NASA TECHNICAL NOTE

NASA TN D-5427

NASA TN D-5427

CASE FILE COPY

FORTRAN PROGRAM FOR CALCULATING TRANSONIC VELOCITIES 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. • SEPTEMBER 1969

1. Report No. NASA TN D-5427	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle FORTRAN PROGRAM FOR CALCULATING TRANSONIC VELOCITIES ON A BLADE-TO-BLADE STREAM SURFACE OF A TURBOMACHINE		5. Report Date September 1969
		6. Performing Organization Code
7. Author(s) Theodore Katsanis		8. Performing Organization Report No. E-5046
9. Performing Organization Name and Address Lewis Research Center National Aeronautics and Space Administration Cleveland, Ohio 44135		10. Work Unit No. 720-03
		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Note
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

A method has been developed to obtain a transonic flow solution on a blade-to-blade surface of a turbomachine. The blade may be fixed or rotating. The flow may be axial, radial, or mixed. The results include velocities on the blade surface and throughout the passage. The transonic solution is obtained by a velocity-gradient method, using information obtained from a finite-difference stream-function solution at a reduced weight flow.

17. Key Words (Suggested by Author(s))	18. Distribution Statement Unclassified - unlimited		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 96	22. Price* $3.00

*For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151

CONTENTS

FORTRAN PROGRAM FOR CALCULATING TRANSONIC VELOCITIES ON A BLADE-TO-BLADE STREAM SURFACE OF A TURBOMACHINE

by Theodore Katsanis

Lewis Research Center

SUMMARY

A method has been developed to obtain a transonic flow solution on a blade-to-blade surface between blades of a turbomachine. A FORTRAN IV computer program has been written based on this method. The flow must be essentially subsonic, but there may be locally supersonic flow. The solution is two-dimensional, isentropic, and shock free. The blades may be fixed or rotating. The flow may be axial, radial, or mixed, and there may be a change in stream channel thickness in the through-flow direction. A loss in relative stagnation pressure may be accounted for.

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

The transonic solution is obtained by a combination of a finite-difference, stream-function solution and a velocity-gradient solution. The finite-difference solution at a reduced weight flow provides information needed to obtain a velocity-gradient solution.

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

INTRODUCTION

Two useful techniques for calculating blade surface velocities are the velocity-gradient (stream filament) method and the finite-difference solution of the stream-function equation. Each has advantages and limitations. In particular, the finite-difference solution of the stream-function equation (e.g., ref. 1) is limited to strictly subsonic flows. The velocity-gradient methods are not limited in this way (e.g., ref. 2). On the other hand, a simple velocity-gradient method is limited to a well-guided channel. The purpose of the program described herein is to combine these methods so as to extend the range of cases which can be solved. Locally supersonic (transonic) solutions can be obtained even with low-solidity blading (channel not well guided). This program is called TSONIC (for transonic).

The TSONIC program is based on the TURBLE program (ref. 3), with the addition of subroutines for solving the velocity-gradient equation. Therefore, the input for TSONIC is identical to that for TURBLE, but with an additional input item to give a reduced weight flow factor which is needed to obtain a preliminary subsonic solution.

TSONIC obtains the numerical solution for ideal, transonic, compressible flow for an axial, radial, or mixed flow cascade of turbomachine blades. The cascade may be circular or straight (infinite) and may be fixed or rotating. To accommodate either radial or axial flow with the same coordinate system, the independent variables are meridional streamline distance and angle in radians.

This report includes the FORTRAN IV program TSONIC with a complete description of the input required. Numerical examples have been included to illustrate the use of the program. Only the parts of the program which differ from the TURBLE program are described, although the complete program listing is given.

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

A TSONIC source deck on tape is available from COSMIC (Computer Software Management and Information Center), Computer Center, University of Georgia 30601. The program can be ordered by using the number of this report as identification.

