A6) (elements of matrix A in eq. (A7)) are computed at the same time as the constants (components of $\underline{k}$ in eq. (A7)). Between the blades it is necessary to compute values for h_3 and h_4 at some mesh points adjacent to the boundary. These values are calculated by INTPL. Also it is necessary to calculate values for b_3 and b_4 along the boundary. These values are calculated by SPLINT and stored in BEU and BEL. These arrays are ordered by increasing m-coordinate. This is not the same order as the order of mesh points (i.e., increasing I subscript) which necessitates some juggling of the subscripts of the BEU and BEL arrays. The subscripts INL and INU increase with I, and ITP is the correct subscript for the BEU or BEL array. A similar situation holds with the RHOLT and RHOUT arrays for the values of ρ_3 and ρ_4 from the previous iteration.

Near the trailing edge, a special situation may arise, as illustrated in figure 15. Here it should be noted that the blade intersects the mesh line twice between two adjacent

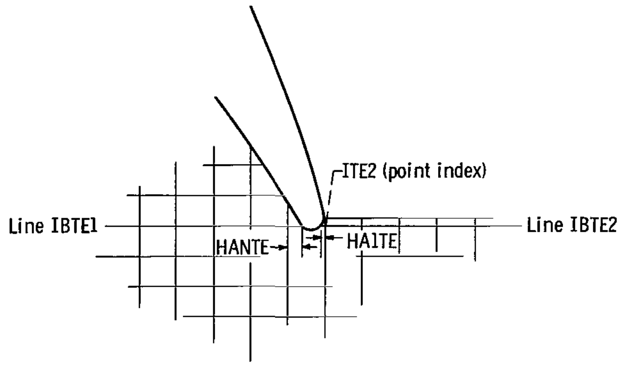


Figure 15. - Special case near trailing edge.

mesh points due to the small trailing-edge radius. This situation would not be detected by the program in the normal procedure, which leads to a large error in the velocity calculation at the next to the last vertical grid line on the lower blade surface. Therefore, a special check is made for the two vertical mesh lines involved at statements 120 and 130, and if this situation occurs, the proper values of H3 or H4 are calculated. Also, the number of the horizontal mesh line is stored in IBTE2 and IBTE1, and this information is used in calculating the tangential velocity components. If the values of HA and HB are such that two horizontal mesh lines could intersect the trailing-edge radius, the message for error condition (8) (p. 36) is printed out.

Subroutine SOR

Estimation of value of optimum overrelaxation factor. - If a value of $W \geq 1$ is given as input, it is used for the overrelaxation factor. Otherwise a value is estimated in the first iteration by using equation (B3) of reference 5 to estimate the value of $\rho^2(B) = \rho(L_1)$ and equation (B1) of reference 5 to obtain the corresponding value of W (see appendix B of ref. 5). Equation (A8) is used to calculate $\underline{u}^{m+1}$ from $\underline{u}^m$ for equation (B3), with $\omega = 1$ and $\underline{k} = \underline{0}$. To start, $u_i^o = 1$ for all i. Equation (A8) becomes

$$u_i^{m+1} = - \sum_{j=1}^{i-1} a_{ij} u_j^{m+1} - \sum_{j=i+1}^{n} a_{ij} u_j^m \tag{7}$$

In the program, i is replaced by I directly. For each i, there are only four values of j for which a_{ij} is nonzero, which are the negative values of the coefficients A(I, 1), A(I, 2), A(I, 3), and A(I, 4). The value of j is determined by the index of the proper neighboring point. These indexes are named I1, I2, I3, and I4, and are defined so $u^m_{I,1}$ has the coefficient A(I, 1), and similarly for the other coefficients. After the values of the indexes are computed, equation (7) is used to compute $\underline{u}^{m+1}$ from $\underline{u}^m$. Then, the minimum and maximum values of the ratio u_i^{m+1}/u_i^m are calculated and given the names LMIN and LMAX, respectively. After convergence, the optimum value of the overrelaxation factor Ω can be calculated from equation (B1) of reference 5, since $\rho^2(B) = LMAX$.

Calculation of initial solution estimate. - For the first outer iteration an initial solution estimate must be made. This is done by assuming $u = 0$ along ABCD and $u = 1$ along EFGH and assuming linear variation along vertical mesh lines. On subsequent iterations, the previous solution is used as the initial solution estimate.

Solution of matrix equation by SOR. - With a value of Ω either as input or estimated by the program, equation (A8) can be used iteratively to calculate a sequence $\{\underline{u}^m\}$ that will converge rapidly to a solution of equation (A7). The indexes i and j and the correspondence of a_{ij} and u_j^m to the program variables is the same as described previously for estimating the optimum overrelaxation factor. During each iteration the maximum change of the stream function is calculated. When this maximum change is reduced below TOLER, set equal to 10^{-6}, the iteration is stopped, and the current estimate of the stream function is accepted as the solution.

Subroutine SLAXVL

Calculation of streamline locations and $\partial u/\partial\theta$. - Along most vertical mesh lines, the stream function is a one-to-one function of the distance in the θ-direction. Therefore, the θ-coordinate is considered to be a function of the value of the stream function, and the value of θ at a given value of the stream function can be obtained by cubic spline interpolation (SPLINT). At the same time, $\partial u/\partial\theta$ is computed along the same mesh line, estimating the derivative at each mesh point by use of the cubic spline (SPLINE). The derivative $\partial u/\partial\theta$ at unknown mesh points is stored in the array WM and $\partial u/\partial\theta$ along the blade surfaces is stored in WMU and WML. These calculations are performed in three sections, as noted by comment cards: (1) upstream, (2) between the blades, and (3) downstream.

Plotting of streamlines. - The streamlines can be plotted to give a rough idea of their locations. These locations are particularly helpful in disclosing quickly any errors of input. The plotting printout is done by PLOTMY, which, with the necessary further subroutines PISTUG and KHAR, is described completely with FORTRAN IV listing in reference 12. The plotting can be omitted by removing statements following statement 420 up to and including statement 470.

Calculation of ρW_m. - The product ρW_m is calculated by multiplying $\partial u/\partial\theta$ by w/br (using eq. (3)). The values of ρW_m are stored in WM, WMU, and WML.

Subroutine TASVEL

Calculation of $\partial u/\partial m$. - The tangential velocity component is calculated from $(\partial u/\partial m) = (-b\rho W_\theta/w)$ by considering each horizontal mesh line. The fact that the various horizontal mesh lines start and end at various places complicates this process. To simplify the procedure, upstream and downstream ends of each mesh line are considered separately. At the upstream end, there are three possibilities: (1) the line starts at AH(IA = 1), (2) the line starts on lower surface of blade, or (3) the line starts on the upper surface of the blade. Similarly, at the downstream end there are three possibilities. For convenience, case numbers are assigned to the various possibilities as follows:

Case	Starts	Ends
1	AH	Upper surface
2	AH	DE
3	AH	Lower surface
4	Lower surface	DE or lower surface
5	Upper surface	DE or lower surface

As mentioned in the description of COEF, a special situation often arises where a horizontal mesh line is intersected twice between two adjacent mesh points, as illustrated in figure 15 (p. 40). When this occurs, the index of the horizontal mesh line is stored in IBTE2 and IBTE1. Under cases 2, 4, or 5, if IB = IBTE2, the special case arises, and previously calculated information is used for this line to the left of the trailing edge. After all other tangential velocities have been calculated, tangential velocities are calculated for the remainder of this line (IB = IBTE1) to the right of the trailing edge.

TASVEL calculates the necessary information about the two end points of each horizontal mesh line by using INTPL to calculate the mesh spacing at the end points. Then VELOC calculates $\partial u/\partial m$ at each point along the line by using SPLINE to calculate the actual derivatives. The derivatives, $\partial u/\partial m$ at unknown mesh points are stored in the array WX, and $\partial u/\partial m$ along the blade surfaces is stored in WXU and WXL, with the corresponding m-coordinates stored in MXU and MXL. The values of WXL and MXL are rearranged in increasing order of MXL by SORTXY.

Calculation of ρW_θ. - The product ρW_θ is calculated by multiplying $\partial u/\partial m$ by -w/b (using eq. (2)). The values of ρW_θ are stored in WX, WXU, and WXL for interior mesh points, upper surface of blade, and lower surface of blade, respectively.

Calculation of mass flow parameter ρW and angles at interior points. - At each interior point, ρW is calculated by $\rho W = \sqrt{\left(\rho W_m\right)^2 + \left(\rho W_\theta\right)^2}$, and the angle β is calculated by $\tan \beta = \rho W_\theta / \rho W_m$.

Calculation of velocity and density at each interior point. - A value of ρW determines a unique subsonic velocity W and corresponding density ρ. These are calculated at each interior point by subroutine DENSTY. Also the relative change in ρ at each point from the previous iteration is calculated, and if the maximum relative change in density is less than 0.001, the outer iteration is considered to have converged sufficiently and the calculation terminates after the final printouts.

The derivative $\partial u/\partial m$ at unknown mesh points is stored in the array WX.

The derivative $\partial u/\partial m$ along the blade surfaces is stored in WXU and WXL, with the corresponding m-coordinates stored in MXU and MXL.

Calculation of surface velocity based on ρW_m. - At each vertical mesh line ρW at the blade surface is calculated by

$$\rho W = \frac{\rho W_m}{\cos \beta} = \rho W_m \sqrt{1 + \left(r \frac{d\theta}{dm}\right)^2}$$

The derivative $d\theta/dm$ of the blade surface at each vertical mesh line is computed by BLDCR1 at the same time the blade coordinates XU and XL are computed, and is stored in DXDZU and DXDZL. The surface velocity is then calculated from the value of ρW by DENSTY. The surface velocity based on ρW_m is more accurate at small values of β and would not be expected to be accurate for $|\beta| > 60^\circ$. The blade surface length is calculated for convenience using equation (B17).

Calculation of surface velocity based on ρW_θ. - At each horizontal mesh line ρW at the blade surface is calculated by

$$\rho W = \frac{\rho W_\theta}{\sin \beta} = \rho W_\theta \sqrt{1 + \frac{1}{\left(r \frac{d\theta}{dm}\right)^2}}$$

The derivative $d\theta/dm$ of the blade surface at each horizontal mesh line is computed by BLDDE1 and is stored in DTDMU and DTDML. The surface velocity is then calculated from the value of ρW by DENSTY. The surface velocity based on ρW_θ is more accurate when $|\beta|$ is close to 90° and would not be expected to be accurate for $|\beta| < 30^\circ$.

Plotting of surface velocities. - If desired, the surface velocities are plotted using a printer plotter. The velocities are plotted using different symbols for upper and lower surface and for velocities based on meridional components or on tangential components. Velocities based on meridional velocity components are plotted if $|\beta| \leq 60^{\circ}$ and velocities based on tangential velocity components are plotted if $|\beta| \geq 30^{\circ}$. Plotting is done by PLOTMY, which is described in reference 12.

Internal Variables for INPUT, COEF, SOR, SLAXVL, and TASVEL

A	array of coefficients of u which are elements of matrix A in eq. (A7)
A12, A34	a_{12}, a_{34} in eq. (A2)
AA	temporary storage
AAA	array used for temporary storage
AΠ	a_o in eq. (A2)
B	temporary storage
B12, B34	b_{12}, b_{34} in eq. (A2)
BE3, BE4	b_3, b_4 in eq. (A2)
BE	array of values of b at vertical mesh lines
BEL(BEU)	array of values of b at horizontal mesh lines on lower (upper) blade surface
BETA	array of values of β at interior mesh points
BETAL(BETAU)	array of values of β on lower (upper) blade surface
BTAIN	free-stream angle β at blade leading edge based on β_{in}, calculated by eq. (B13)
BTAOUT	free-stream angle β at blade trailing edge based on β_{out}, calculated by eq. (B15)
CASE	number (integer) of case in calculating tangential velocity components
CHANGE	change in value of stream function at a particular point when using SOR iteration
CP	c_p
CPTIP	$2\, c_p T'_{in}$

DELINT	increment of stream function for which streamline locations are to be calculated
DTDML(DTDMU)	array of $d\theta/dm$ at horizontal mesh lines for lower (upper) blade surface
DTLR	tolerance for mesh points near boundary mesh point (If a mesh point is closer than DTLR to the boundary, the boundary is considered to go through the mesh point. The program uses DTLR = 0.001 HB.)
DX	θ plotting increment for streamline plot
DXDZL(DXDZU)	array of values of slope of lower (upper) blade surface at each vertical mesh line
DZ	m plotting increment for streamline plot
EML(EMU)	array of second derivatives of spline curve at each spline point for lower (upper) blade surface, calculated by SPLN22
ERROR	maximum absolute value of the change in u for an overrelaxation iteration
EXPON	$1/(\gamma - 1)$
FIRST	value (integer) of I at lowest mesh point for given vertical mesh line
H1, H2, H3, H4	h_1, h_2, h_3, h_4 (see fig. 18 in appendix A)
HA	basic mesh space in meridional (m) direction
HAMRO	HA-RO
HB	basic mesh space in blade-to-blade (θ) direction
HA1	length of first mesh space along horizontal mesh line
HA1TE	HA1 for special case shown in fig. 15 for line segment to right of trailing edge
HAN	length of last mesh space along horizontal mesh line
HANTE	HAN for special case shown in fig. 15 for line segment to left of trailing edge
HL	θ distance between EF on boundary and first mesh line below (fig. 12)
HU	θ distance between CD on boundary and first mesh line above (fig. 12)

I	index of mesh point
I1(I2,I3,I4)	index of mesh point located at 1 (2, 3, 4) in fig. 18 with I at 0
IA	index of mesh line in meridional (m) direction
IB	index of mesh line in blade-to-blade (θ) direction
IBDL	difference between NL for a vertical mesh line and the next one
IBDU	difference between NU for a vertical mesh line and the previous one
IBTE1, IBTE2	indexes of special mesh lines shown in fig. 15
IA1	index of first mesh point along horizontal mesh line
IAN	index of last mesh point along horizontal mesh line
IL(IU)	array of indexes of highest (lowest) mesh point for each vertical mesh line
INL(INU)	index counting number of intersections of horizontal mesh lines with lower (upper) surface
ITE2	index of mesh point indicated in fig. 15
ITER	outer iteration number
ITERA	temporary storage of outer iteration number
ITP	temporary index
J, JI, JB	temporary indexes
JL(JU)	number of points where horizontal mesh line intersects lower (upper) blade surface
JUM1	JU-1
K	array (real) of constants that is vector $\underline{k}$ in eq. (A7)
K1, K2, K3, K4, K5	code variables (real) used in determining values of coefficients A(I, J) and constants K(I)
KK1	code to specify whether first point of horizontal mesh line is on AH(KK1 = 0) or upper blade surface (KK1 = 0) or lower blade surface (KK1 = 1)
KN	same as KK1, but for last point
KKK	array containing information used in plotting subroutine PLOTMY
LAMBDA	λ

LAST	value of I at highest mesh point for given vertical mesh line
LMAX	upper bound (real) for $\rho(L_1)$ from eq. (B2) of ref. 5
LMIN	lower bound (real) for $\rho(L_1)$ from eq. (B2) of ref. 5
MPL	array of m-coordinates of vertical mesh lines
MXBIM1	MXBI - 1
MXBIP1	MXBO + 1
MXBOM1	MXBO - 1
MXBOP1	MXBO + 1
MXL(MXU)	array of m-coordinates of intersections of horizontal mesh lines with lower (upper) blade surface
NBB	number of mesh points along vertical mesh line
NBBO	number of mesh lines above mesh line AB for first mesh line below EF (may be negative)
NBUO	number of mesh lines above mesh line AB for line CD (usually negative, unless STGR is positive)
NCH	number of vertical mesh lines in length of blade
NL	array of number of mesh points on vertical mesh line above line AB (may be negative)
NP1(NP2)	number of plotted upper (lower) blade surface velocities based on meridional components
NP3(NP4)	number of plotted upper (lower) blade surface velocities based on tangential velocities
NSP	number of mesh points plus boundary points along vertical mesh line
NU	array; on vertical mesh line IA, the mesh point nearest the upper blade surface is NU(IA) mesh points above line AB (NU(IA) may be negative)
NUTEMP	temporary storage of NULAKI
P	array of input information for plotting subroutine PLOTMY
PITCH	2π/NBL
RATIO	value of u_i^{m+1}/u_i^m for use in eqs. (B2) and (B3) of ref. 5

RELER	maximum relative change in density at interior mesh points, between two outer iterations
RHO1, RHO2, RHO3, RHO4	$\rho 1$, $\rho 2$, $\rho 3$, $\rho 4$ in eq. (A2)
RHO	array of densities ρ at interior mesh points
RHOL(RHOU)	array of densities ρ at vertical mesh lines for lower (upper) blade surface
RHOLT(RHOUT)	array of densities ρ at hozizontal mesh lines for lower (upper) blade surface
RHONEW	newly calculated estimate of ρ
RHOT	temporary storage of a value of ρ
RHOVI	$(\rho W)_{in}$
RHOWMI	maximum value of ρW along AH
RHOWMO	maximum value ρW along DE
RM	array of values of r at each vertical mesh line
RML(RMU)	array of values of r at the intersections of horizontal mesh lines with the lower (upper) blade surface
SAL	array of values of $\sin \alpha = (dr/dm)$ at each vertical mesh line
SL	array of streamline coordinates for input data to the plotting subroutine PLOTMY
SLLPE(SLUPE)	array of slopes of spline curve at each spline point for lower (upper) blade surface, calculated by SPLN22
SRW	code (integer) variable that will cause certain subroutines to write out data useful for debugging:

SRW = 13	SPLINE will write input and output data
= 16	SPLINT will write input and output data
= 18	SPLN22 will write input and output data
= 19	BLDCR1 will write out blade coordinates and slopes at each mesh line
= 19	INTPL1 will write out pertinent data for each iteration
= 20	BLDDE1 will write out m-coordinates and slopes

TANTH	$\tan \beta$ at unknown mesh points
TANTHL(TANTHU)	$\tan \beta$ along lower (upper) blade surface

TBI	$\tan\ \beta_{in}$
TBO	$\tan\ \beta_{out}$
TGROG	$2\gamma R/(\gamma + 1)$
TOLER	when the maximum absolute value of the change in the stream function is less than TOLER, the SOR iteration is considered converged; value used in the program is 10^{-6}
TPP	T''
TTIP	T/T'_{in}
TWL	$2\omega\lambda$
TWLMR	$2\omega\lambda - (\omega r)^2$
TWW	$2\omega/w$
U	array of estimated values of stream function or of eigenvector associated with spectral radius of L_1, $\rho(L_1)$ as estimated by power method (ref. 5)
UINT	array of values of stream function for which it is desired to obtain interpolated values of θ-coordinate
UNEW	new value of eigenvector estimate at single point, as calculated by eq. (7)
UNI	$(\partial u/\partial\eta)_{in}$ (eqs. (5) and (A3))
UNO	$(\partial u/\partial\eta)_{out}$ (eqs. (6) and (A4))
USP	array of values of stream function along vertical mesh line, including boundary points
V	array of relative velocities W at unknown mesh points, also used for storing values of ρW
VEL	temporary storage, W
VI	W_{in}
VO	W_{out}
VTOL	in calculating W from ρW by eq. (B6), the procedure is considered converged when the relative change in W is less than VTOL; program uses a value of 10^{-4}
WCRI	W''_{cr} at B, fig. 4
WCRO	W''_{cr} at C, fig. 4

WL(WU)	array of velocities along lower (upper) blade surface
WM	array of ρW_m at interior mesh points
WMAX	upper bound for optimum Ω from eqs. (B1) and (B2) of ref. 5
WMAX1	temporary storage for WMAX
WMIN	lower bound for optimum Ω from eqs. (B1) and (B2) of ref. 5
WML(WMU)	array of ρW_m where vertical mesh lines intersect lower (upper) blade surface
WR	tolerance specified for the calculation of overrelaxation factor Ω, program uses 10^{-5}
WX	array of ρW_θ at interior mesh points
WXL(WXU)	array of ρW_θ where horizontal mesh lines intersect lower (upper) blade surface
X1	value of θ for which INTPL1 is to compute H3 or H4
XBB	array of θ-coordinates associated with array USP
XDOWN	array of m-coordinates where surface velocities are plotted
XFACT	scaling exponent for streamline plot
XINT	array of interpolated θ-coordinates calculated by SPLINT and corresponding to array UINT
XL(XU)	array of θ-coordinates of lower (upper) surface of blade at each vertical mesh line
XMAX	maximum value of θ in streamline plot
XMIN	minimum value of θ in streamline plot
YACROS	array of surface velocities plotted
ZINT	argument of BLDCR1, not used in main program
ZFACT	scaling exponent for streamline plot
ZMIN	minimum value of m for streamline plot