SYMBOLS

A	coefficient in differential eq. (4)
B	coefficient in differential eq. (4)
b	stream-channel thickness normal to meridional streamline, meters
m	meridional streamline distance, meters, see figs. 2 and 3
R	gas constant, J/(kg)(K)
r	radius from axis of rotation to meridional stream-channel mean line, meters

S	streamline distance, meters
s	angular blade spacing or pitch, rad
T	temperature, K
t	time, sec
u	stream function
V	absolute fluid velocity, meters/sec
W	fluid velocity relative to blade, meters/sec
w	mass flow per blade flowing through stream channel, kg/sec
z	axial coordinate, meters
α	angle between meridional streamline and axis of rotation, rad, see figs. 1 and 3
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 figs. 1 and 2
λ	prerotation, $(rV_\theta)_{in}$, meters^2/sec
ρ	density, kg/meters^3
ω	rotational speed, rad/sec

Subscripts:

cr	critical velocity
in	inlet or upstream
j	dummy variable
le	leading edge
m	component in direction of meridional streamline
out	outlet or downstream
r	radial component
te	trailing edge
z	axial component
θ	tangential component

Superscripts:

' absolute stagnation condition

'' relative stagnation condition

MATHEMATICAL ANALYSIS

The calculations are performed in two stages. The first stage is to obtain a solution based on a reduced weight flow by the finite-difference solution of the stream-function equation as described in references 1 and 3. For this first stage of the calculations, weight flow must be reduced sufficiently so that the flow is completely subsonic throughout the passage.

The second stage is to obtain a velocity distribution based on the actual weight flow by means of a velocity-gradient equation. The velocity-gradient solution requires information obtained in the first stage. There may be locally supersonic flow.

The velocity across the width of a curved passage will vary. Hence, at the throat of a curved passage that is choked, there will be both supersonic and subsonic velocities across the passage width. If the weight flow is just slightly less than choking, two solutions are possible, and both solutions will have both subsonic and supersonic velocities. However, the solutions obtained by TSONIC are always the overall "subsonic" solution (i.e., the velocities are always less than those corresponding to choking weight flow).

The simplifying assumptions used are those in references 1 and 3. These assumptions are

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

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

(3) The fluid is nonviscous, and there is no heat transfer (therefore, the flow is isentropic).

(4) The flow is absolutely irrotational.

(5) The blade-to-blade surface is a surface of revolution. (This does not exclude straight infinite cascades.)

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

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

(8) The velocity magnitude and direction are uniform across both the upstream and downstream boundaries.

(9) The only forces are those due to momentum and pressure gradient.

These assumptions are used throughout the program. The following assumption is used only in the first stage of the calculation: The relative velocity is subsonic everywhere. (This is accomplished by using a reduced weight flow in the first stage of the calculation.)

The flow may be axial, radial, or mixed and there may be a variation in the normal stream-channel thickness b in the through-flow direction. Since the stream-channel thickness can be specified as desired, a loss in relative stagnation pressure can be accounted for by reducing b by a percentage equal to the percentage loss in relative stagnation pressure.

The notation for velocity components is shown in figure 1. For generality, the

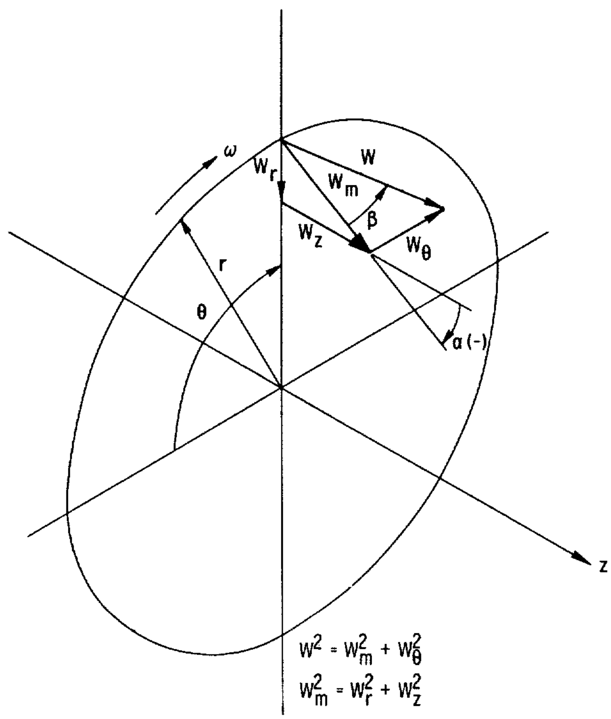


Figure 1. - Cylindrical coordinate system and velocity components.

meridional streamline distance m is used as an independent coordinate (see fig. 2). Thus, m and θ are the two basic independent variables. A stream channel is defined by specifying a meridional streamline radius r and a stream-channel thickness b as functions of m alone (see fig. 3). The variables r and b are constant functions of θ.

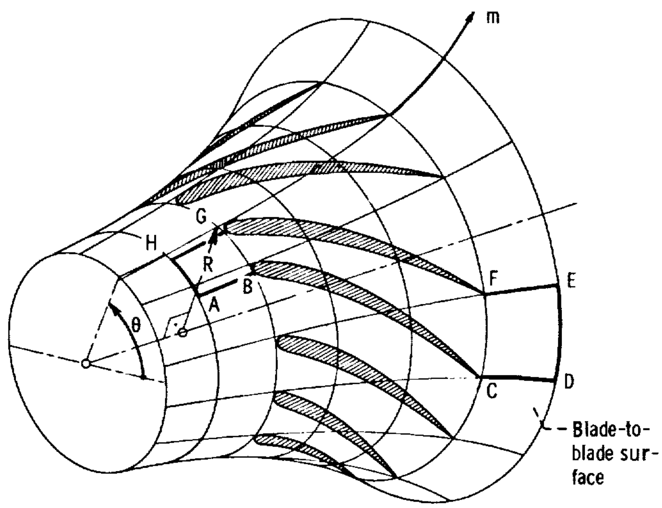


Figure 2. - Blade-to-blade surface of revolution, showing m - θ coordinates.

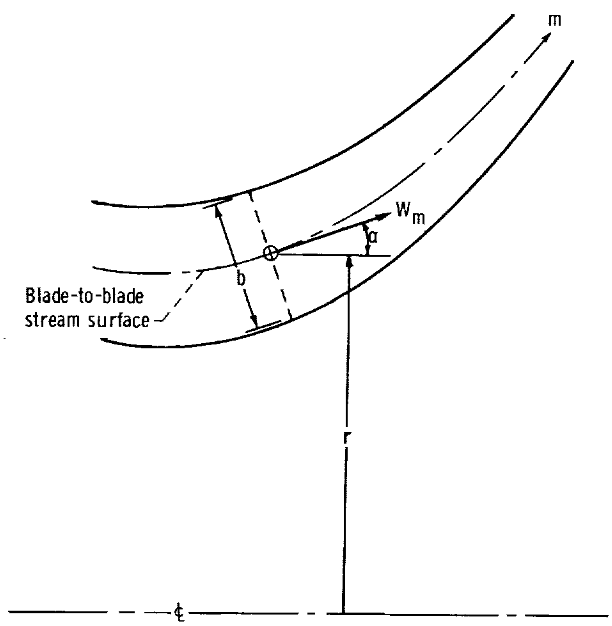


Figure 3. - Flow in a mixed-flow stream channel.

For the mathematical formulation of the problem the stream function is used. The stream function u is normalized so that u is 0 on the upper surface of the lower blade, and 1 on the lower surface of the upper blade. The stream function satisfies the following equation (ref. 1).

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

The derivatives of the stream function satisfy

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

$$\frac{\partial u}{\partial \theta} = \frac{b\rho r}{w} W_m \tag{3}$$

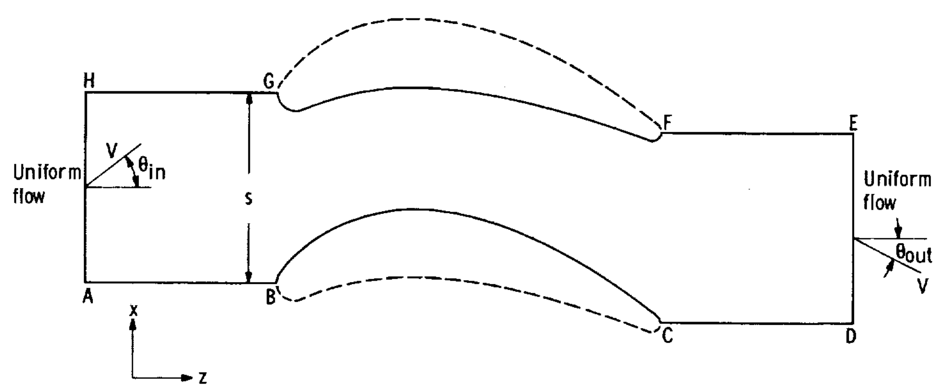


Figure 4. - Finite flow region.