MAIN PROGRAM

```
      COMMON SRW,ITER
      SRW = 0
      CALL TIME1(T1)
   10 CALL INPUT
   20 CALL COEF(NXN)
      IF(NXN.GT.2500) GO TO 10
      CALL SOR
      CALL TIME1(T2)
      TIME = (T2-T1)/3600.
      WRITE (6,1000) TIME
      CALL SLAXVL
      CALL TASVEL
      CALL TIME1(T2)
      TIME = (T2-T1)/3600.
      WRITE (6,1000) TIME
      IF(ITER.EQ.0) GO TO 10
      GO TO 20
 1000 FORMAT (1HL,7HTIME = ,F7.4,5H MIN. )
      END
```

```
      SUBROUTINE INPUT
C
C     IA IS AXIAL INDEX
C     IB IS BLADE-TO-BLADE INDEX
C     I IS OVERALL INDEX
C     HA IS BASIC AXIAL INCREMENT
C     HB IS BASIC BLADE-TO-BLADE INCREMENT
C
C     THE FOLLOWING CODE IS USED FOR OUTPUT OPTIONS
C           1 - LAST ITERATION ONLY (AFTER CONVERGENCE)
C           2 - FIRST AND LAST ITERATION
C           3 - ALL ITERATIONS
C
      REAL K,K1,K2,K3,K4,K5,LMAX,LMIN,LAMBDA,MU,ML,MR,MXU,MXL,MPL
      INTEGER SRW,FIRST,CASE,BLDATA,ERPRT,STRFN,SLCRD,SLPLT,ARPRT,SURVEL
      COMMON SRW,ITER,GAM,AR,TIP,RHOIP,WTFL,OMEGA,LAMBDA,CP,EXPON,PITCH,
     1   CHORD,STGR,BETAI,BETAO,DTLR,RI,ALUI,ALLI,RO,ALUO,ALLO,
     2   MXBI,MXBO,MX,NBBI,NUSP,NLSP,NRSP,NINT,VTOL,
     3   BLDATA,NULAKI,ERPRT,STRFN,SLCRD,ARPRT,INTVEL,SURVEL,
     4   MU(50),XSPL(50),ML(50),XSPL(50),MR(50),RMSP(50),BESP(50),
     5   W,WR,TOLER,BDA,BDD,U(2500),A(2500,4),K(2500),RHO(2500),
     6   DXDZU(100),DXDZL(100),SLOPE(50),EMU(50),SLLPE(50),EML(50),
     7   RM(100),BE(100), SAL(100),XU(100),XL(100),RMU(100),RML(100),
     8   NU(100),NL(100),UINT(11),XINT(11),MPL(100),
     9   HA,HB,NXN,MXBIM1,MXBOP1,JU,JL,HU,HL,NBBC,NBUC,NCH,
     1   IBTE1,IBTE2,ITE2,HALTE,HANTE,UNI,LNC,TWW,ITERA,
     2   RHOU(100),RHOL(100),RHOUT(100),RHOLT(100),BEU(100),BEL(100),
     3   AAA(100)
```

```
C    THE FOLLOWING VARIABLES ARE ALL IN COMMON BY EQUIVALENCE
      DIMENSION WM(2500),WX(2500),V(2500), BETA(2500),SL(1100)
      EQUIVALENCE (A(1,1),WM(1)),(A(1,2),WX(1)),(A(1,3),V(1)),
     1   (A(1,4), BETA(1)),(K(1401),SL(1))
   10 WRITE (6,999)
      ITER = 0
      READ (5,1010) GAM,AR,TIP,RHOIP,WTFL,OMEGA,W
      WRITE (6,1100)
      WRITE(6,1020) GAM,AR,TIP,RHOIP,WTFL,OMEGA,W
      WRITE (6,1110)
      READ (5,1010) CHORD,STGR,BETAI,BETAO
      WRITE(6,1020) CHORD,STGR,BETAI,BETAO
      WRITE (6,1120)
      READ (5,1010) RI,ALUI,ALLI,RO,ALUO,ALLO
      WRITE(6,1020) RI,ALUI,ALLI,RO,ALUO,ALLO
      WRITE (6,1130)
      READ (5,1000) MXBI,MXBO,MX,NBBI,NUSP,NLSP,NRSP,NBL,NINT
      WRITE(6,1000) MXBI,MXBO,MX,NBBI,NUSP,NLSP,NRSP,NBL,NINT
      IF(MX.GT.100.OR.NBBI.GT.50.OR.NUSP.GT.50.OR.NLSP.GT.50.OR.NRSP.GT.
     1   50.OR.NINT.GT.10) GO TO 300
      WRITE (6,1140)
      READ (5,1010) (  MU(IA),IA=1,NUSP)
      WRITE(6,1020) (  MU(IA),IA=1,NUSP)
      WRITE (6,1150)
      READ (5,1010) (XSPU(IA),IA=1,NUSP)
      WRITE(6,1020) (XSPU(IA),IA=1,NUSP)
      WRITE (6,1160)
      READ (5,1010) (  ML(IA),IA=1,NLSP)
      WRITE(6,1020) (  ML(IA),IA=1,NLSP)
      WRITE (6,1170)
      READ (5,1010) (XSPL(IA),IA=1,NLSP)
      WRITE(6,1020) (XSPL(IA),IA=1,NLSP)
      WRITE (6,1180)
      READ (5,1010) (MR(IA),IA=1,NRSP)
      WRITE(6,1020) (MR(IA),IA=1,NRSP)
      WRITE (6,1190)
      READ (5,1010) (RMSP(IA),IA=1,NRSP)
      WRITE(6,1020) (RMSP(IA),IA=1,NRSP)
      WRITE (6,1200)
      READ (5,1010) (BESP(IA),IA=1,NRSP)
      WRITE(6,1020) (BESP(IA),IA=1,NRSP)
      WRITE (6,1210)
      READ (5,1000) BLDATA,NULAKI,ERPRT,STRFN,SLCRD,ARPRT,INTVEL,
     1   SURVEL
      WRITE(6,1005) BLDATA,NULAKI,ERPRT,STRFN,SLCRD,ARPRT,INTVEL,
     1   SURVEL
C
C    END OF INPUT, CALCULATION OF CONSTANTS AND INITIALIZATION
C
      PITCH = 2.*3.1415927/FLOAT(NBL)
      DO 15 IA = 1,NLSP
   15 XSPL(IA) = XSPL(IA)+PITCH
      HA = CHORD/FLOAT(MXBO-MXBI)
      HB = PITCH/FLOAT(NBBI)
      DTLR = HB/1000.
      A = (PITCH+STGR)/HB
      B = STGR/HB
      NBBO = A-SIGN(DTLR,A)
      IF(A.LT.DTLR) NBBO = NBBO-1
```

```
      NBUO = B-SIGN(DTLR,B)
      IF(B.LT.-DTLR) NBUO = NBLO-1
      HU = FLOAT(NBUO+1)*HB-STGR
      HL = PITCH+STGR-FLOAT(NBBO)*HB
      BETAI =  BETAI/57.29577
      BETAO =  BETAO/57.29577
      ALUI=ALUI/57.29577
      ALLI=ALLI/57.29577
      ALUO=ALUO/57.29577
      ALLO=ALLO/57.29577
      NCH = MXBO-MXBI+1
      MXBIM1 = MXBI-1
      MXBOP1 = MXBO+1
      IBTE1 = 1000
      IBTE2 = 1000
      WR = .00001
      TOLER = .000001
      VTOL = .0001
      DO 20 IA=1,MX
   20 MPL(IA) = FLOAT(IA-MXBI)*HA
      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,MPL,MX,RM,SAL)
      CALL SPLINT (MR,BESP,NRSP,MPL,MX,BE,AAA)
      TBI = SIN(BETAI)/COS(BETAI)
      TBO = SIN(BETAO)/COS(BETAO)
      UNI = TBI/PITCH/RM(1)
      UNO = -TBO/PITCH/RM(MX)
      ALUI=SIN(ALUI)/COS(ALUI)/RM(1)
      ALLI=SIN(ALLI)/COS(ALLI)/RM(1)
      ALUO=SIN(ALUO)/COS(ALLO)/RM(MX)
      ALLO=SIN(ALLO)/COS(ALLO)/RM(MX)
C     CALCULATE LAMBDA AND VI
   25 RHOT = RHOIP
      RHOVI = WTFL/BE(1)/PITCH/COS(BETAI)/RM(1)
   30 VI = RHOVI/RHOT
      LAMBDA = RM(1)*(VI*SIN(BETAI)+OMEGA*RM(1))
      TTIP = 1.-(VI**2+2.*OMEGA*LAMBDA-(OMEGA*RM(1))**2)/CPTIP
      IF(TTIP.LE.0.) GO TO 35
      RHONEW = RHOIP*TTIP**EXPON
      IF(ABS(RHONEW-RHOT)/RHOIP.LT..000001) GO TO 40
      RHOT = RHONEW
      GO TO 30
   35 WTFL = WTFL/2.
      CALL ARERR(39HWTFL IS TOO LARGE AT UPSTREAM BOUNDARY$)
      WRITE(6,1400) WTFL
      GO TO 25
C     CALCULATE MAXIMUM VALUE FOR RHO*W
   40 VI = RHOVI/RHONEW
      LAMBDA = RM(1)*(VI*SIN(BETAI)+OMEGA*RM(1))
      TWL = 2.*OMEGA*LAMBDA
      AA = (TWL-(OMEGA*RM(1 ))**2)/CPTIP
      TPP = TIP*(1.-AA)
      B = TGROG*TPP
      TTIP = 1.-B/CPTIP-AA
      RHOT = RHOIP*TTIP**EXPON
```

```
      RHOWMI = RHOT*SQRT(B)
      AA = (TWL-(OMEGA*RM(MX))**2)/CPTIP
      TPP = TIP*(1.-AA)
      B = TGROG*TPP
      TTIP = 1.-B/CPTIP-AA
      RHOT = RHOIP*TTIP**EXPON
      RHOWMO = RHOT*SQRT(B)
C    CALCULATE VO AND W-CRITICAL
      RHOVI = WTFL/BE(MX)/PITCH/COS(BETAO)/RM(MX)
      RHOT = RHOIP
      TWLMR = TWL-(OMEGA*RM(MX))**2
      CALL CENSTY (RHOVI,RHOT,VO,TWLMR,CPTIP,EXPCN,RHCIP,GAM,AR,TIP,
     1   VTOL)
      WCRI = SQRT(TGROG*TIP*(1.-(TWL-(OMEGA*RM(MXBI))**2)/CPTIP))
      WCRO = SQRT(TGROG*TIP*(1.-(TWL-(CMEGA*RM(MXBO))**2)/CPTIP))
C    CALCULATE BETA CORRECTED TO BLADE LE CR TE
      TBI = (LAMBDA-OMEGA*RM(MXBI)**2)/WTFL*RHONEW*BE(MXBI)*PITCH
      BTAIN = ATAN(TBI)*57.29577
      TBO = (TBO/BE(MX)+OMEGA*(RM(MX)**2-RM(MXBO)**2)*RHOT/WTFL*
     1      PITCH)*BE(MXBO)
      BTAOUT = ATAN(TBO)*57.29577
      WRITE (6,1220) LAMBDA,VI,RHOWMI,BTAIN,WCRI,VO,RHCWMC,BTAOUT,WCRO
      WRITE (6,123C) HA,HB,HU,HL,PITCH,NBBC,NBUU
      IF (HB*RM(MXBO).LT.RO.AND.RO.LT.HA) WRITE (6,1240)
      DO 50 IA=1,1CC
      RHOU(IA) = RHOIP
      RHOL(IA) = RHOIP
      RHOUT(IA) = RHOIP
      RHOLT(IA) = RHOIP
      BEU(IA) = C.
      BEL(IA) = C.
   5C CONTINUE
      DO 60 I=1,25CC
   6C RHO(I) = RHOIP
      IF(BLDATA.GT.C)WRITE(6,125C) (MPL(IA),RM(IA),SAL(IA),BE(IA),
     1   AAA(IA),IA=1,MX)
C
C    CALCULATE MESH COORDINATES CN BOUNDARY
C
      AA = RM(MXBI)*ALUI
      MU(1) = RI*(1.-AA/SQRT(1.+AA**2))
      XSPU(1) =  RI/SQRT(1.+AA**2)/RM(MXBI)
      AA = RM(MXBI)*ALLI
      ML(1) = RI*(1.+AA/SQRT(1.+AA**2))
      XSPL(1) = -RI/SQRT(1.+AA**2)/RM(MXBI)+PITCH
      AA = RM(MXBO)*ALUO
      MU(NUSP) = CHORD-RO*(1.+AA/SQRT(1.+AA**2))
      XSPU(NUSP) =  RO/SQRT(1.+AA**2)/RM(MXBC)+STGR
      AA = RM(MXBO)*ALLO
      ML(NLSP) = CHORD-RO*(1.-AA/SQRT(1.+AA**2))
      XSPL(NLSP) = -RO/SQRT(1.+AA**2)/RM(MXBC)+STGR+PITCH
      CALL SPLN22(MU,XSPU,ALUI,ALUO,NUSP,SLUPE,EMU)
      CALL SPLN22(ML,XSPL,ALLI,ALLC,NLSP,SLLPE,EML)
      CALL BLDCRI (MU,XSPU,SLUPE,EMU,NUSP,RI,ALUI,RO,ALUC,CHCRC,STGR,
     1   PITCH, 1.,XU,NCH,ZINT,MXBI,DXDZU,RM(MXBI),RM(MXBC))
      CALL BLDCRI (ML,XSPL,SLLPE,EML,NLSP,RI,ALLI,RO,ALLO,CHORC,STGR,
     1   PITCH,-1.,XL,NCH,ZINT,MXBI,DXDZL,RM(MXBI),RM(MXBC))
      IF(BLDATA.LE.C) GO TO 65
      WRITE(6,126C)
```

```
      WRITE(6,1270)(MU(IA),XSPU(IA),SLOPE(IA),EMU(IA),IA=1,NUSP)
      WRITE(6,1265)
      WRITE(6,1270)(ML(IA),XSPL(IA),SLLPE(IA),EML(IA),IA=1,NLSP)
      WRITE(6,1280)(MPL(IA),XU(IA),DXDZU(IA),XL(IA),DXDZL(IA),
     1    IA=MXBI,MXBO)
   65 CONTINUE
C    CALCULATE NU,NL
      DO 70  IA=1,MXBIM1
      NL(IA) = NBBI-1
      NU(IA) = 0
      XU(IA) = 0.
   70 XL(IA) = PITCH
      DO 80  IA = MXBI,MXBO
      NU(IA) =  INT((XU(IA)+DTLR)/HB)
      IF(XU(IA).GT.-DTLR) NU(IA)=NU(IA)+1
      NL(IA) =  INT((XL(IA)-DTLR)/HB)
   80 IF(XL(IA).LT.DTLR) NL(IA) = NL(IA)-1
      DO 90  IA=MXBOP1,MX
      NU(IA) = NBUO
      NL(IA) = NBBO
      XU(IA) =  STGR
   90 XL(IA) = PITCH+STGR
      RETURN
  300 CALL  DEDERR (46H MX,NBBI,NUSP,NLSP,NRSP, OR NINT IS TOO LARGE$)
      RETURN
  999 FORMAT (1H1)
 1000 FORMAT (16I5)
 1005 FORMAT (1X,16I7)
 1010 FORMAT (8F10.5)
 1020 FORMAT (1X,8G16.7)
 1100 FORMAT (7X,3HGAM,14X,2HAR,13X,3HTIP,12X,5HRHOIP,12X,4HWTFL,11X,
     1    5HOMEGA,13X,1HW)
 1110 FORMAT (6X,5HCHORD,12X,4HSTGR,11X,5HBETAI,11X,5HBETAO)
 1120 FORMAT (8X,2HRI,13X,4HALUI,12X,4HALLI,13X,2HRO,13X,4HALUO,12X,
     1    4HALLO)
 1130 FORMAT (45H MXBI MXBO   MX NBBI  NUSP NLSP NRSP  NBL NINT)
 1140 FORMAT (10X,8HMU ARRAY)
 1150 FORMAT ( 10X,10HXSPU ARRAY)
 1160 FORMAT ( 10X,8HML ARRAY)
 1170 FORMAT ( 10X,10HXSPL ARRAY)
 1180 FORMAT ( 10X,8HMR ARRAY)
 1190 FORMAT ( 10X,10HRMSP ARRAY)
 1200 FORMAT ( 10X,10HBESP ARRAY)
 1210 FORMAT (57H  BLDATA NULAKI  ERPRT  STRFN  SLCRD  ARPRT INTVEL SURV
     1EL )
 1220 FORMAT (9H1LAMBDA =,G14.6,/12X,68HFREESTREAM       MAXIMUM VALUE
     1 BETA CORRECTED TO    BLADE CRITICAL/13X,63HVELOCITY         FOR RH
     2O*W          BLADE LE OR TE        VELOCITY/8H INLET  ,4G18.7/8H OUT
     3LET ,4G18.7)
 1230 FORMAT ( 1H1, 10X,28HCALCULATED PROGRAM CONSTANTS/7X,2HHA,14X,2HHB,
     1    14X,2HHU,14X,2HHL,13X,5HPITCH,12X,4HNBBO,6X,4HNBUO/1X,5G16.7,
     2    2I10)
 1240 FORMAT (78HL CAUTION - HB*RM(MXBO) LESS THAN RO LESS THAN HA MAY N
     1OT GIVE CORRECT RESULTS)
 1250 FORMAT (1H1,13X,44HSTREAM SHEET COORDINATES AND THICKNESS TABLE /
     1  7X,1HM,14X,1HR,13X,3HSAL,13X,1HB,12X,5HDB/DM / (5G15.5))
 1260 FORMAT ( 1H1,13X,27HBLADE DATA AT SPLINE POINTS /18X,16HUPPER    SU
     1RFACE)
 1265 FORMAT(18X,16HLOWER     SURFACE)
```

```
1270 FORMAT (7X ,1HM,10X,5HTHETA,1CX,10HDERIVATIVE,5X,10H2ND DERIV. /
    1  (4G15.5) )
1280 FORMAT (1H1,13X,22HBLADE COORDINATE TABLE/ 7X,1HM,14X,2HXU,11X,
    1  5HDXDZU,12X,2HXL,11X,5HDXDZL / (5G15.5) )
1400 FORMAT (23HLWEIGHT FLOW REDUCED TO,G14.6,7H KG/SEC)
     END
```

```
      SUBROUTINE COEF(NXN1)
      REAL K,K1,K2,K3,K4,K5,LMAX,LMIN,LAMBDA,MU,ML,MR,MXU,MXL,MPL
      INTEGER SRW,FIRST,CASE,BLDATA,ERPRT,STRFN,SLCRD,SLPLT,ARPRT,SURVEL
      COMMON SRW,ITER,GAM,AR,TIP,RHOIP,WTFL,OMEGA,LAMBDA,CP,EXPON,PITCH,
     1   CHORD,STGR,BETAI,BETAO,DTLR,RI,ALUI,ALLI,RO,ALUO,ALLO,
     2   MXBI,MXBO,MX,NBBI,NUSP,NLSP,NRSP,NINT,VTOL,
     3   BLDATA,NULAKI,ERPRT,STRFN,SLCRD,ARPRT,INTVEL,SURVEL,
     4   MU(50),XSPL(50),ML(50),XSPL(50),MR(50),RMSP(50),BESP(50),
     5   W,WR,TOLER,BDA,BDD,U(2500),A(2500,4),K(2500),RHO(2500),
     6   DXDZU(100),DXDZL(100),SLOPE(50),EMU(50),SLLPE(50),EML(50),
     7   RM(100),BE(100), SAL(100),XL(100),XL(100),RMU(100),RML(100),
     8   NU(100),NL(100),LINT(11),XINT(11),MPL(100),
     9   HA,HB,NXN,MXBIM1,MXBOP1,JU,JL,HU,HL,NBBC,NBUC,NCH,
     1   IBTE1,IBTE2,ITE2,HA1TE,HANTE,LNI,LNC,TWW,ITERA,
     2   RHOU(100),RHOL(100),RHOLT(100),RHCLT(100),BEU(100),BEL(100),
     3   AAA(100)
C    THE FOLLOWING VARIABLES ARE ALL IN COMMON BY EQUIVALENCE
      DIMENSION WM(2500),WX(2500),V(2500), BETA(2500),SL(1100)
      EQUIVALENCE (A(1,1),WM(1)),(A(1,2),WX(1)),(A(1,3),V(1)),
     1   (A(1,4), BETA(1)),(K(1401),SL(1))
      ITER = ITER+1
      IF(GAM.EQ.1.5.AND.AR.EQ.1000..AND.TIP.EQ.1.E6) GO TO 94
      IF(ITER.NE.1.AND.ITER.NE.2) GO TO 92
      NULAKI = NULAKI-1
      ERPRT = ERPRT-1
      STRFN = STRFN-1
      SLCRD = SLCRD-1
      ARPRT = ARPRT-1
      INTVEL = INTVEL-1
      SURVEL = SURVEL-1
   92 IF(ITER.NE.0) GO TO 94
      NULAKI = NULAKI+2
      ERPRT = ERPRT+2
      STRFN = STRFN+2
      SLCRD = SLCRD+2
      ARPRT = ARPRT+2
      INTVEL = INTVEL+2
      SURVEL = SURVEL+2
   94 I = 0
      INL = 0
      INU = 0
      IF(BLDATA.GT.0) WRITE(6,1440)
```

```
C
C   CALCULATE COEFFICIENTS
C
      DO 200 IA=1,MX
      X1 = FLOAT(NL(IA))*HB
      IF(IA.GT.MXBO) X1 = STGR
      NBB = NL(IA)-NL(IA)+1
      K5 = 0.
      IBDU = NU(IA-1)-NU(IA)
      IBDL = NL(IA)-NL(IA+1)
      DO 200 IB=1,NBB
      I = I+1
      IF(I.GT.2500) GO TO 200
      IF(IA.NE.1) GO TO 100
      A(I,1) = 0.
      A(I,2) = 0.
      A(I,3) = 0.
      A(I,4) = 1.
      K(I) = HA*UNI
      GO TO 190
  100 IF(IA.NE.MX) GO TO 110
      A(I,1) = 0.
      A(I,2) = 0.
      A(I,4) = 0.
      A(I,3) = 1.
      K(I) = HA*UNO
      GO TO 190
  110 K1 = 0.
      K2 = 0.
      K3 = 0.
      K4 = 0.
      H3 = HA
      H4 = HA
      IF((NU(IA)+IB).LE.NU(IA-1)) CALL INTPL1(XU,IA,X1,H3,0,K3,MU,XSPU,
     1   ALUI,ALUO,NUSP,SLUPE,EMU,RI,RO,CHORD,STGR,PITCH, 1.,HA,HB,
     2   MXBI,MXBO,RM(MXBI),RM(MXBO))
      IF((NU(IA)+IB).LE.NU(IA+1)) CALL INTPL1(XU,IA,X1,H4,1,K4,MU,XSPU,
     1   ALUI,ALUO,NUSP,SLUPE,EMU,RI,RO,CHORD,STGR,PITCH, 1.,HA,HB,
     2   MXBI,MXBO,RM(MXBI),RM(MXBO))
      IF((NU(IA)+IB).GT.(NL(IA-1)+1))  CALL INTPL1(XL,IA,X1,H3,0,K3,ML,
     1   XSPL,ALLI,ALLO,NLSP,SLLPE,EML,RI,RO,CHORD,STGR,PITCH,-1.,HA,HB,
     2   MXBI,MXBO,RM(MXBI),RM(MXBO))
      IF((NU(IA)+IB).GT.(NL(IA+1)+1)) CALL INTPL1(XL,IA,X1,H4,1,K4,ML,
     1   XSPL,ALLI,ALLO,NLSP,SLLPE,EML,RI,RO,CHORD,STGR,PITCH,-1.,HA,HB,
     2   MXBI,MXBO,RM(MXBI),RM(MXBO))
      IF(IA-MXBO+1) 140,120,130
C   SPECIAL CALCULATION OF COEFFICIENTS AT TRAILING EDGE OF LOWER SURFACE OF
C       UPPER BLADE
  120 IF(X1.LT.PITCH+STGR-RO/RM(MXBO)) GO TO 140
      IF(X1.GT.PITCH+STGR) GO TO 140
      IF(IB+NU(IA)-1.GT.NBBO) GO TO 140
      HAMRO = HA-RO
      XL(MXBO) = PITCH+STGR-RO/RM(MXBO)