If the flow is entirely subsonic, equation (1) is elliptic. Boundary conditions for the entire boundary ABCDEFGHA of figure 4 will determine a unique solution for u. These boundary conditions (ref. 1) are as follows:

Boundary segment	Boundary condition
AB	u is 1 less than the value of u on GH at the same m coordinate.
BC	u = 0
CD	u is 1 less than the value of u on EF at the same m coordinate.
DE	$\left(\frac{\partial u}{\partial \eta}\right)_{out} = -\frac{\tan \beta_{out}}{sr_{out}}$
EF	u is 1 greater than the value of u on CD at the same m coordinate.
FG	u = 1
GH	u is 1 greater than the value of u on AB at the same m coordinate.
AH	$\left(\frac{\partial u}{\partial \eta}\right)_{in} = \frac{\tan \beta_{in}}{sr_{in}}$

For the case where there is locally supersonic flow, equation (1) is no longer elliptic in the entire region, but is hyperbolic in the region of supersonic flow. (This is discussed in chapter 14 and appendix A of ref. 4.) With a mixed-type problem like this, an analytical solution to equation (1) probably does not exist. This is discussed in reference 5 and means simply that there is probably a shock loss. However, the shock loss may be so small as to be negligible in a numerical solution. In this case we are justified in looking for a numerical solution to equation (1).

At first one may think that equation (1) could be solved by a finite-difference method even when there is locally supersonic flow. There are, however, difficulties with this approach. The difficulty has to do with the fact that there are two velocities, one subsonic and one supersonic, which will give the same value for the weight flow parameter ρW. If a stream-function solution is obtained, we can calculate the stream-function derivatives to obtain values of ρW by using equations (2) and (3). However, if there is locally supersonic flow, there is no easy way of telling which points should use the subsonic velocity and which points should use the supersonic velocity. This is further complicated by the fact that equation (1) is nonlinear and requires iteration to obtain the coefficients involving the density ρ. Therefore, in the initial iteration the predicted values of ρW near the supersonic region will be too large, so that no velocity W can be found to correspond to the predicted value of ρW.

Because of the difficulties with the finite-difference method of solution a different technique is used for the case with locally supersonic flow. The method is based on the following velocity-gradient equation:

$$\frac{\partial W}{\partial \theta} = AW + B \tag{4}$$

where

$$A = r^2 \cos^2 \beta \frac{d^2\theta}{dm^2} + \sin \alpha \tan \beta \left(1 + \cos^2 \beta\right) \tag{5a}$$

is used on blade surface,

$$A = \sin^2 \beta \left[2 \frac{\dfrac{\partial^2 u}{\partial \theta \, \partial m}}{\dfrac{\partial u}{\partial m}} - \frac{\dfrac{\partial u}{\partial \theta}}{\left(\dfrac{\partial u}{\partial m}\right)^2} \frac{\partial^2 u}{\partial m^2} - \frac{\dfrac{\partial^2 u}{\partial \theta^2}}{\dfrac{\partial u}{\partial \theta}} \right] + \sin \alpha \tan \beta \left(1 + \cos^2 \beta\right) \tag{5b}$$

is used at interior points, and

$$B = r \tan \beta \frac{\partial W}{\partial m} + \frac{2\omega r \sin \alpha}{\cos \beta} \tag{6}$$

Equations (4) to (6) are derived in appendix A from the force equation. The quantities in equations (5) and (6) are known if a solution to equation (1) is known. Therefore, it is desired to obtain an approximate solution to equation (1) at a reduced weight flow such that the streamlines at the reduced weight flow correspond closely to those for the actual weight flow. If the flow were incompressible, there would be no change at all in the streamlines if ω is reduced by the same ratio as the weight flow w. This can be seen from equation (1), since the coefficients are constants for incompressible flow and equation (1) does not change if ω/w is kept constant. None of the quantities in equations (5) and (6) would change, except for $\partial W/\partial m$, which is proportional to the weight flow w for incompressible flow. For the compressible case there will be some change in streamline shape, and $\partial W/\partial m$ will not be strictly proportional to the weight flow. However, for many cases, the error introduced by using quantities in equations (5) and (6) from a reduced weightflow solution is not significant. Hence, a solution to equation (1) can be obtained for a reduced weight flow (with correspondingly reduced ω), such that the values of β, $\partial W/\partial m$, and the partial derivatives of u can all be estimated reasonably. More complete detail on obtaining the reduced weight flow solution is given in appendix B. Alternate expressions for the parameter A are given since $d^2\theta/dm^2$ is known on the blade surface, but the partials of u are known in the interior of the region.

Equation (4) can be solved numerically along vertical line in figure 4 (i.e., constant m) by using the approximate values of A and B calculated from the completely subsonic solution. To solve equation (4) directly requires a known value of W at some initial point, for example, the intersection of the vertical line with the lower boundary of the region. However, the condition that determines a unique solution to equation (4) is not a value of W at some point, but rather that continuity must be satisfied. That is, we require that

$$\int_{\theta_1}^{\theta_2} \rho W \cos \beta \, br \, d\theta = w \tag{7}$$

where θ_1 is the value of θ at the lower boundary and θ_2 is the value of θ at the upper boundary. One way to satisfy equation (7) is to try some initial value of W, and then solve equation (4), and calculate the integral in equation (7). If the value of the integral is not equal to w, then other initial values can be tried successively, iterating until the correct solution is obtained. The numerical solution of equation (4) is obtained by a Runge-Kutta method as described in the description of subroutine VELGRA.

When equation (4) has been solved, subject to satisfying equation (7), the velocity distribution is known along vertical line running from the lower to the upper boundary of the region (fig. 4). By doing this for a large number of vertical lines, we obtain the velocity distribution for the entire region, including both blade surfaces.

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 an axial flow turbine stator, and the other is a radial inflow turbine.

Axial Stator

This example is a stator nozzle mean blade section (fig. 5) for a turbine built at Lewis (ref. 6). This blade section has also been analyzed by using the subsonic compressible flow program TURBLE of reference 3. These results are the same as that

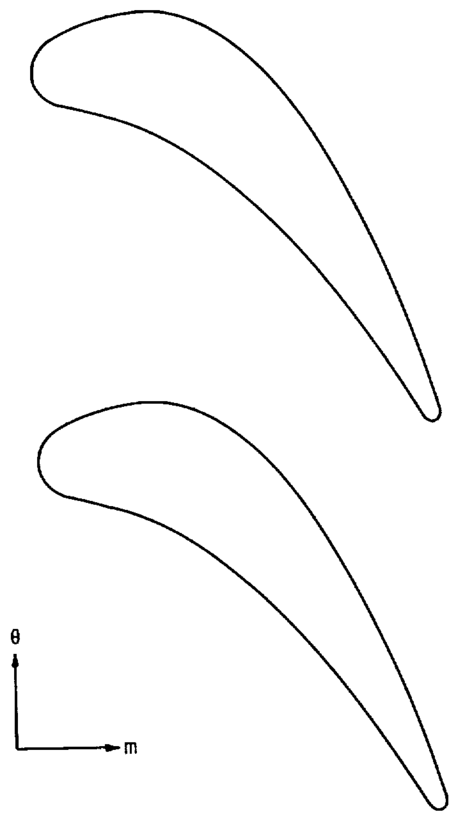


Figure 5. - Axial stator blade for numerical example.

reported in reference 7. The design weight flow could not be analyzed with the TURBLE program, because the velocities were too close to the sonic velocity. However, satisfactory results were obtained with TSONIC by using a value of REDFAC = 0.8. This means that the stream-function solution was obtained first with 80 percent of design weight flow, then the velocity-gradient method was used to obtain the velocities at design weight flow.