      CALL INTPL1(XL,IA,X1,H4,1,K4,ML,XSPL,ALLI,ALLO,NLSP,SLLPE,EML,
     1 RI,RO,CHORD,STGR,PITCH,-1.,HAMRO,HB,MXBI,MXBO,RM(MXBI),RM(MXBO))
      HANTE = H4
      IBTE2 = NL(IA)+IB-1
      XL(MXBO) = PITCH+STGR
      IBDL = IBDL+1
      GO TO 140
```

```
130 IF( IA.GT.MXBO) GO TO 140
    IF( IBTE2.EQ.1000) GO TO 140
    IF(X1.LT.PITCH+STGR-RO/RM(MXBO)) GO TC 140
    B = XL(MXBO-1)
    XL(MXBO-1) = PITCH+STGR-RO/RM(MXBO)
    CALL INTPL1( XL,IA,X1,H3,0,K3,ML,XSPL,ALLI,ALLC,NLSP,SLLPE,EML,
   1   RI,RO,CHORD,STGR,PITCH,-1.,RC,HB,MXBI,MXBO,RM(MXBI),RM(MXBO))
    HAITE = H3
    IBTE1 = NL( IA)+IB-1
    ITE2 = I
    XL(MXBO-1) = B
140 IF( IB.EQ.1) K1 = 1.
    IF( IB.EQ.NBB) K2 = 1.
    H1 = HE*RM( IA)
    H2 = H1
    I1 = I-1
    I2 = I+1
    I3 = I-NL( IA-1)+NL( IA)-1
    I4 = I+NL( IA)-NL( IA+1)+1
    IF( IA.GE.MXBI) GO TO 142
    IF( IB.EQ.1) I1 = I1+NBB
    IF( IB.EQ.NBB) I2 = I2-NBB
    GO TO 148
142 IF( IA.GT.MXBO) GO TO 144
    IF( IB.EQ.1) H1 = (X1-XU(IA))*RM(IA)
    IF( IB.EQ.NBB) H2 = (XL(IA)-X1)*RM(IA)
    GO TO 148
144 IF( IB.NE.1) GO TO 146
    I1 = I1+NBB
    H1 = HL*RM( IA)
    H2 = HL*RM( IA)
    GO TO 148
146 IF( IB.EQ.2) H1 = HU*RM(IA)
    IF( IB.NE.NBB) GO TO 148
    I2 = I2-NBB
    H2 = HL*RM( IA)
148 RHO1 = RHO( I1)
    RHO2 = RHO( I2)
    RHO3 = RHO( I3)
    RHO4 = RHO( I4)
    BE3 = BE( IA-1)
    BE4 = BE( IA+1)
    IF(( IA.GE.MXBI.AND.IA.LE.MXBO).AND.IB.EQ.1) RHO1 = RHOU(IA)
    IF(( IA.GE.MXBI.AND.IA.LE.MXBO).AND.IB.EQ.NBB) RHO2 = RHOL(IA)
    IF(K5.LT..5) GO TO 160
    IF(K3.LT..5) GO TO 150
    INL = INL+1
    BE3 = BEL( INL)
    RHO3 = RHOLT( INL)
    IF( BE3.NE.0.) GO TO 150
    B = MPL( IA)-H3
    CALL SPLINT (MR,BESP,NRSP,B,1,BEL(INL),AAA)
    BE3 = BEL( INL)
    IF( BLDATA.GT.0) WRITE(6,1460) B,INL
150 IF(K4.LT..5) GO TO 180
    INL = INL+1
    ITP = INL-IBDL+1+2*(NBB-IB)
    BE4 = BEL( ITP)
    RHO4 = RHOLT( ITP)
    IF( BE4.NE.0.) GO TO 180
```

```
      B = MPL(IA)+H4
      CALL  SPLINT (MR,BESP,NRSP,B,1,BEL(ITP),AAA)
      BE4 = BEL(ITP)
      IF(BLCATA.GT.0) WRITE(6,146C) B,INL,ITP
      GO TO 180
  16C CONTINUE
      IF(K3.LT..5) GO TO 17C
      INU =  INU+1
      ITP =  INU+IBDU+1-2*IB
      BE3 = BEU(ITP)
      RHO3 = RHOUT(ITP)
      IF(BE3.NE.C.) GO TO 170
      B = MPL(IA)-H3
      CALL  SPLINT (MR,BESP,NRSP,B,1,BEU(ITP),AAA)
      BE3 = BEU(ITP)
      IF(BLCATA.GT.C) WRITE(6,1450) B,INU,ITP
  170 IF(K4.LT..5) GO TO 180
      INU =  INU+1
      BE4 = BEU(INU)
      RHO4 = RHOUT(INU)
      IF(BE4.NE.C.) GO TO 18C
      B = MPL(IA)+H4
      CALL  SPLINT (MR,BESP,NRSP,B,1,BEL(INU),AAA)
      BE4 = BEU(INU)
      IF(BLCATA.GT.C) WRITE(6,145C) B,INU
  18C CONTINUE
      A12 = 2./H1/H2
      A34 = 2./H3/H4
      AII = A12+A34
      B12 = (RHO2-RHO1)/RHO(I)/(H1+H2)
      B34 = (BE4*RHO4-BE3*RHO3)/BE(IA)/RHO(I)/(H3+H4)-SAL(IA)/RM(IA)
      A(I,1) = (2./H1+B12)/(H1+H2)/AII
      A(I,2) = A12/AII-A(I,1)
      A(I,3) = (2./H3+B34)/(H3+H4)/AII
      A(I,4) = A34/AII-A(I,3)
      K(I) = -TWW*BE(IA)*RHO(I)*SAL(IA)/AII
      IF(K3.LT..5.AND.K4.LT..5) K5 = 1.
      IF(IA.LT.MXBI.OR.IA.GT.MXBO) GO TO 185
      K(I) = K(I)+K5*(K2*A(I,2)+K3*A(I,3)+K4*A(I,4))
      IF(K1.GT..5) A(I,1) = C.
      IF(K2.GT..5) A(I,2) = C.
  185 IF(K3.GT..5) A(I,3) = C.
      IF(K4.GT..5) A(I,4) = C.
  19C X1 = FLOAT(NU(IA)+IB)*HB
      IF (IA.GE.MXBI.AND.IA.LE.MXBC) GO TC 200
      IF(IB.EQ.1) K(I) = -A(I,1)+K(I)
      IF(IB.EQ.NBB) K(I) = A(I,2)+K(I)
  20C CONTINUE
      IF(ITER.EQ.1)WRITE(6,147C) INU,INL
      IF(INU.GT.100.OR.INL.GT.1CO) GC TO 310
      IF((ITER .GT. 1).AND.(INU.NE.JL.CR.INL.NE.JL)) GC TC 320
      NXN = I
      WRITE (6,142C) NXN
      IF(NXN.GT.25CO) WRITE(6,1430)
      NXN1 = NXN
      IF(BLCATA.LE.0) GO TO 21C
      WRITE (6,141C)  (NU(IA),NL(IA) ,IA=1,MX)
  21C BLCATA = C
      RETURN
```

```
  310 CALL DEDERR (35H INU AND INL MUST BE LESS THAN 100$)
  320 CALL DEDERR (44H INU NOT EQUAL TO JU OR INL NOT EQUAL TO JL$)
      RETURN
 1410 FORMAT (18HLLIST OF NU AND NL / (2I10))
 1420 FORMAT (1HL,5X,32HNUMBER OF INTERIOR MESH POINTS =,I5)
 1430 FORMAT(76HKTHE NUMBER OF UNKNOWN MESH POINTS EXCEEDS 2500, A COARS
     1ER GRID MUST BE USED)
 1440 FORMAT (31HL         M           INU  INL  ITP)
 1450 FORMAT (1X,G13.4,I5,I10)
 1460 FORMAT (1X,G13.4,I10,I5)
 1470 FORMAT(6HLINU =,I3,5X,5HINL =,I3)
      END
```

```
      SUBROUTINE SOR
      REAL K,K1,K2,K3,K4,K5,LMAX,LMIN,LAMBDA,MU,ML,MR,MXU,MXL,MPL
      INTEGER SRW,FIRST,CASE,BLDATA,ERPRT,STRFN,SLCRD,SLPLT,ARPRT,SURVEL
      COMMON SRW,ITER,GAM,AR,TIP,RHOIP,WTFL,OMEGA,LAMBDA,CP,EXPON,PITCH,
     1   CHORD,STGR,BETAI,BETAO,DTLR,RI,ALUI,ALLI,RO,ALUO,ALLO,
     2   MXBI,MXBO,MX,NBBI,NUSP,NLSP,NRSP,NINT,VTOL,
     3   BLDATA,NULAKI,ERPRT,STRFN,SLCRD,ARPRT,INTVEL,SURVEL,
     4   MU(50),XSPU(50),ML(50),XSPL(50),MR(50),RMSP(50),BESP(50),
     5   W,WR,TOLER,BDA,BDD,U(2500),A(2500,4),K(2500),RHO(2500),
     6   DXDZU(100),DXDZL(100),SLUPE(50),EMU(50),SLLPE(50),EML(50),
     7   RM(100),BE(100), SAL(100),XU(100),XL(100),RMU(100),RML(100),
     8   NU(100),NL(100),UINT(11),XINT(11),MPL(100),
     9   HA,HB,NXN,MXBIM1,MXBOP1,JU,JL,HU,HL,NBBC,NBUC,NCH,
     1   IBTE1,IBTE2,ITE2,HA1TE,HANTE,UNI,LNO,TWW,ITERA,
     2   RHOU(100),RHOL(100),RHOUT(100),RHOLT(100),BEU(100),BEL(100),
     3   AAA(100)
C   THE FOLLOWING VARIABLES ARE ALL IN COMMON BY EQUIVALENCE
      DIMENSION WM(2500),WX(2500),V(2500), BETA(2500),SL(1100)
      EQUIVALENCE (A(1,1),WM(1)),(A(1,2),WX(1)),(A(1,3),V(1)),
     1   (A(1,4), BETA(1)),(K(1401),SL(1))
      DIMENSION IU(100),IL(100)
      EQUIVALENCE (K(1101),IU(1)),(K(1201),IL(1))
C
C   ESTIMATE WBEST
C
      NUTEMP = NULAKI
      IF(W.GE.1.) GO TO 225
  190 DO 200 I=1,NXN
  200 U(I) = 1.
      WMAX = 2.
  210 I = 0
      WMAX1 = WMAX
      LMAX = 0.
      LMIN = 1.
      DO 220 IA=1,MX
      NBB = NL(IA)-NU(IA)+1
      DO 220 IB =1,NBB
      I = I+1
```

```
      I1 = I-1
      I2 = I+1
      IF(((IA.LT.MXBI).OR.(IA.GT.MXBO)).AND.(IB.EQ.1)) I1 = I1+NBB
      IF(((IA.LT.MXBI).OR.(IA.GT.MXBO)).AND.(IB.EQ.NBB)) I2 = I2-NBB
      I3 = I-NL(IA-1)+NU(IA)-1
      I4 = I+NL(IA)-NU(IA+1)+1
      UNEW = A(I,1)*U(I1)+A(I,2)*U(I2)+A(I,3)*U(I3)+A(I,4)*U(I4)
      RATIO = UNEW/U(I)
      LMAX = AMAX1(RATIO,LMAX)
      LMIN = AMIN1(RATIO,LMIN)
  215 U(I) = UNEW
  220 CONTINUE
      WMAX = 2./(1.+SQRT(ABS(1.-LMAX)))
      WMIN = 2./(1.+SQRT(1.-LMIN))
      WRITE (6,1500) WMAX,WMIN,LMAX,LMIN
      IF(((WMAX1-WMAX).GT.WR).OR.(WMAX.GT.(2.-100.*WR))) GO TO 210
      W = WMAX
C
C   CALCULATE INITIAL SOLUTION ESTIMATE
C
  225 I = 0
      IF(ITER.NE.1) GO TO 260
      DO 230 IA=1,MXBIM1
      NBB = NL(IA)-NU(IA)+1
      DO 230 IB=1,NBB
      I = I+1
      U(I) =    FLOAT(IB-1)/FLOAT(NBB)
  230 CONTINUE
      DO 240 IA=MXBI,MXBO
      NBB = NL(IA)-NU(IA)+1
      DO 240 IB=1,NBB
      I = I+1
      J = NU(IA)+IB-1
      U(I) = (HB*FLOAT(J)-XU(IA))/(XL(IA)-XU(IA))
  240 CONTINUE
      DO 250 IA=MXBOP1,MX
      NBB = NL(IA)-NU(IA)+1
      DO 250 IB=1,NBB
      I = I+1
      U(I) =    FLOAT(IB-1)/FLOAT(NBB)
  250 CONTINUE
C
C   SOLVE MATRIX EQUATION BY SOR
C
  260 I = 0
      IF(NUTEMP.GT.0) WRITE (6,1450)
      ERROR = 0.
      DO 270 IA=1,MX
      NBB = NL(IA)-NU(IA)+1
      DO 270 IB=1,NBB
      I = I+1
      I1 = I-1
      I2 = I+1
      IF(((IA.LT.MXBI).OR.(IA.GT.MXBO)).AND.(IB.EQ.1)) I1 = I1+NBB
      IF(((IA.LT.MXBI).OR.(IA.GT.MXBO)).AND.(IB.EQ.NBB)) I2 = I2-NBB
      I3 = I-NL(IA-1)+NU(IA)-1
      I4 = I+NL(IA)-NU(IA+1)+1
      CHANGE = W*(R(I)-U(I)+A(I,1)*U(I1)+A(I,2)*U(I2)+A(I,3)*U(I3)+
```

```
     1   A(I,4)*U(I4))
      ERROR = AMAX1(ERROR,ABS(CHANGE))
      U(I) = U(I)+CHANGE
      IF(NUTEMP.LE.0) GO TO 270
      IF(IA.EQ.1) I3=0
      IF(IA.EQ.MX) I4=0
      WRITE (6,1460) IA,I,(A(I,J),J=1,4),I1,I2,I3,I4,K(I)
  270 CONTINUE
      NUTEMP = 0
      IF(ERPRT.GT.0) WRITE (6,1510) ERROR
      IF(ERROR.GT.TOLER) GO TO 260
      IF(STRFN.GT.0) WRITE(6,1520)
      LAST = 0
      DO 280 IA=1,MX
      IF(STRFN.GT.0) WRITE (6,1525) IA
      FIRST = LAST+1
      LAST = FIRST+NL(IA)-NU(IA)
      IU(IA)=FIRST
      IL(IA)=LAST
  280 IF(STRFN.GT.0) WRITE (6,1530) (U(I),I=FIRST,LAST)
      RETURN
 1450 FORMAT (88H1   IA     I      A(I,1)     A(I,2)     A(I,3)     A(I,4)
     1I1      I2      I3      I4       K(I)   )
 1460 FORMAT (2X,I3,I6,4F10.5,4I7,F10.5)
 1500 FORMAT (7H WMAX =,F9.6,5X,6HWMIN =,F9.6,5X,6HLMAX =,F9.6,5X,
     1   6HLMIN =,F9.6)
 1510 FORMAT ( 8H ERROR =,F11.8)
 1520 FORMAT (1H1,10X,22HSTREAM FUNCTION VALUES)
 1525 FORMAT (5H IA =,I3)
 1530 FORMAT (2X,10F13.8)
      END
```

```
      SUBROUTINE SLAXVL
      REAL K,K1,K2,K3,K4,K5,LMAX,LMIN,LAMBDA,MU,ML,MR,MXU,MXL,MPL
      INTEGER SRW,FIRST,CASE,BLDATA,ERPRT,STRFN,SLCRD,SLPLT,ARPRT,SURVEL
      COMMON SRW,ITER,GAM,AR,TIP,RHOIP,WTFL,OMEGA,LAMBDA,CP,EXPON,PITCH,
     1   CHORD,STGR,BETAI,BETAO,DTLR,RI,ALUI,ALLI,RO,ALUO,ALLO,
     2   MXBI,MXBO,MX,NBBI,NUSP,NLSP,NRSP,NINT,VTOL,
     3   BLDATA,NULAKI,ERPRT,STRFN,SLCRD,ARPRT,INTVEL,SURVEL,
     4   MU(50),XSPU(50),ML(50),XSPL(50),MR(50),RMSP(50),BESP(50),
     5   W,WR,TOLER,BDA,BDD,U(2500),A(2500,4),K(2500),RHO(2500),
     6   DXDZU(100),DXDZL(100),SLUPE(50),EMU(50),SLLPE(50),EML(50),
     7   RM(100),BE(100), SAL(100),XU(100),XL(100),RMU(100),RML(100),
     8   NU(100),NL(100),UINT(11),XINT(11),MPL(100),
     9   HA,HB,NXN,MXBIM1,MXBOP1,JU,JL,HU,HL,NBBC,NBUC,NCH,
     1   IBTE1,IBTE2,ITE2,HAITE,HANTE,LNI,LNC,TWW,ITERA,
     2   RHOU(100),RHOL(100),RHOUT(100),RHOLT(100),BEU(100),BEL(100),
     3   AAA(100)
C    THE FOLLOWING VARIABLES ARE ALL IN COMMON BY EQUIVALENCE
      DIMENSION WM(2500),WX(2500),V(2500), BETA(2500),SL(1100)
      DIMENSION WMU(100),WML(100),WXU(100),WXL(100),MXU(100),MXL(100),
```

```
     1    BETAU(100), BETAL(100),WU(100),WL(100),
     2    USP(100),IU(100),IL(100)
      EQUIVALENCE (A(1,1),WM(1)),(A(1,2),WX(1)),(A(1,3),V(1)),
     1    (A(1,4), BETA(1)),(K(1401),SL(1))
      EQUIVALENCE (K(1),WMU(1)),(K(101),WML(1)),(K(201),WXU(1)),
     1    (K(301),WXL(1)),(K(401),MXU(1)),(K(501),MXL(1)),
     2    (K(601), BETAU(1)),(K(701), BETAL(1)),(K(801),WU(1)),
     3    (K(901),WL(1)),(K(1001),USP(1)),(K(1101),IU(1)),
     4    (K(1201),IL(1))
      DIMENSION XBB(52),WMSP(52),KKK(24),P(11)
      DATA (KKK(J),J=4,24,2)/11*1H*/
C
C     CALCULATE STREAMLINE LOCATION -- UPSTREAM
C
      I1=1
      I = 0
      DELINT = 1./FLOAT(NINT)
      NBB = NL(1)+1
      XBB(1) = 0.
      IF(SLCRD.GT.0) WRITE(6,1550)
      DO 320 IB =1,NBB
  320 XBB(IB+1) = XBB(IB)+HB
      DO 350 IA = 1,MXBIM1
      UINT(1) = AINT(U(I+1)/DELINT)*DELINT
      IF(U(I+1).GT.0.) UINT(1) = UINT(1)+DELINT
      DO 330 JB=2,NINT
  330 UINT(JB) = UINT(JB-1)+DELINT
      DO 340 IB=1,NBB
      I = I+1
  340 USP(IB) = U(I)
      NSP = NBB+1
      USP(NSP) = USP(1)+1.
      IF(SLCRD.GT.0) CALL SPLINT (USP,XBB,NSP,UINT,NINT,XINT,AAA)
      CALL SPLINE(XBB,USP,NSP,WM(I1),AAA)
      I1=I1+NL(IA)+1
      IF(SLCRD.GT.0) WRITE(6,1540) MPL(IA),(UINT(J),XINT(J),J=1,NINT)
      DO 345 JB=1,NINT
      J = MX*(JB-1)+IA
      SL(J) = XINT(JB)
  345 CONTINUE
      J1 = MX*NINT+IA
      SL(J1) = SL(J)
  350 CONTINUE
C
C     CALCULATE STREAMLINE LOCATION -- BLADE
C
      JU=MXBI-1
      NINT = NINT+1
      UINT(1) = 0.
      DO 360 JB=2,NINT
  360 UINT(JB) = UINT(JB-1)+DELINT
      USP(1) = 0.
      DO 380 IA=MXBI,MXBO
      XBB(1) = XU(IA)
      NBB = NL(IA)-NU(IA)+1
      XBB(2) = FLOAT(NU(IA))*HB
      DO 370 IB = 1,NBB
      I = I+1
      USP(IB+1) = U(I)
```

```
  370 XBB(IB+1) = FLOAT(IB-1)*HB+XBB(2)
      NSP = NBB+2
      USP(NSP) = 1.
      XBB(NSP) = XL(IA)
      IF(SLCRD.GT.C) CALL SPLINT (USP,XBB,NSP,UINT,NINT,XINT,AAA)
      CALL SPLINE(XBB,USP,NSP,WMSP,AAA)
      DO 375 IB=1,NBB
      WM(I1)=WMSP(IB+1)
  375 I1=I1+1
      JU=JU+1
      WMU(JU) = WMSP(1)
      WML(JU) = WMSP(NSP)
      IF(SLCRD.GT.C) WRITE(6,154C) MPL(IA),(UINT(J),XINT(J),J=1,NINT)
      DO 38C JB=1,NINT
      J = MX*(JB-1)+IA
      SL(J) = XINT(JB)
  38C CONTINUE
C
C    CALCULATE STREAMLINE LOCATION -- DOWNSTREAM
C
      NINT = NINT-1
      NBB = NL(MXBOP1)-NU(MXBOP1)+1
      NSP = NBB+1
      XBB(1) = STGR
      XBB(2) = STGR+HU
      DO 39C IB=3,NBB
  39C XBB(IB) = XBB(IB-1)+HB
      XBB(NSP) = STGR+PITCH
      DO 42C IA=MXBOP1,MX
      UINT(1) = AINT(U(I+1)/DELINT)*DELINT
      IF(U(I+1).GT.0.) UINT(1) = UINT(1)+DELINT
      DO 40C JB=2,NINT
  40C UINT(JB) = UINT(JB-1)+DELINT
      DO 41C IB=1,NBB
      I = I+1
  410 USP(IB) = U(I)
      USP(NSP) = USP(1)+1.
      IF(SLCRD.GT.0) CALL SPLINT (USP,XBB,NSP,UINT,NINT,XINT,AAA)
      CALL SPLINE(XBB,USP,NSP,WM(I1),AAA)
      I1=I1+NL(IA)-NU(IA)+1
      IF(SLCRD.GT.C) WRITE(6,154C) MPL(IA),(UINT(J),XINT(J),J=1,NINT)
      DO 415 JB=1,NINT
      J = MX*(JB-1)+IA
      SL(J) = XINT(JB)
  415 CONTINUE
      J1 = MX*NINT+IA
      SL(J1) = SL(J)
  42C CONTINUE
C
C    PLOT STREAMLINES
C
      IF(SLCRC.LE.C) GO TO 48C
C    CALCULATE PLOTTING PARAMETERS
      ZMIN = MPL(1)
      XMAX = XL(1)
      XMIN = XU(1)
      DO 43C IA = 2,MX
      XMAX = AMAX1(XMAX,XL(IA))
```

```
 430 XMIN = AMIN1(XMIN,XU(IA))
     DX = XMAX-XMIN
     XFACT = 2.
 440 IF(DX.GT.10.) GO TO 450
     DX = DX*10.
     XFACT = XFACT+1.
     GO TO 440
 450 IF(DX.LE.100.) GO TO 460
     DX = DX/10.
     XFACT = XFACT-1.
     GO TO 450
 460 DX = AINT(DX+1.)
     DZ = AINT(5.*DX/3.*RM(MXBI))
     ZFACT = XFACT
     ZMIN = AINT(ZMIN*10.**ZFACT)
     XMIN = AINT(XMIN*10.**XFACT)
     KKK(1) = 49
     KKK(2) = NINT+1
     P(1) = 1.
     P(5) = 0.
     P(6) = 6.-ZFACT
     P(7) = ZMIN
     P(8) = DZ
     P(9) = 6.-XFACT
     P(10) = XMIN
     P(11) = DX
     KKK(3) = MX
     WRITE(6,1530)
     CALL PLOTMY (MPL,SL,KKK,P)
     WRITE (6,1560)
     DO 470 IA=1,MX
 470 MPL(IA) = FLOAT(IA-MXBI)*HA
 480 LAST = 0
     DO 482 IA=1,MX
     FIRST = LAST+1
     LAST = FIRST+NL(IA)-NU(IA)
     DO 482 I=FIRST,LAST
 482 WM(I) = WM(I)*WTFL/BE(IA)/RM(IA)
     DO 484 IA=MXBI,MXBU