The input for this case is given in table I. The design weight flow velocities calculated by the program are given in table II. These velocities are plotted against blade surface length in figure 6. Also shown in figure 6 are experimental data obtained from the investigation described in reference 6. The experimental velocities shown are taken from figure 13(b) of reference 6. Most of the velocities are in very close agreement.

TABLE I. - INPUT FOR AXIAL FLOW TURBINE STATOR CASE

```
 AXIAL FLOW STATOR - MEAN SECTION
  GAM              AR              TIP             RHOIP            WTFL
1.4000000       287.05300       288.15000       1.2250000       0.3619600      -0              -0   OMEGA          -0   ORF
  BETAI            BETAO           CHORDF          STGRF
-0              -67.000000      0.4265000E-01   -0.1116150
  REDFAC           DENTOL
0.8000000       0.1000000E-02
MBI  MBO                 MM   NBBI  NBL  NRSP
15   49    -0    -0      63    20   50    2

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

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

  MR  ARRAY
-1.0000000      1.0000000
  RMSP  ARRAY
0.3302000       0.3302000
  BESP  ARRAY
0.1016000       0.1016000

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

TABLE II. - DESIGN WEIGHT FLOW VELOCITIES FOR AXIAL FLOW TURBINE STATOR CASE

SURFACE VELOCITIES BASED ON MERIDIONAL COMPONENTS - FULL WEIGHT FLOW

	Blade surface 1		Blade surface 2	
M	VELOCITY	W/WCR	VELOCITY	W/WCR
0.1254E-02	131.29	0.4226	91.073	0.2932
0.2509E-02	112.81	0.3632	101.71	0.3274
0.3763E-02	119.33	0.3841	79.076	0.2546
0.5018E-02	128.84	0.4147	75.463	0.2429
0.6272E-02	139.08	0.4477	72.647	0.2339
0.7526E-02	150.03	0.4830	70.493	0.2269
0.8781E-02	161.65	0.5204	69.206	0.2228
0.1004E-01	173.75	0.5593	69.349	0.2232
0.1129E-01	186.06	0.5990	70.102	0.2257
0.1254E-01	197.91	0.6371	71.480	0.2301
0.1380E-01	208.60	0.6715	73.405	0.2363
0.1505E-01	218.05	0.7019	75.772	0.2439
0.1631E-01	226.50	0.7291	78.581	0.2530
0.1756E-01	234.32	0.7543	81.884	0.2636
0.1882E-01	241.79	0.7783	85.602	0.2756
0.2007E-01	245.95	0.7917	89.865	0.2893
0.2132E-01	250.86	0.8076	94.623	0.3046
0.2258E-01	252.71	0.8135	99.933	0.3217
0.2383E-01	256.40	0.8254	105.77	0.3405
0.2509E-01	256.21	0.8248	112.19	0.3612