     WMU(IA) = WMU(IA)*WTFL/BE(IA)/RM(IA)
 484 WML(IA) = WML(IA)*WTFL/BE(IA)/RM(IA)
     IF(ARPRT.LE.0) RETURN
     WRITE (6,1600)
     LAST=0
     DO 490 IA=1,MX
     FIRST=LAST+1
     LAST = FIRST+NL(IA)-NU(IA)
 490 WRITE (6,1020) (WM(I),I=FIRST,LAST)
     WRITE (6,1610)
     WRITE (6,1020) (WMU(IA), IA=MXBI,MXBU)
     WRITE (6,1620)
     WRITE (6,1020) (WML(IA), IA=MXBI,MXBU)
     RETURN
1020 FORMAT (1X,8G16.7)
1530 FORMAT (2HPT,50X,16HSTREAMLINE PLOTS   )
1540 FORMAT (1X,7G18.7/(19X,6G18.7))
1550 FORMAT (1H1,26X,22HSTREAMLINE COORDINATES///7X,8HM COORD.,
    1   3(9X,10HSTREAM FN.,9X,8H   THETA )//)
```

```
 1560 FORMAT (2HPL,40X,7CHSTREAMLINES ARE PLOTTED WITH THETA ACROSS THE
     1PAGE AND M DOWN THE PAGE)
 1600 FORMAT (1H0,22HWM ARRAY (RHO*W-SUB-M))
 1610 FORMAT (1H0,40HWMU ARRAY (RHO*W-SUB-M ON UPPER SURFACE))
 1620 FORMAT (1H0,40HWML ARRAY (RHO*W-SUB-M ON LOWER SURFACE))
      END
```

```
      SUBROUTINE TASVEL
      REAL K,K1,K2,K3,K4,K5,LMAX,LMIN,LAMBDA,MU,ML,MR,MXU,MXL,MPL
      INTEGER SRW,FIRST,CASE,BLDATA,ERPRT,STRFN,SLCRD,SLPLT,ARPRT,SURVEL
      COMMON SRW,ITER,GAM,AR,TIP,RHOIP,WTFL,OMEGA,LAMBDA,CP,EXPON,PITCH,
     1   CHORD,STGR,BETAI,BETAO,DTLR,RI,ALUI,ALLI,RO,ALUO,ALLO,
     2   MXBI,MXBO,MX,NBBI,NUSP,NLSP,NRSP,NINT,VTOL,
     3   BLDATA,NULAKI,ERPRT,STRFN,SLCRD,ARPRT,INTVEL,SURVEL,
     4   MU(50),XSPU(50),ML(50),XSPL(50),MR(50),RMSP(50),BESP(50),
     5   W,WR,TOLER,BDA,BDD,U(2500),A(2500,4),K(2500),RHO(2500),
     6   DXDZU(100),DXDZL(100),SLOPE(50),EMU(50),SLLPE(50),EML(50),
     7   RM(100),BE(100), SAL(100),XU(100),XL(100),RMU(100),RML(100),
     8   NU(100),NL(100),UINT(11),XINT(11),MPL(100),
     9   HA,HB,NXN,MXBIM1,MXBOP1,JU,JL,HU,HL,NBBC,NBUC,NCH,
     1   IBTE1,IBTE2,ITE2,HA1TE,HANTE,UNI,UNC,TWW,ITERA,
     2   RHOU(100),RHOL(100),RHOUT(100),RHOLT(100),BEU(100),BEL(100),
     3   AAA(100)
C   THE FOLLOWING VARIABLES ARE ALL IN COMMON BY EQUIVALENCE
      DIMENSION WM(2500),WX(2500),V(2500), BETA(2500),SL(1100)
      DIMENSION WMU(100),WML(100),WXU(100),WXL(100),MXU(100),MXL(100),
     1    BETAU(100), BETAL(100),WU(100),WL(100),
     2   USP(100),IU(100),IL(100)
      EQUIVALENCE (A(1,1),WM(1)),(A(1,2),WX(1)),(A(1,3),V(1)),
     1   (A(1,4), BETA(1)),(K(1401),SL(1))
      EQUIVALENCE (K(1),WMU(1)),(K(101),WML(1)),(K(201),WXU(1)),
     1   (K(301),WXL(1)),(K(401),MXU(1)),(K(501),MXL(1)),
     2   (K(601), BETAU(1)),(K(701), BETAL(1)),(K(801),WU(1)),
     3   (K(901),WL(1)),(K(1001),USP(1)),(K(1101),IU(1)),
     4   (K(1201),IL(1))
      DIMENSION DTDMU(100),DTDML(100),BBB(100)
      DIMENSION XDOWN(400),YACROS(400),KKK(14)
      EQUIVALENCE (K(1301),DTDMU(1)),(K(1401),DTDML(1)),
     1   (K(1501),XDOWN(1)),(K(1901),YACROS(1))
C
C  CALCULATE RHO*W-SUB-THETA
C
C  START AT IA = 1 (CASES 1,2,3)
C
      CASE =1
      JU = 0
      JL = 0
      KK1 = 0
      KN = 0
      IB = 0
      IA1 = 1
```

```
      HA1 = HA
      IAN = MXBI
      I = NBBI+1
C  END ON UPPER SURFACE (CASE 1)
  510 IF(IB.GT.NL(1)) GO TO 600
      DO 530 IA=IAN,MXBO
  530 IF(NU(IA).GT.IB) GO TO 540
      CASE = 2
      GO TO 550
  540 IAN = IA
      IA = IAN-1
      X1 = FLOAT(IB)*HB
      CALL INTPL1(XU,IA,X1,HAN,1,K4,MU,XSPL,ALUI,ALUO,NUSP,SLOPE,EMU,
     1   RI,RO,CHORD,STGR,PITCH,1.,HA,HB,MXBI,MXBO,RM(MXBI),RM(MXBO))
      GO TO 770
C  END AT IA = MX (CASE 2)
  550 IF(IB.NE.IBTE1) GO TO 554
      IAN = MXBO
      HAN = HANTE
      KN = 1
      GO TO 770
  554 DO 555 IA=MXBI,MXBO
  555 IF(IB.GT.NL(IA)) GO TO 560
      IF(IB.GT.NL(1)) GO TO 600
      IAN = MX
      HAN = HA
      GO TO 770
C  END ON LOWER SURFACE (CASE 3)
  560 CASE = 3
      IF(IB.GT.NL(1)) GO TO 600
      DO 570 IA=MXBI,MXBO
  570 IF(NL(IA).LT.IB) GO TO 580
      IA=MXBO
  580 IAN = IA
      IA = IAN-1
      X1 = FLOAT(IB)*HB
      CALL INTPL1(XL,IA,X1,HAN,1,K4,ML,XSPL,ALLI,ALLO,NLSP,SLOPE,EML,
     1   RI,RO,CHORD,STGR,PITCH,-1.,HA,HB,MXBI,MXBO,RM(MXBI),RM(MXBO))
      KN = 1
      GO TO 770
C  START ON LOWER SURFACE (CASE 4)
  600 CASE = 4
      KK1=1
  620 DO 630 IA=MXBI,MXBO
  630 IF(NL(IA+1).GT.NL(IA)) GO TO 640
      CASE=5
      GO TO 680
  640 IB = NL(IA)+1
      IA1 = IA
  650 DO 660 IA=IA1,MXBO
  660 IF(NL(IA+1).GE.IB) GO TO 670
      CASE = 5
      GO TO 680
  670 IA1 = IA
      IA = IA1+1
      X1 = FLOAT(IB)*HB
      I = IL(IA)+IB-NL(IA)
```

```
      CALL  INTPL1(XL,IA,X1,HA1,0,K3,ML,XSPL,ALLI,ALLC,NLSP,SLLPE,EML,
     1   RI,RO,CHORD,STGR,PITCH,-1.,HA,HB,MXBI,MXBO,RM(MXBI),RM(MXBO))
      GO TO 740
C  START ON UPPER SURFACE (CASE 5)
  680 DO 690 IA=MXBI,MXBO
  690 IF(NU(IA+1).LT.NU(IA)) GO TO 700
      CASE = 6
      GO TO 765
  700 IB = NU(IA)-1
      IA1 = IA
      KK1=0
  710 DO 720 IA=IA1,MXBO
  720 IF(NU(IA+1).LE.IB) GO TO 730
      CASE = 6
      GO TO 765
  730 IA1 = IA
      IA = IA +1
      X1 = FLOAT(IB)*HB
      IF(IB.EQ.NU(MX)) X1=STGR
      I = IU(IA)+IB-NU(IA)
      CALL  INTPL1(XU,IA,X1,HA1,0,K3,MU,XSPU,ALUI,ALUC,NUSP,SLUPE,EMU,
     1   RI,RO,CHORD,STGR,PITCH,1.,HA,HB,MXBI,MXBO,RM(MXBI),RM(MXBO))
C  CASES 4 AND 5
  740 IAN=IA1+1
      DO 750 IA=IAN,MXBO
  750 IF(NL(IA).LT.IB) GO TO 760
      IF(IB.NE.IBTE1) GO TO 755
      IAN = MXBO
      HAN = HANTE
      KN = 1
      GO TO 770
C  END AT IA = MX
  755 IAN = MX
      HAN = HA
      KN = 0
      GO TO 770
C  END ON LOWER SURFACE
  760 IAN = IA
      IA = IA-1
      KN = 1
      CALL  INTPL1(XL,IA,X1,HAN,1,K4,ML,XSPL,ALLI,ALLC,NLSP,SLLPE,EML,
     1   RI,RO,CHORD,STGR,PITCH,-1.,HA,HB,MXBI,MXBO,RM(MXBI),RM(MXBO))
      GO TO 770
  765 IF(IBTE2.EQ.1000) GO TO 780
      IA1 = MXBO-1
      HA1 = HA1TE
      I = ITE2
      IAN = MX
      HAN = HA
      IB = IBTE2
      KK1 =1
  770 CALL VELOC (U,MPL,IA1,IAN,HA1,HAN,I,IB,MX,NXN,NBBO,JU,JL,KK1,KN,
     1   USP,WX,WXL,MXU,WXL,MXL,NU,NL,AAA)
      IB = IB+1
      I = I+1
      IF(CASE.EQ.5) IB=IB-2
      GO TO (510,550,560,650,710,780),CASE
```

```
  780 CALL SORTXY (MXL,WXL,JL)
      CALL SPLINT (MR,BESP,NRSP,MXU,JU,BEU,AAA)
      CALL SPLINT (MR,BESP,NRSP,MXL,JL,BEL,AAA)
      CALL SPLINT (MR,RMSP,NRSP,MXU,JU,RMU,AAA)
      CALL SPLINT (MR,RMSP,NRSP,MXL,JL,RML,AAA)
      LAST = 0
      DO 790 IA=1,MX
      FIRST = LAST+1
      LAST = FIRST+NL(IA)-NU(IA)
      DO 790 I=FIRST,LAST
  790 WX(I) = -WX(I)/BE(IA)*WTFL
      DO 800 I=1,JU
  800 WXU(I) = -WXU(I)/BEU(I)*WTFL
      DO 810 I=1,JL
  810 WXL(I) = -WXL(I)/BEL(I)*WTFL
C
C  END OF RHO*W-SUB-THETA CALCULATION
C
      IF(ARPRT.LE.0) GO TO 830
      WRITE (6,1630)
      LAST=0
      DO 820 IA=1,MX
      FIRST=LAST+1
      LAST=FIRST+NL(IA)-NU(IA)
  820 WRITE(6,1020)  (WX(I), I=FIRST,LAST)
      WRITE (6,1640)
      WRITE(6,1020)  (MXU(I),WXU(I) , I=1,JU)
      WRITE (6,1650)
      WRITE(6,1020)   (MXL(I),WXL(I)  ,I=1,JL)
C
C   CALCULATE RHO*W AND ANGLES AT INTERIOR POINTS
C
  830 CONTINUE
      LAST = 0
      DO 850 IA=1,MX
      FIRST = LAST+1
      LAST = FIRST+NL(IA)-NU(IA)
      DO 850 I=FIRST,LAST
      V(I) = SQRT(WX(I)**2+WM(I)**2)
      IF(WM(I).EQ.0.) GO TO 840
      BETA(I)  = ATAN(WX(I)/WM(I))*57.29577
      GO TO 850
  840 BETA(I) = 90.
  850 CONTINUE
      IF(ARPRT.LE.0) GO TO 870
      WRITE (6,1660)
      LAST=0
      DO 860 IA=1,MX
      FIRST=LAST+1
      LAST=FIRST+NL(IA)-NU(IA)
  860 WRITE(6,1020) (V(I),I=FIRST,LAST)
C
C   CALCULATE DENSITY AND VELOCITY AT EACH POINT
C
  870 LAST = 0
      RELER = 0.
      IF(INTVEL.GT.0) WRITE (6,1730)
      CPTIP = 2.*CP*TIP
      DO 910 IA=1,MX
```

```
      FIRST = LAST+1
      LAST  = FIRST+NL(IA)-NU(IA)
      TWLMR  = 2.*OMEGA*LAMBDA-(OMEGA*RM(IA))**2
      DO 900 I=FIRST,LAST
      RHOT = RHO(I)
      CALL DENSTY (V(I),RHO(I),VEL,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP,
     1   VTOL)
      RELER  = AMAX1(RELER,ABS((RHOT-RHO(I))/RHO(I)))
  900 V(I) = VEL
      IF(INTVEL.LE.0) GO TO 910
      WRITE(6,1670)IA, (V(I), BETA(I) ,I=FIRST,LAST)
  910 CONTINUE
      IF(ITER.GT.0) ITERA = ITER
      WRITE (6,1735) ITERA,RELER
      ITERA  = ITER+1
      IF(RELER.LT..001.AND.ITER.GT.1) ITER = -1
C
      IF(GAM.EQ.1.5.AND.AR.EQ.1000..AND.TIP.EQ.1.E6)ITER=0
C  CALCULATE SURFACE VELOCITIES BASED ON AXIAL COMPONENTS
C
      IF(SURVEL.GT.0) WRITE (6,1680)
      BETAU(MXBI) = 90.
      BETAL(MXBI) = -90.
      WU(MXBI) = 0.
      WL(MXBI) = 0.
      BETAU(MXBO) = -90.
      BETAL(MXBO) = 90.
      WU(MXBO) = 0.
      WL(MXBO) = 0.
      MXBIP1=MXBI+1
      MXBOM1=MXBO-1
      AAA(MXBI) = 0.
      BBB(MXBI) = 0.
      DO 920 IA=MXBIP1,MXBOM1
      AAA(IA)=AAA(IA-1)+SQRT(HA**2+((XU(IA)-XU(IA-1))*(RM(IA)+RM(IA-1))
     1         /2.)**2)
      TANTHU=DXDZU(IA)*RM(IA)
      BETAU(IA) =ATAN(TANTHU)*57.29577
      WMU(IA)=WMU(IA)*SQRT(1.+TANTHU*TANTHU)
      TWLMR  = 2.*OMEGA*LAMBDA-(OMEGA*RM(IA))**2
      CALL DENSTY (WMU(IA),RHOU(IA),WU(IA),TWLMR,CPTIP,EXPON,RHOIP,GAM,
     1   AR,TIP,VTOL)
      BBB(IA)=BBB(IA-1)+SQRT(HA**2+((XL(IA)-XL(IA-1))*(RM(IA)+RM(IA-1))
     1         /2.)**2)
      TANTHL=DXDZL(IA)*RM(IA)
      BETAL(IA) =ATAN(TANTHL)*59.29577
      WML(IA)=WML(IA)*SQRT(1.+TANTHL*TANTHL)
      CALL DENSTY (WML(IA),RHOL(IA),WL(IA),TWLMR,CPTIP,EXPON,RHOIP,GAM,
     1   AR,TIP,VTOL)
  920 CONTINUE
      IF(SURVEL.LE.0) GO TO 927
      AAA(MXBO)=AAA(MXBOM1)+SQRT(HA**2+((XL(MXBO)-XU(MXBOM1))*(RM(MXBO)
     1            +RM(MXBOM1))/2.)**2)
      BBB(MXBO)=BBB(MXBOM1)+SQRT(HA**2+((XL(MXBO)-XL(MXBOM1))*(RM(MXBO)
     1            +RM(MXBOM1))/2.)**2)
      WRITE (6,1690)  (MPL(IA),WU(IA), BETAL(IA),AAA(IA),WMU(IA),WL(IA),
     1     BETAL(IA),BBB(IA),WML(IA),IA=MXBI,MXBO)
      NP1 =0
      DO 923 IA=MXBI,MXBO
```

```
      IF(ABS(BETAU(IA)).GT.60.) GO TO 923
      NP1 = NP1+1
      YACROS(NP1) = WU(IA)
      XDOWN(NP1) = MPL(IA)
  923 CONTINUE
      NP2 = NP1
      DO 925 IA=MXBI,MXBO
      IF(ABS(BETAL(IA)).GT.60.) GO TO 925
      NP2 = NP2+1
      YACROS(NP2) = WL(IA)
      XDOWN(NP2) = MPL(IA)
  925 CONTINUE
      NP3 = NP2
      NP2 = NP2-NP1
C
C    CALCULATE SURFACE VELOCITIES BASED ON TANGENTIAL COMPONENTS
C
      WRITE (6,1700)
  927 CONTINUE
      CALL BLDDE1(MU,XSPL,SLUPE,EMU,NUSP,RI,ALUI,RO,ALUO,CHORD,STGR,
     1 PITCH,1.,JU,MXU,MXBI,HA,DTDMU,RM(MXBI),RM(MXBO))
      CALL BLDDE1(ML,XSPL,SLLPE,EML,NLSP,RI,ALLI,RO,ALLO,CHORD,STGR,
     1 PITCH,-1.,JL,MXL,MXBI,HA,DTDML,RM(MXBI),RM(MXBO))
C    UPPER SURFACE
      BETAU(1) = 90.
      WXU(1)=ABS(WXU(1))
      TWLMR=2.*OMEGA*LAMBDA-(OMEGA*RMU( 1))**2
      CALL  DENSTY(WXU( 1),RHOUT( 1),WU ( 1),TWLMR,CPTIP,EXPON,RHOIP,
     1    GAM,AR,TIP,VTOL)
      BETAU(JU) = -90.
      WXU(JU)=ABS(WXU(JU))
      TWLMR=2.*OMEGA*LAMBDA-(OMEGA*RMU(JU))**2
      CALL  DENSTY(WXU(JU),RHOUT(JU),WU (JU),TWLMR,CPTIP,EXPON,RHOIP,
     1    GAM,AR,TIP,VTOL)
      JUM1=JU-1
      DO 940 I=2,JUM1
      TANTHU = DTDMU(I)*RMU(I)
      BETAU(I) =ATAN(TANTHU)*57.29577
      IF(TANTHU.EQ.0.) GO TO 930
      WXU(I)= ABS(WXU(I))*SQRT(1.+1./(TANTHU*TANTHU))
      TWLMR  = 2.*OMEGA*LAMBDA-(OMEGA*RMU(I))**2
      CALL DENSTY (WXU(I),RHOUT(I),WU(I),TWLMR,CPTIP,EXPON,RHOIP,GAM,
     1    AR,TIP,VTOL)
      GO TO 940
  930 WU(I) = 0.
  940 CONTINUE
      IF(SURVEL.LE.0) GO TO 947
      WRITE (6,1710)  (MXU(I),WU(I), BETAU(I),WXU(I)  , I=1,JU)
      DO 945 I=1,JU
      IF(ABS(BETAU(I)).LT.30.) GO TO 945
      NP3 = NP3+1
      YACROS(NP3) = WU(I)
      XDOWN(NP3) = MXU(I)
  945 CONTINUE
      NP4 = NP3
      NP3 = NP3-NP2-NP1
C    LOWER SURFACE
      WRITE (6,1720)
```

```
 947 CONTINUE
     DO 960 I=1,JL
     TANTHL = DTDML(I)*RML(I)
     BETAL(I) =ATAN(TANTHL)*57.29577
     IF (TANTHL.EQ.C.) GO TO 950
     WXL(I)=ABS(WXL(I))*SQRT(1.+1./(TANTHL*TANTHL))
     TWLMR = 2.*OMEGA*LAMBDA-(OMEGA*RML(I))**2
     CALL DENSTY (WXL(I),RHOLT(I),WL(I),TWLMR,CPTIP,EXPON,RHOIP,GAM,
    1   AR,TIP,VTOL)
     GO TO 960
 950 WL(I) = 0.
 960 CONTINUE
     IF(SURVEL.LE.0) RETURN
     WRITE (6,1710)  (MXL(I),WL(I), BETAL(I),WXL(I)  , I=1,JL)
     DO 970 I=1,JL
     IF(ABS(BETAL(I)).LT.30.) GO TO 970
     NP4 = NP4+1
     YACROS(NP4) = WL(I)
     XDOWN(NP4) = MXL(I)
 970 CONTINUE
     NP4 = NP4-NP3-NP2-NP1
     KKK(1) = 0
     KKK(2) = 4
     KKK(3) = NP1
     KKK(5) = NP2
     KKK(7) = NP3
     KKK(9) = NP4
     P = 5.
     WRITE (6,1740)
     CALL PLOTMY(XDOWN,YACROS,KKK,P)
     WRITE (6,1750)
     RETURN
1020 FORMAT (1X,8G16.7)
1630 FORMAT (1H1,26HWX ARRAY(RHO*W-SUB-THETA) )
1640 FORMAT (1H0,4(6X,3HMXU,13X,3HWXU,7X)/48H (M COORD. VS. RHO*W-SUB-T
    1HETA ON UPPER SURFACE) )
1650 FORMAT (1H0,4(6X,3HMXL,13X,3HWXL,7X)/48H (M COORD. VS. RHO*W-SUB-T
    1HETA ON LOWER SURFACE) )
1660 FORMAT (34H1ARRAY OF RHO*W AT INTERIOR POINTS )
1670 FORMAT (1H0,3HIA=,I2,5(24H    VELOCITY   ANGLE(DEG))/(3X,
    1   5(G15.4,F9.2)))
1680 FORMAT (1H1,15X,1H*,21X,44HSURFACE VELOCITIES BASED ON AXIAL COMPO
    1NENTS,45X,1H*,/16X,1H*,12X,16HUPPER     SURFACE,25X,1H*,15X,16HLOWE
    2R     SURFACE,25X,1H*,/7X,1HM,8X,1H*,2(3X,8HVELOCITY,3X,43HANGLE(DE
    3G) SURF. LENGTH     RHO*W       *     ))
1690 FORMAT (1H ,G13.4,3H  *,G12.4,F9.2,2G15.4,6H   *    ,G12.4,F9.2,
    1   2G15.4,3H  * )
1700 FORMAT (51H1 SURFACE VELOCITIES BASED ON TANGENTIAL COMPONENTS/
    1   25X,13HUPPER SURFACE/7X,1HM,10X,8HVELOCITY,3X,10HANGLE(DEG),
    2   3X,5HRHO*W)
1710 FORMAT (1H ,2G13.4,F9.2,G15.4)
1720 FORMAT (25X,13HLOWER SURFACE/7X,1HM,10X,8HVELOCITY,3X,
    1   10HANGLE(DEG),3X,5HRHO*W)
1730 FORMAT (1H1,40X,34HVELOCITIES AT INTERIOR MESH POINTS )
1735 FORMAT (14HLITERATION NO., I3,3X,36HMAXIMUM RELATIVE CHANGE IN DEN
    1SITY =,G11.4)
```

```
174C FORMAT (2HPT,5CX,24HBLADE SURFACE VELOCITIES)
175C FORMAT (2HPL,37X,82HVELOCITY(METERS/SECCND)  VS. MERIDIONAL STREAM
    1LINE DISTANCE(METERS) DOWN THE PAGE  /2HPL/
    2   2HPL,5CX,43H* - UPPER SURFACE, BASED CN AXIAL CCMPONENT /
    3   2HPL,5CX,43H+ - LOWER SURFACE, BASED ON AXIAL COMPONENT /
    4   2HPL,50X,48HC - UPPER SURFACE, BASED CN TANGENTIAL CCMPONENT/
    5   2HPL,5CX,48HX - LOWER SURFACE, BASED CN TANGENTIAL CCMPONENT)
     END
```