0.2634E-01	254.64	0.8197	119.18	0.3836
0.2760E-01	251.72	0.8103	126.70	0.4079
0.2885E-01	251.15	0.8085	134.66	0.4335
0.3011E-01	248.53	0.8000	143.39	0.4616
0.3136E-01	247.89	0.7980	152.73	0.4917
0.3261E-01	248.42	0.7997	162.76	0.5239
0.3387E-01	251.03	0.8081	173.16	0.5574
0.3512E-01	260.78	0.8395	185.03	0.5956
0.3638E-01	270.18	0.8697	196.93	0.6339
0.3763E-01	276.10	0.8888	209.72	0.6751
0.3889E-01	273.96	0.8819	218.35	0.7029
0.4014E-01	263.68	0.8488	217.54	0.7003
0.4140E-01	240.48	0.7741	413.83	1.3322

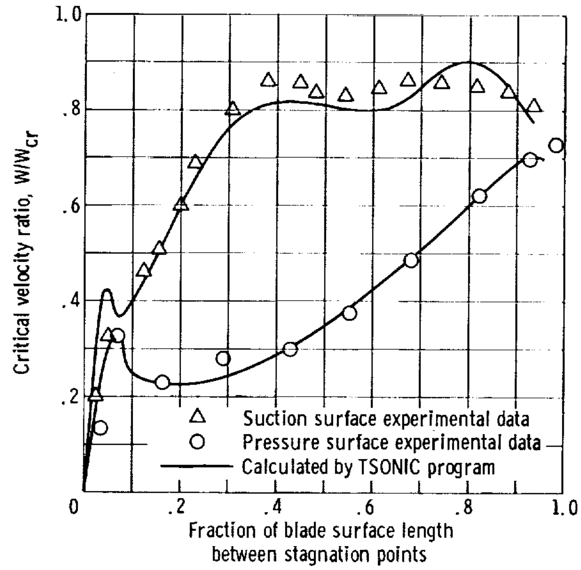


Figure 6. - Blade surface velocities for axial stator mean section compared with experimental data.

Radial Inflow Turbine

An example turbine rotor profile is shown in figure 7. This is a high-specific-speed turbine, with a specific speed of 1.01 (131 (rpm)(ft)$^{3/4}$/(sec)$^{1/2}$). First a meridional plane analysis was made by the quasi-orthogonal method described in reference 11. This analysis determined the meridional streamline spacing. The meridional streamline spacing gives the information for array BESP in the input. A preliminary solution

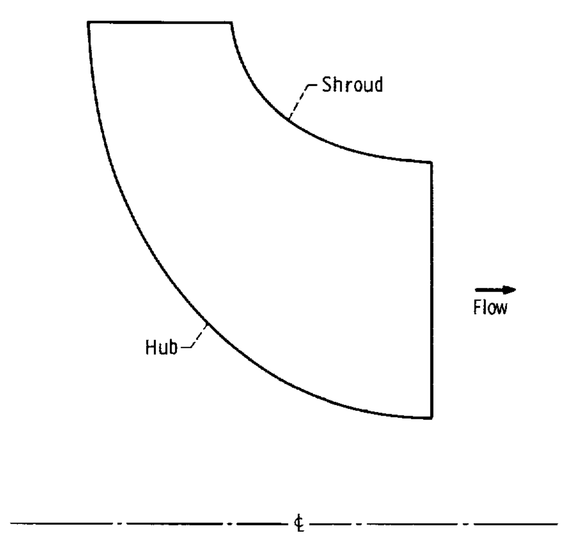


Figure 7. - Hub-shroud profile of radial turbine rotor.

is presented here for a blade-to-blade stream channel adjacent to the shroud. The outlet flow angle β_{te} is estimated based on the blade trailing-edge angle. The solution will help decide whether the estimated value of β_{te} is correct.

The blade-to-blade channel is shown in figure 8. The input data for the program

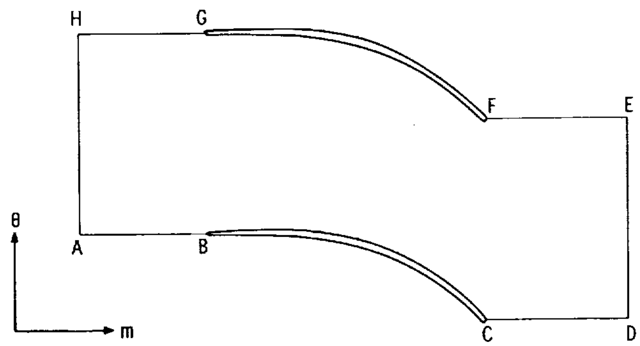


Figure 8. - Blade-to-blade region for radial turbine rotor.

are given in table III. The final velocities, based on the full weight flow, are given in table IV. These velocities are plotted against the meridional streamline distance m in figure 9.

Figure 9 indicates a high blade loading with moderate suction surface diffusion. However, the blade loading probably is higher than would actually be obtained near the trailing edge. This indicates that, most likely, less turning would be obtained than that

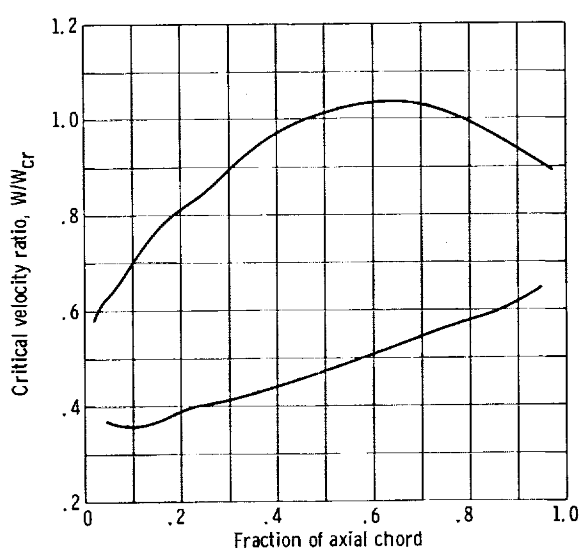


Figure 9. - Blade surface velocities for radial turbine rotor.

TABLE III. - INPUT FOR RADIAL INFLOW TURBINE CASE