Subroutine DENSTY

DENSTY calculates the subsonic relative velocity W and corresponding density ρ that result in a given value of the mass flow parameter ρW. This is done using equations (B5) and (B6) which are an algorithm based on Newton's method.

If the value of ρW is too large, there is no solution. In this case an error message is printed out. If a "continue" control card is used (see p. 86), W_{cr} and the corresponding density are calculated as output and the program continues. This makes it possible to get an approximate solution even though there may be one or two points with too large a value for ρW.

The input arguments are as follows:

RHOW	ρW
RHO	initial estimate for $\rho (\rho'_{in}$ may be used)
TWLMR	$2\omega\lambda - (\omega r)^2$
CPTIP	$2c_pT'_{in}$
EXPON	$1/(\gamma - 1)$
RHOIP	ρ'_{in}
GAM	γ
AR	R
TIP	T'_{in}
VTOL	convergence tolerance on relative change in W

The output arguments are as follows:

RHO	ρ
VEL	W

The internal variables are as follows:

RHOT	newly calculated estimate for ρ
RHOWP	$d(\rho W)/d\rho$

TEMP	$(T/T'_{in})^{(2-\gamma)/(\gamma-1)}$
TGROG	$2\gamma R/(\gamma + 1)$
TTIP	T/T'_{in}
VELNEW	newly calculated estimate for W

```
      SUBROUTINE DENSTY(RHOW,RHO,VEL,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP,
     1   VTOL)
      VEL = RHOW/RHO
   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.C.) GO TO 30
      VELNEW = VEL+(RHOW-RHOT*VEL)/RHOWP
      IF(ABS(VELNEW-VEL)/VELNEW.LT. VTOL) GO TO 20
      VEL = VELNEW
      GO TO 10
   20 VEL = VELNEW
      RHO = RHOW/VEL
      RETURN
   30 CALL  ARERR (29H VALUE OF RHO*W IS TOO LARGE$)
      TGROG = 2.*GAM*AR/(GAM+1.)
      VEL = SQRT(TGROG*TIP*(1.-TWLMR/CPTIP))
      RHO = RHOIP*(1.-(VEL**2+TWLMR)/CPTIP)**EXPON
      RETURN
      END
```

Subroutine BLDCR1

BLDCR1 obtains the θ-coordinates and the slopes of the blade surfaces corresponding to the given m-coordinates. BLDCR1 may be used in two ways, either to obtain the information at all vertical mesh lines in one call, as when called by COEF directly, or at a single specified point, as when called by INTPL1. The value of NCH is the number of points at which output is desired. Since BLDCR1 is used for either the upper or lower blade surface, SURF is used as a code to determine which surface is desired. SURF = 1. for the upper surface and SURF = -1. for the lower surface.

The entire blade surface is defined by the leading- and trailing-edge radii and by two cubic spline curves (upper and lower surfaces), which are piecewise cubic polynomials. The procedure then is to scan the spline points to determine which interval the m-coordinate (ZINT) is in, and then to calculate the θ-coordinate and derivative, both of which are specified analytically.

The input arguments are as follows:

Z	array of m-coordinates of spline points for blade surface (upper or lower)
XSP	array of θ-coordinate of spline points for blade surface (upper or lower)
SLOPE	array of slopes at spline points for blade surface (upper or lower)
EM	array of second derivatives at spline points for blade surface (upper or lower)
NSP	number of spline points on blade surface (upper or lower)
RI	see fig. 10 (p. 15)
ALI	either ALUI or ALLI (see fig. 10)
RO	see fig. 10
ALO	either ALUO or ALLO (see fig. 10)
CHORD	see fig. 10
STGR	see fig. 10
PITCH	see fig. 10
SURF	code to indicate upper or lower surface, SURF = 1., for upper surface and, SURF = -1., for lower surface
NCH	number of points for which output is desired
ZINT	used as input only when NCH = 1, then it is m-coordinate for which corresponding θ-coordinate for blade surface is desired
MXBI	same as main program, number of mesh points on line AB
RMI	r at leading edge
RMO	r at trailing edge

The output arguments are as follows:

X	array of θ-coordinates at the vertical mesh lines for blade surface, or if NCH = 1, at m = ZINT
DXDZ	array of slopes at same points as X

The internal variables are as follows:

HA	basic mesh spacing in meridional (m) direction
IA	index of vertical mesh line
IFST	index of first point considered
ILST	index of last point considered

K index of spline point

RMZ difference between m-coordinate of point considered and m-coordinate of center of leading- or trailing-edge radii

SRW integer variable in common used to obtain output useful in debugging; when SRW = 19, BLDCR1 will write out calculated blade coordinates and corresponding slopes

SW coefficient with value of zero on upper blade surface and 1 on lower blade surface; used to add pitch to computed blade coordinate for lower surface only

ZINT m-coordinate at which θ-coordinate and slope of blade surface are required

```
      SUBROUTINE BLDCR1(Z,XSP,SLOPE,EM,NSP,RI,ALI,RC,ALC,CHORD,STGR,
     1   PITCH,SURF,X,NCH,ZINT,MXBI,DXDZ,RMI,RMO)
C
C    SURF = 1.  --   UPPER SURFACE
C    SURF = -1. --   LOWER SURFACE
C
      DIMENSION XSP(NSP),Z(NSP),X(NCH),SLOPE(NSP),EM(NSP),DXDZ(NCH)
      COMMON SRW
      INTEGER SRW
      SW = 0.
      IF(SURF.LT.0.) SW = 1.
      IFST = 1
      ILST = 1
      IF(NCH.EQ.1) GO TO 10
      IFST = MXBI
      ILST = NCH+MXBI-1
      HA = CHORD/FLOAT(NCH-1)
      ZINT = 0.
   10 K = 2
      DO 100 IA=IFST,ILST
   20 IF(ZINT.GT.Z(1)) GO TO 30
      X(IA) = SQRT(ZINT*(2.*RI-ZINT))/RMI*SURF+PITCH*SW
      RMZ = RI-ZINT
      IF(IA.NE.IFST) DXDZ(IA) = RMZ/SQRT(RI**2-RMZ**2)*SURF/RMI
      ZINT = ZINT+HA
      GO TO 100
   30 IF(ZINT.LE.Z(K)) GO TO 50
      IF(K.GE.NSP) GO TO 60
      K = K+1
      GO TO 30
   50 S = Z(K)-Z(K-1)
      X(IA) = EM(K-1)*(Z(K)-ZINT)**3/6./S+EM(K)*(ZINT-Z(K-1))**3/6./S
     1   +(XSP(K)/S-EM(K)*S/6.)*(ZINT-Z(K-1))+(XSP(K-1)/S-EM(K-1)*S/6.)
     2   *(Z(K)-ZINT)
      DXDZ(IA) = -EM(K-1)*(Z(K)-ZINT)**2/2./S+EM(K)*(Z(K-1)-ZINT)**2/2.
     1   /S+(XSP(K)-XSP(K-1))/S-(EM(K)-EM(K-1))*S/6.
      ZINT = ZINT+HA
      GO TO 100
   60 IF ((IA.EQ.NCH).AND.(IA.GT.1)) GO TO 70
      X(IA) = STGR+SURF*SQRT((CHORD-ZINT)*(2.*RO-CHORD+ZINT))/RMO+PITCH
     1   *SW
      RMZ = CHORD-ZINT-RO
```

```
      IF( IA.NE. ILST) DXDZ(IA) = RMZ/SQRT(RC**2-RMZ**2)*SURF/RMO
      ZINT = ZINT+HA
      GO TO 100
   70 X( IA) = STGR+PITCH*SW
  100 CONTINUE
      IF(SRW.EQ.19) WRITE(6,1000) (X(IA),DXDZ(IA),IA=IFST,ILST)
      RETURN
 1000 FORMAT ( 1X,54HINTERPOLATED COORDINATES AND SLOPES COMPUTED BY BLDC
     1RD,/( 5X,2E16.8))
      END
```

Subroutine BLDDE1

BLDDE1 obtains the slopes of the blade at given m-coordinates. It is used by TASVEL to obtain the blade slopes at each horizontal mesh line. BLDDE1 is similar to BLDCR1, except that the m-coordinates are an input array and the θ-coordinates are not given as output.

The input arguments for BLDDE1 are the same as those for BCDCR1, except that ZINT is not input. Also included are

ZX — array of m-coordinates from the line BG in fig. 4 for which slopes for blade surface are desired; these values must be arranged in increasing order

HA — basic mesh spacing in axial direction

The output of BLDDE1 is

DXDZ — array of slopes at m-coordinates in array ZX

The internal variables are as follows:

IA — index of point in ZX array

K — index of spline point

RMZ — difference between m-coordinate of point considered and m-coordinate of center of leading- or trailing-edge radii

SRW — integer variable in COMMON used to obtain output useful in debugging; when SRW = 20, BLDDE1 writes out calculated blade slopes

ZINT — m-coordinate from blade leading edge at which blade slope is desired

```
      SUBROUTINE BLDDE1(Z,XSP,SLOPE,EM,NSP,RI,ALI,RC,ALC,CHORD,STGR,
     1   PITCH,SURF,NCH,ZX,MXBI,HA,DXDZ,RMI,RMO)
C
C    SURF = 1.  --   UPPER SURFACE
C    SURF = -1. --   LOWER SURFACE
C
      DIMENSION XSP(NSP),Z(NSP),SLOPE(NSP),EM(NSP),DXDZ(NCH),ZX(NCH)
      COMMON SRW
      INTEGER SRW
   10 K = 2
      DO 100 IA = 1,NCH
      ZINT = ZX(IA)
   20 IF(ZINT.GT.Z(1)) GO TO 30
      RMZ = RI-ZINT
      IF(IA.NE.1.OR.SURF.LT.0.) DXDZ(IA) = RMZ/SQRT(RI**2-RMZ**2)*SURF
     1   /RMI
      GO TO 100
   30 IF(ZINT.LE.Z(K)) GO TO 50
      IF(K.GE.NSP) GO TO 60
      K = K+1
      GO TO 30
   50 S = Z(K)-Z(K-1)
      DXDZ(IA) = -EM(K-1)*(Z(K)-ZINT)**2/2./S+EM(K)*(Z(K-1)-ZINT)**2/2.
     1   /S+(XSP(K)-XSP(K-1))/S-(EM(K)-EM(K-1))*S/6.
      GO TO 100
   60 RMZ = CHORD-ZINT-RO
      IF((IA.NE.1.AND.IA.NE.NCH).OR.SURF.LT.0.) DXDZ(IA)=RMZ/SQRT(RO**2
     1-RMZ**2)*SURF/RMO
      GO TO 100
  100 CONTINUE
      IF(SRW.EQ.20) WRITE(6,1000) (ZX(IA),DXDZ(IA),IA=1,NCH)
      RETURN
 1000 FORMAT (1X,56HZ COORD. AND INTERPOLATED DERIVATIVES COMPUTED BY BL
     1DDER,/(5X,2E16.8))
      END
```

Subroutine INTPL1

To compute the terms of the matrix A of equation (A7), it is necessary to obtain the distance along a horizontal mesh line from a mesh point near the blade to the blade itself. This is the quantity h_3 or h_4 in equation (A1). This value is computed by INTPL1. Since the equation of the blade surface is known, this amounts to finding the root of an equation. The root is found by INTPL1 by an iterative procedure, sometimes called the method of false position (falsi reguli). The variables shown in figure 16 correspond to those used in INTPL1. After H has been calculated, the actual value of the spline curve (XI) is computed by BLDCR1 and a reduced interval is considered so that the curve still crosses the value X1. Then the procedure is repeated on the smaller interval. A few iterations will determine the value of z for which the spline curve crosses the mesh line, and from this, h_3 or h_4 is determined.

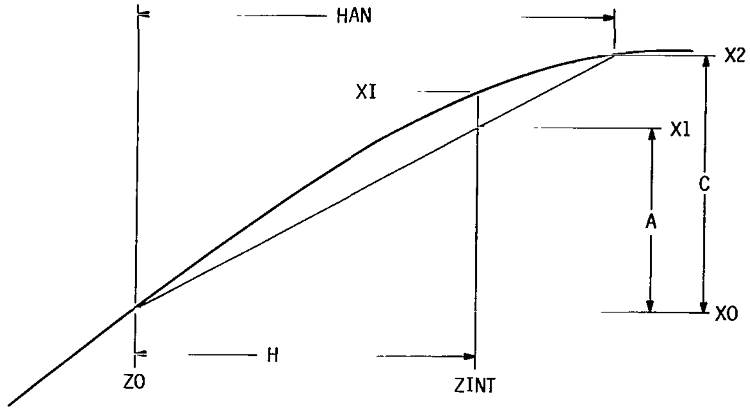


Figure 16. - Notation used in subroutine INTPL1 $H = \frac{A \times HAN}{C}$.

The input arguments are as follows:

HA	basic mesh spacing in axial direction
HB	basic mesh spacing in blade-to-blade direction
IA	index of vertical mesh line on which mesh point lies
MXBI	number of mesh points on line AB
N	integer which is zero when h_3 is to be computed and which is 1 when h_4 is to be computed
X	array of blade θ-coordinates at each vertical mesh line
X1	θ-coordinate of mesh point considered

The remaining input arguments are transmitted to BLDCR1. Their definitions are included in the description of this subroutine.

The output variables are as follows:

H	horizontal distance from mesh point to blade, which is h_3 or h_4 of fig. 17
K	real code variable changed to 1 by INTPL1

The internal variables are as follows:

A	X1 - X0 (see fig. 16)
C	X2 - X0 (see fig. 16)
H	distance from Z0 to approximate root ZINT determined by linear interpolation (see fig. 16)

HAN	length of interval in which root has been determined to lie (see fig. 16)
IAPN	index of vertical mesh line to right of interval
IAPNM1	index of vertical mesh line to left of interval
SRW	integer variable in COMMON used to obtain output useful in debugging; when SRW = 19, values of IA, N, HA, X1, X2, X0, and Z0 are printed to start, followed by values of H, XI, X1, Z0, and ZBASE for each iteration, and final value of H after convergence
XI	θ-coordinate of blade computed by BCDCR1 for m = ZINT (see fig. 16)
X0	θ-coordinate of blade at left end of interval (see fig. 16)
X2	θ-coordinate of blade at right end of interval (see fig. 16)
ZBASE	m-coordinate of left end of interval for first iteration
ZINT	m-coordinate determined by linear interpolation (see fig. 16)
Z0	m-coordinate of left end of interval (see fig. 16)

```
      SUBROUTINE INTPL1(X,IA,X1,H,N,K,Z,XSP,ALI,ALO,NSP,SLOPE,EM,RI,RO,
     1   CHORD,STGR,PITCH,SURF,HA,HB,MXBI,MXBO,RMI,RMO)
      DIMENSION X(100),Z(NSP),XSP(NSP),SLOPE(NSP),EM(NSP)
      COMMON SRW
      INTEGER SRW
      REAL K
      K = 1.
      IAPNM1 = IA+N-1
      IAPN = IA+N
      X0 = X (IAPNM1)
      X2 = X (IAPN)
      HAN = HA
      Z0 = FLOAT(IAPNM1-MXBI)*CHORD/FLOAT(MXBO-MXBI)
      IF(IA.EQ.MXBO) Z0 = CHORD-HA
      H=HA
      IF(IAPN.EQ.MXBO.AND.SURF.LE.0..AND.N.NE.0) X2=X2-RO
      IF(SRW.EQ.19) WRITE(6,1010) IA,N,HA,X1,X2,X0,Z0
      IF(Z0.LT.0..OR.Z0.GT.(CHORD-.001*HA)) RETURN
      ZBASE = Z0
      IF(ABS(X1-X0).GT.(.001*HB)) GO TO 10
      IF(IA.EQ.MXBI+1) GO TO 15
      A = 0.
      C = H
      GO TO 20
   15 H = 2.*RI
      A = -1.
      GO TO 25
   10 A = X1-X0
      C = X2-X0
   20 H = A/C*HAN
   25 CONTINUE
      IF(SRW.EQ.19) WRITE(6,1020) H,XI,X1,Z0,ZBASE
      ZINT = Z0+H
      CALL BLDCR1 (Z,XSP,SLOPE,EM,NSP,RI,ALI,RO,ALO,CHORD,STGR,PITCH,
```

```
     1   SURF,XI,1,ZINT,MXBI,DXDZ,RMI,RMO)
      IF(ABS(XI-X1).LE.(HB*.001)) GO TO 40
      IF(A*(XI-X1).LT.0.) GO TO 30
      HAN = H
      X2 = XI
      GOTO10
   30 HAN = HAN-H
      XO = XI
      ZO = ZC+H
      GO TO 10
   40 H = ZO+H-ZBASE
      IF(N.EQ.0) H = HA-H
      IF(SRW.EQ.19) WRITE(6,1000) H
      RETURN
 1000 FORMAT (1X,22HH AS COMPUTED BY INTPL /(5X,5E16.8))
 1010 FORMAT (1X,4HIA =,I4,5X,3HN =,I4,5X,4HHA =,E14.6,5X,4HX1 =,E14.6,
     1   4X,4HX2 =,E14.6,4X,4HXO =,E14.6,4X,4HZO =,E14.6)
 1020 FORMAT (1X,3HH =,E14.6,5X,4HXI =,E14.6,5X,4HX1 =,E14.6,5X,4HZO =,
     1   E14.6,5X,7HZBASE =,E14.6)
      END
```

Subroutine VELOC

The partial derivatives $\partial u/\partial m$ along each horizontal mesh line are calculated by VELOC. This subroutine is described in reference 5. There are no changes in the subroutine, however WX refers here to $\partial u/\partial m$ rather than W_θ.

```
      SUBROUTINE VELOC(U,ZPL,IA1,IAN,HA1,HAN,I,IB,MX,NXN,NBBO,JU,JL,
     1   KK1,KN,USP,WX,WXU,ZXU,WXL,ZXL,NU,NL,AAA)
      DIMENSION U(NXN),ZPL(100),WX(NXN),WXL(100),ZXU(100),ZXL(100),
     1   ZA(100),USP(100),WXSP(100),WXL(100),NU(100),NL(100),AAA(100)
C KK1 OR KN = 1, LOWER SURFACE
C KK1 OR KN = 0, UPPER SURFACE
      I1 = I
      IA2 = IA1+1
      IANM1 = IAN-1
      ZA(IA1) = ZPL(IA2)-HA1
      ZA(IAN) = ZPL(IANM1)+HAN
      NSP = IAN-IA1+1
      DO 10 IA=IA2,IANM1
      USP(IA) = U(I1)
      I1=I1+NL(IA)-NU(IA+1)+1
   10 ZA(IA) = ZPL(IA)
      I1 = NXN+IB-NBBO
      USP(IA1)=0.0
      USP(IAN) = 0.
      IF(IA1.EQ.1) USP(1) = U(IB+1)
      IF(KK1.NE.0) USP(IA1) =1.0
      IF(IAN.EQ.MX) USP(IAN) = U(I1)
      IF(KN.NE.0) USP(IAN) = 1.
      CALL SPLINE (ZA(IA1),USP(IA1),NSP,WXSP(IA1),AAA)
      I1 = I
      DO 20 IA=IA2,IANM1
      WX(I1) = WXSP(IA)
```

```
   20 I1 = I1+NL(IA)-NU(IA+1)+1
C  TAKE CARE OF FIRST POINT
      IF(IA1.NE.1) GO TO 30
      WX(IB+1) = WXSP(IA1)
      GO TO 50
   30 IF(KK1.NE.0) GO TO 40
      JU = JL+1
      WXU(JU) = WXSP(IA1)
      ZXU(JU) = ZA(IA1)
      GO TO 50
   40 JL = JL+1
      WXL(JL) = WXSP(IA1)
      ZXL(JL) = ZA(IA1)
C  TAKE CARE OF LAST POINT
   50 IF(IAN.NE.MX) GO TO 60
      I1 = NXN+IB-NBBO
      WX(I1) = WXSP(IAN)
      RETURN
   60 IF(KN.NE.0) GO TO 70
      JU = JL+1
      WXU(JU) = WXSP(IAN)
      ZXU(JU) = ZA(IAN)
      RETURN
   70 JL = JL+1
      WXL(JL) = WXSP(IAN)
      ZXL(JL) = ZA(IAN)
      RETURN
      END
```

Subroutines SPLINE, SPLN22, SPLINT, SORTXY

These subroutines are all described in reference 5.

```
      SUBROUTINE SPLINE (X,Y,N,SLOPE,EM)
      DIMENSION X(N),Y(N),EM(N),SLOPE(N)
      COMMON Q/BOX/S(100),A(100),B(100),C(100),F(100),W(100),SB(100),
     1G(100)
      INTEGER Q
      DO 10 I=2,N
   10 S(I)=X(I)-X(I-1)
      NO=N-1
      IF(NO.LT.2) GO TO 25
      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)
   25 A(N) =-1.0
      B(1)=1.
      B(N)=1.
      C(1)=-1.0
      F(1)=0.
      F(N)=0.
```

```
      W( 1)=B( 1)
      SB( 1)=C( 1 )/W( 1)
      G( 1)=C .
      DO 30  I=2,N
      W( I )=B( I )-A( I )*SB ( I-1 )
      SB( I )=C( I )/W( I )
   30 G( I )=( F( I )-A( I )*G( I-1 ) )/W( I )
      EM( N )=G( N )
      DO 40  I=2,N
      K=N+1- I
   40 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 )
      DO50  I=2,N
   50 SLOPE( I )=S( I )/6.*( 2.*EM( I )+EM( I-1 ) )+( Y( I )-Y( I-1 ) )/S( I )
      IF ( Q.EQ.13)  WRITE  (6,100)  N,( X( I ) ,Y( I ) ,SLOPE( I ) ,EM( I ) ,I=1,N)
  100 FORMAT ( 2X15HNO. OF POINTS =I3/10X5HX     15X5HY     15X5HSLOPE15X5H
     1EM    /( 4F20.8 ) )
      RETURN
      END
```

```
      SUBROUTINE SPLN22 ( X,Y,Y1P,YNP,N,SLOPE,EM)
      DIMENSION  X( 50 ),Y( 50 ),S( 50 ),A( 50 ),B( 50 ),C( 50 ),F( 50 ),W( 50 ),SB( 50 ),
     1G( 50 ), EM( 50 ), SLOPE( 50 )
      COMMON  Q
      INTEGER  Q
      DO  10  I=2,N
   10 S( I )=X( I )-X( I-1 )
      NO=N-1
      IF( NO .LT. 2 )  GO  TO  25
      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 )
   25 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  30  I=2,N
      W( I )=B( I )-A( I )*SB( I-1 )
      SB( I )=C( I )/W( I )
   30 G( I )=( F( I )-A( I )*G( I-1 ) )/W( I )
      EM( N )=G( N )
      DO  40  I=2,N
      K=N+1- I
```

```
 40 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)
    DO50 I=2,N
 50 SLOPE(I)=S(I)/6.*(2.*EM(I)+EM(I-1))+(Y(I)-Y(I-1))/S(I)
    IF (Q.EQ.18) WRITE (6,100) N,(X(I),Y(I),SLOPE(I),EM(I),I=1,N)
    RETURN
100 FORMAT (2X15HNO. OF POINTS =I3/10X5HZ     15X5HX     10X10HDERIVATIV
   1E10X10H2ND DERIV./(4G20.8))
    END
```

```
    SUBROUTINE SPLINT (X,Y,N,Z,MAX,YINT,DYDX)
    DIMENSION   X(N),Y(N),Z(MAX),YINT(MAX),DYDX(MAX)
    COMMON Q/BOX/S(50),A(50),B(50),C(50),F(50),W(50),SB(50),G(50),
   1EM(400)
    INTEGER Q
    III = Q
    DO 10 I=2,N
 10 S(I)=X(I)-X(I-1)
    NO=N-1
    IF(NO.LT.2) GO TO 25
    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)
 25 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 30 I=2,N
    W(I)=B(I)-A(I)*SB(I-1)
    SB(I)=C(I)/W(I)
 30 G(I)=(F(I)-A(I)*G(I-1))/W(I)
    EM(N)=G(N)
    DO 40 I=2,N
    K=N+1-I
 40 EM(K)=G(K)-SB(K)*EM(K+1)
    DO 90 I=1,MAX
    K=2
    IF(Z(I)-X(1)) 60,50,70
 50 YINT(I)=Y(1)
    GO TO 87
 60 IF(Z(I).GE.(1.1*X(1)-.1*X(2))) GO TO 85
    WRITE (6,1000) Z(I)
    Q = 16
    GO TO 85
```

```
1000 FORMAT (17H OUT OF RANGE Z =F10.6)
  65 K=N
     IF(Z(I).LE.(1.1*X(N)-.1*X(N-1))) GO TO 85
     WRITE (6,1000) Z(I)
     Q = 16
     GO TO 85
  70 IF(Z(I)-X(K)) 85,75,80
  75 YINT(I)=Y(K)
     GO TO 87
  80 K=K+1
     IF(K-N) 70,70,65
  85 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))
  87 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
  90 CONTINUE
     MXA = MAXO(N,MAX)
     IF(Q.EQ.16) WRITE(6,1010) N,MAX,(X(I),Y(I),Z(I),YINT(I),DYDX(I),
    1    I=1,MXA)
     Q = III
1010 FORMAT (2X21HNO. OF POINTS GIVEN =,I3,30H, NO. OF INTERPOLATED POI
    1NTS =, I3,/10X5HX     15X5HY     12X11HX-INTERPOL.9X11HY-INTERPOL.
    2    8X14HDYDX-INTERPOL./(5E20.8))
 100 RETURN
     END
```

```
     SUBROUTINE SORTXY(X,Y,NPTS)
     DIMENSION X(100),Y(100)
 100 N=NPTS
 102 NN=N-1
 104 DO 140 KT=1,NN
     XMIN=X(KT)
     JAD=KT
     JKL=KT+1
 112 DO 120 JK=JKL,N
 114 IF (XMIN-X(JK)) 120,120,116
 116 XMIN=X(JK)
 118 JAD=JK
 120 CONTINUE
 122 YMIN=Y(JAD)
     X(JAD)= X(KT)
     Y(JAD)= Y(KT)
     X(KT)= XMIN
     Y(KT)=YMIN
 140 CONTINUE
      RETURN
     END
```

Lewis Library Subroutines TIME1, ARERR, and DEDERR

These three subroutines are part of the Lewis Systems Library. TIME1 gives the time in clock pulses of 1/60 of a second. To get elapsed time in minutes, the clock must be read twice and the difference divided by 3600. TIME1 may be replaced by a user's clock reading subroutine, or it may be removed from the program.

DEDERR and ARERR are error subroutines which return control to the monitor after printing out an error message and a trace back giving the location at which the error occurred. ARERR has the additional feature that it can be overriden by using a "continue" control card at the beginning of the deck. The error message is still printed. These subroutines should be replaced by a similar type of error return subroutine at the user's installation.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, November 17, 1967,
720-03-01-35-22.

APPENDIX A

FINITE-DIFFERENCE APPROXIMATION

An approximate numerical solution for the stream function u can be obtained by finite-difference methods. These methods involve first establishing a rectangular grid of mesh points in the region as shown in figure 12 (p. 16). Then at each point where the value of the stream function is unknown, a finite-difference approximation to equation (1) can be written. Adjacent to the boundary, the boundary conditions are included. If there are n unknown values, n nonlinear equations are obtained in n unknowns. The equations are nonlinear since the coefficients involve the density, which depends on the solution. The equations may be solved by an iterative procedure.

First, the inlet absolute total density is used for determining the coefficients. This results in n linear equations. These linear equations may be solved iteratively by successive overrelaxation as described in reference 5. There are two major levels of iteration in the solution. The inner iteration consists of the iterative solution of n linear equations by successive overrelaxation. This solution is an approximate solution of equation (1) for the stream function. This approximate solution may be differentiated numerically and approximate velocities obtained from equations (2) and (3). The approximate velocities are then used to obtain a better approximation to the density at each point, and the coefficients of equation (1) are recalculated using new densities. Thus, the solution to the nonlinear equation (1) is approached by a sequence of solutions to linear equations.

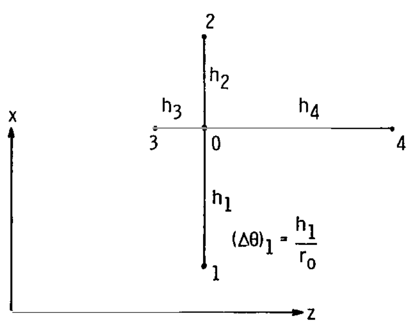


Figure 17. - Notation for adjacent mesh points and mesh spaces.

A typical mesh point with the numbering used to indicate neighboring mesh points is shown in figure 17. The value of the stream function or the other variables at 0 is denoted by using the subscript 0, and similarly for the neighboring points. It can be shown (ref. 13) that equation (1) can be approximated by

$$\left[\frac{2u_1}{h_1(h_1 + h_2)} + \frac{2u_2}{h_2(h_1 + h_2)} - \frac{2u_0}{h_1h_2}\right] + \left[\frac{2u_3}{h_3(h_3 + h_4)} + \frac{2u_4}{h_4(h_3 + h_4)} - \frac{2u_0}{h_3h_4}\right] - \frac{1}{\rho_0}\left(\frac{\rho_2 - \rho_1}{h_1 + h_2}\right)\left(\frac{u_2 - u_1}{h_1 + h_2}\right)$$

$$+\left[\frac{\sin \alpha_0}{r_0} - \frac{b_4\rho_4 - b_3\rho_3}{b_0\rho_0(h_3 + h_4)}\right]\left(\frac{u_4 - u_3}{h_3 + h_4}\right) = \frac{2\omega}{w} b_0\rho_0 \sin \alpha_0 \qquad \textbf{(A1)}$$

where $h_1 = r_0(\Delta\theta)_1$ and $h_2 = r_0(\Delta\theta)_2$ (since $r_0 = r_1 = r_2$). For setting up our equations for solution, the coefficients of the u_i in equation (A1) must be calculated. This was done by expressing equation (A1) as

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

where

$$\left.\begin{aligned}
a_{12} &= \frac{2}{h_1h_2} \\
a_{34} &= \frac{2}{h_3h_4} \\
a_0 &= a_{12} + a_{34} \\
b_{12} &= \frac{\rho_2 - \rho_1}{\rho_0(h_1 + h_2)} \\
b_{34} &= \frac{b_4\rho_4 - b_3\rho_3}{b_0\rho_0(h_3 + h_4)} - \frac{\sin \alpha_0}{r_0} \\
a_1 &= \frac{1}{a_0(h_1 + h_2)}\left(\frac{2}{h_1} + b_{12}\right) \\
a_2 &= \frac{a_{12}}{a_0} - a_1 \\
a_3 &= \frac{1}{a_0(h_3 + h_4)}\left(\frac{2}{h_3} + b_{34}\right) \\
a_4 &= \frac{a_{34}}{a_0} - a_3 \\
k_0 &= -\frac{2\omega}{w} b_0\rho_0 \sin \alpha_0
\end{aligned}\right\} \qquad \textbf{(A2)}$$

This equation can be used at all interior mesh points, and for mesh points adjacent to the blade surfaces BC or FG.

Along the boundary where the value of u is unknown, the equation will vary. For example, along the upstream boundary, $\partial u/\partial \eta$ is known, and a finite-difference approximation to $(\partial u/\partial \eta)_{in}$ in equation (5) gives

$$u_0 = u_4 + h_4\left(\frac{\partial u}{\partial \eta}\right)_{in} = u_4 + h_4\left(\frac{\tan \beta_{in}}{sr_{in}}\right) \tag{A3}$$

Similarly, along the downstream boundary,

$$u_0 = u_3 + h_3\left(\frac{\partial u}{\partial \eta}\right)_{out} = u_3 - h_3\left(\frac{\tan \beta_{out}}{sr_{out}}\right) \tag{A4}$$

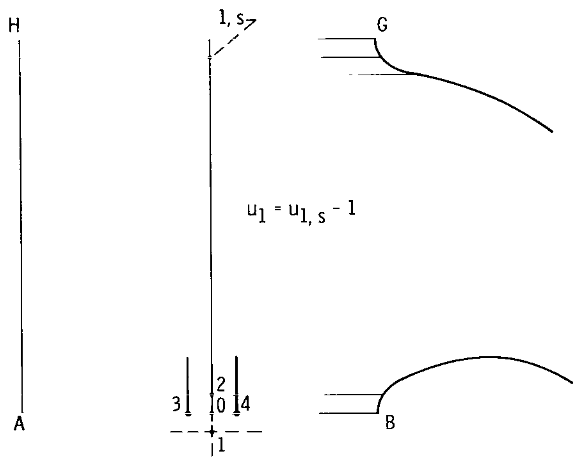


Figure 18. - Mesh point on line AB.

For the points along AB and CD equations can be derived by using the periodic boundary condition. If the point 0 (fig. 18) is on the boundary between A and B, the point 1 is outside the boundary. However, it is known that $u_1 = u_{1,s} - 1$ where the point 1, s is a distance s above point 1 in the θ direction, as shown in figure 18. Substituting this condition in equation (A2) gives

$$u_0 = a_1 u_{1,s} + \sum_{i=2}^{4} a_i u_i - a_1 + k_0 \tag{A5}$$

where the a_i are the same as defined in equation (A2). Of course, equation (A5) holds along CD also.

The points along GH need not be considered, since they are just 1 greater than the corresponding point along AB. The equation for the first mesh line below HG must be modified. In this case $u_2 = u_{2,-s} + 1$, where the point $2,-s$ is a distance s below point 2 in the negative θ direction, as indicated in figure 19. Substituting this condition in equation (A2) gives

$$u_0 = a_1u_1 + a_2u_{2,-s} + a_3u_3 + a_4u_4 + a_2 + k_0 \tag{A6}$$

Equation (A6) also applies to the first mesh line below FE.

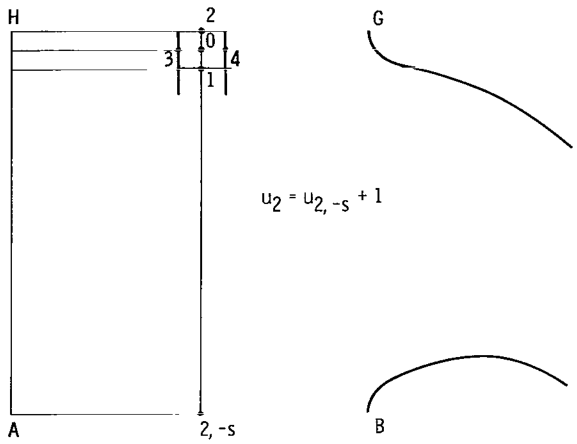


Figure 19. - Mesh point on first line below HG.

One of equations (A2) to (A6) can be applied to each mesh point for which the stream function is unknown in the region of interest, giving the same number of equations as there are unknowns. These points where the stream function is unknown will be referred to simply as unknown mesh points.

This system of n equations is represented in matrix form as

$$A\underline{u} = \underline{k} \tag{A7}$$

where $\underline{u} = (u_1, \ldots, u_n)^T$ is a vector whose components are the unknown values of the stream function, A is the coefficient matrix of equations (A2) to (A6), and $\underline{k} = (k_1, \ldots, k_n)^T$ is the vector whose components are the known constants of equations (A2) to (A6). If the mesh size is sufficiently small, the coefficients, a_1 to a_4 in equation (A2) will be all positive (for any given continuous function b and ρ). In this case, the coefficient matrix A is irreducibly diagonally dominant, and there is a unique solution to equation (A2) (ref. 13).

The solution to equation (A2) is obtained by using two levels of iteration. The inner iteration consists of solving (A2) using fixed values of ρ based on the previous inner iteration. The inner iteration is successive overrelaxation using an optimum overrelaxation factor Ω, as described in reference 5 (p. 77). The iterative procedure is given by

$$u_i^{m+1} = u_i^m + \Omega \left(- \sum_{j=1}^{i-1} a_{ij} u_j^{m+1} - \sum_{j=i+1}^{n} a_{ij} u_j^m + k_i - u_i^m \right)$$

$$\text{for } i = 1, 2, \ldots, n \qquad \text{(A8)}$$

where Ω is the overrelaxation factor. The a_{ij} are the elements of the matrix A, and the k_i are the components of the vector $\underline{k}$ of equation (A7). The u_i^o are the initial estimates of the u_i and are obtained from the previous inner iteration. The optimum value of Ω can be determined as described in reference 5, appendix B. The optimum value of Ω will vary slightly each time the coefficients are corrected; however, the change is usually small, and it has been adequate to use the same overrelaxation factor for the entire calculation.

APPENDIX B

NUMERICAL TECHNIQUES USED IN PROGRAM

Calculation of Velocity and Density

When the stream function u has been calculated, it is possible to then calculate the derivatives $\partial u/\partial m$ and $\partial u/\partial \theta$ by numerical techniques. Then, with equations (2) and (3), and since $W^2 = W_m^2 + W_\theta^2$, values for ρW can be calculated. It is assumed that the values of ω, λ, r, γ, c_p, T_{in}', and ρ_{in}' are all fixed and known. Then ρ, and hence ρW, is a function of W. The product ρW has its maximum value when $W = W_{cr}$. If ρW is less than this maximum value, there are two values of W which will give this value of ρW, one being subsonic and the other supersonic. It is desired to find the subsonic value of W corresponding to the given value of ρW. The method used in Newton's method, which converges quadratically.

It is necessary to express ρW as a function of W. Since

$$V^2 = W^2 + 2\omega\lambda - (\omega r)^2$$

we have

$$\frac{T}{T_{in}'} = 1 - \frac{W^2 + 2\omega\lambda - (\omega r)^2}{2\ c_p T_{in}'} \tag{B1}$$

With the assumption of isentropic flow

$$\frac{\rho}{\rho_{in}'} = \left(\frac{T}{T_{in}'}\right)^{\frac{1}{\gamma - 1}} \tag{B2}$$

hence,

$$\rho W = \rho_{in}' \left[1 - \frac{W^2 + 2\omega\lambda - (\omega r)^2}{2\ c_p T_{in}'}\right]^{\frac{1}{\gamma - 1}} W \tag{B3}$$

For Newton's method, the derivative with respect to W is needed,

$$\frac{d(\rho W)}{dW} = -\frac{W^2\rho'_{in}}{\gamma RT'_{in}}\left[1 - \frac{W^2 + 2\omega\lambda - (\omega r)^2}{2\ c_pT'_{in}}\right]^{\frac{2-\gamma}{\gamma-1}} + \rho'_{in}\left[1 - \frac{W^2 + 2\omega\lambda - (\omega r)^2}{2\ c_pT'_{in}}\right]^{\frac{1}{\gamma-1}} \tag{B4}$$

Suppose that $(\rho W)_{giv}$ is a given value of ρW. A first estimate of W is

$$W_0 = \frac{(\rho W)_{giv}}{\rho'_{in}} \tag{B5}$$

Then, using Newton's method,

$$W_{n+1} = W_n + \frac{(\rho W)_{giv} - \rho(W_n)W_n}{\left.\frac{d(\rho W)}{dW}\right|_{W=W_n}} \qquad n = 0,\ 1,\ 2,\ .\ .\ . \tag{B6}$$

Since the convergence is quadratic only a few iterations are needed and the relative change in W_n is an excellent measure of the relative error in W_n. If an estimate for W is available from a previous iteration, then this value is used for W_0 instead of using equation (B5). The algorithm given by equation (B6) is done by subroutine DENSTY.

Calculation of λ

The input information for the program determines the value of $\lambda = \left(rV_\theta\right)_{in}$. The value of $(\rho W)_{in}$ can be calculated by

$$(\rho W)_{in} = \frac{w}{r_{in}sb_{in} \cos \beta_{in}} \tag{B7}$$

since there is assumed to be uniform flow across AH. The value of W can be estimated by dividing this value of $(\rho W)_{in}$ by ρ'_{in}. Then λ can be estimated by

$$\lambda = r_{in}(W_{in} \sin \beta_{in} + \omega r_{in}) \tag{B8}$$

and from this a better value of ρ is calculated by

$$\rho = \rho_{in}' \left[1 - \frac{W^2 + 2\omega\lambda - (\omega r)^2}{2\, c_p T_{in}'}\right]^{\frac{1}{\gamma - 1}} \tag{B9}$$

Use of this value of ρ gives a better estimate of the value of W, and then iteration can be used with equations (B8) and (B9) until there is a negligible change in ρ. This calculation also gives the value of W along AH. These calculations are performed in INPUT, after reading all input cards.

Calculation of W_{cr}

For reference the critical relative velocity W_{cr} is calculated at blade leading and trailing edges. This is given by

$$W_{cr}^2 = \frac{2\gamma R}{\gamma + 1} T'' \tag{B10}$$

where

$$T'' = T_{in}' - \frac{2\omega\lambda - (\omega r)^2}{2\, c_p} \tag{B11}$$

This calculation is performed by COEF after reading the input cards.

Calculation of Maximum Value of Mass Flow Parameter ρW

The mass flow parameter ρW attains its maximum value when $W = W_{cr}$. For reference, the maximum values of ρW along AH and along DE are computed by the program. The maximum value of ρW is calculated by

$$(\rho W)_{max} = \rho_{in}' \left[1 - \frac{W_{cr}^2 + 2\omega\lambda - (\omega r)^2}{2\, c_p T_{in}'}\right]^{\frac{1}{\gamma - 1}} W_{cr} \tag{B12}$$

where W_{cr} is calculated by equations (B10) and (B11).

Calculation of β at Leading and Trailing Edges

If the radius r or streamsheet thickness b is not constant, β may change for freestream conditions. At the inlet, the hypothetical freestream angle β may be calculated by

$$\tan \beta = \frac{(\lambda - \omega r^2)}{w} \rho b s \tag{B13}$$

Equation (B13) may be derived from the following relations for uniform flow in the θ direction:

$$\left.\begin{aligned} \tan \beta &= \frac{W_\theta}{W_m} \\ W_\theta &= V_\theta - \omega r \\ \lambda &= r V_\theta \\ W_m &= \frac{w}{\rho b r s} \end{aligned}\right\} \tag{B14}$$

At the exit rV_θ is constant ($\neq \lambda$) and the other relations in equation (B14) hold. Using this gives

$$\tan \beta = \left[\frac{\tan \beta_*}{b_*}\left(\frac{\rho}{\rho_*}\right) + \frac{\omega(r_*^2 - r^2)\rho s}{w}\right] b \tag{B15}$$

for the freestream angle β where * denotes values at some reference coordinate of $m = m_*$.

Equation (B15) may be used at either inlet or outlet to calculate β_{in} or β_{out}. In the program, equation (B13) is used to calculate the freestream angle β at the leading edge BG, and equation (B15) is used to calculate the freestream angle β at the trailing edge CF. In the program ρ_{in} is used for ρ in equation (B13) and ρ_{out} is used in equation (B15) since there is little change in density in the freestream region.

Equation for Leading- and Trailing-Edge Radii

The equation for the leading- and trailing-edge radii is needed. If the radius r were constant,

$$(m - m_*)^2 + r^2(\theta - \theta_*)^2 = R^2 \tag{B16}$$

where R is the leading- or trailing-edge radius and (m_*, θ_*) are the coordinates of the center of the radius. Since r changes by a relatively small amount on this circle, it was deemed adequate to use this equation with r taken at the leading or trailing edge. Equation (B16) is used by the program to calculate coordinates on the leading- and trailing-edge radii. It is also used to calculate the points of tangency to the spline curves describing the rest of the blade surfaces, and to calculate slopes on the leading- and trailing-edge radii.

Calculation of Surface Length

It is often desired to plot the velocities as a function of blade surface length. For convenience, the approximate blade surface length is calculated by the program. The calculation is based on straight line distances between each vertical grid line on the blade surface. If h_A is the spacing between vertical grid lines, r_i the radius at the i^{th} vertical grid line, and θ_i the coordinate of the i^{th} vertical grid line, the surface length S_n to the n^{th} grid line is approximately

$$S_n = \sum_{i=2}^{n} \sqrt{h_A^2 + \left(\theta_i - \theta_{i-1}\right)^2 \left(\frac{r_i + r_{i+1}}{2}\right)^2} \tag{B17}$$

This may be in error near the leading or trailing edge, but is quite accurate over most of the blade surfaces.

REFERENCES

1. Stanitz, John D.: Two-Dimensional Compressible Flow in Turbomachines With Conic Flow Surfaces. NACA TR 935, 1949.
2. Stanitz, John D.; and Ellis, Gaylord O.: Two-Dimensional Compressible Flow in Centrifugal Compressors With Straight Blades. NACA TR 954, 1950.
3. Kramer, J. J.: Analysis of Incompressible, Non-viscous Blade-to-Blade Flow in Rotating Blade Rows. Trans. ASME, vol. 80, no. 2, Feb. 1958, pp. 263-275.
4. Kramer, James J.; Stockman, Norbert O.; and Bean, Ralph J.: Non viscous Flow Through a Pump Impeller on a Blade-to-Blade Surface of Revolution. NASA TN D-1108, 1962.
5. Katsanis, Theodore: A Computer Program For Calculating Velocities and Streamlines for Two-Dimensional, Incompressible Flow in Axial Blade Rows. NASA TN D-3762, 1967.
6. Vavra, Michael H.: Aero-Thermodynamics and Flow in Turbomachines. John Wiley and Sons, Inc., 1960.
7. Katsanis, Theodore: Use of Arbitrary Quasi-Orthogonals for Calculating Flow Distribution in the Meridional Plane of a Turbomachine. NASA TN D-2546, 1964.
8. Katsanis, Theodore: Use of Arbitrary Quasi-Orthogonals for Calculating Flow Distribution on a Blade-to-Blade Surface in a Turbomachine. NASA TN D-2809, 1965.
9. Whitney, Warren J.; Szanca, Edward M.; Moffitt, Thomas P.; and Monroe, Daniel E.: Cold-Air Investigation of a Turbine for High-Temperature-Engine Application. I. Turbine Design and Overall Stator Performance. NASA TN D-3751, 1967.
10. 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.
11. Mechtly, E. A.: The International System of Units. Physical Constants and Conversion Factors. NASA SP-7012, 1964.
12. Dellner, Lois T.: A Set of FORTRAN IV Subroutines for Generating Printed Plots. NASA TM X-1419, 1967.
13. Varga, Richard S.: Matrix Iterative Analysis. Prentice-Hall, Inc., 1962.

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

—National Aeronautics and Space Act of 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

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

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

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

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

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

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

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

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

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546