```
 RADIAL GAS TURBINE - SHROUD
  GAM              AR              TIP             RHOIP             WTFL                           OMEGA           ORF
 1.3240000       1718.7000       1960.0000        0.2314000E-02    0.1898000E-02   -0                2513.0000       1.8400000
  BETAI           BETAO           CHORDF           STGRF
-4.2000000      -36.000000       0.2931000       -0.1855000
  REDFAC           DENTOL
 0.7000000       0.1000000E-02
MBI  MBO               MM   NBBI  NBL  NRSP
10   30   -0   -0      40   14    15   11

 BLADE SURFACE 1  --  UPPER SURFACE
   RI1              RO1             BETI1            BETO1            SPLNO1
 0.2500000E-02    0.2500000E-02  -0                -39.410000        9.0000000
   MSP1  ARRAY
-0
-0                0.3530000E-01   0.6450000E-01    0.1092000        0.1452000        0.2000000        0.2429000        0.2723000
   THSP1  ARRAY
-0
-0.               0.4900000E-02   0.5100000E-02    0.1900000E-02   -0.8900000E-02   -0.4530000E-01   -0.9540000E-01   -0.1420000

 BLADE SURFACE 2  --  LOWER SURFACE
   RI2              RO2             BETI2            BETO2            SPLNO2
 0.2500000E-02    0.2500000E-02  -0                -38.800000        9.0000000
   MSP2  ARRAY
-0
-0                0.3530000E-01   0.6450000E-01    0.1092000        0.1452000        0.2000000        0.2429000        0.2723000
   THSP2  ARRAY
-0
-0              -0.4600000E-02  -0.6100000E-02   -0.1150000E-01   -0.2310000E-01   -0.6030000E-01   -0.1105000       -0.1580000

   MR  ARRAY
-0.1600000      -0                0.3530000E-01    0.6450000E-01    0.1092000        0.1452000        0.2000000        0.2429000
 0.2723000       0.2931000        0.4531000
   RMSP  ARRAY
 0.7058000       0.5458000        0.5110000        0.4847000        0.4507000        0.4296000        0.4072000        0.3971000
 0.3937000       0.3930000        0.3930000
   BESP  ARRAY
 0.6470000E-02   0.6470000E-02    0.5560000E-02    0.5620000E-02    0.5870000E-02    0.6110000E-02    0.6570000E-02    0.7080000E-02
 0.7560000E-02   0.7930000E-02    0.7930000E-02

 BLDAT  AANDK  ERSOR  STRFN  SLCRD  INTVL  SURVL
   1      0      0      0      0      1      1
```

TABLE IV. - DESIGN WEIGHT FLOW VELOCITIES FOR RADIAL INFLOW TURBINE CASE

SURFACE VELOCITIES BASED ON MERIDIONAL COMPONENTS - FULL WEIGHT FLOW

	BLADE SURFACE 1		BLADE SURFACE 2	
M	VELOCITY	W/WCR	VELOCITY	W/WCR
0.1465E-01	1238.6	0.6544	676.19	0.3573
0.2931E-01	1315.5	0.6963	642.68	0.3402
0.4396E-01	1485.0	0.7874	699.76	0.3710
0.5862E-01	1546.7	0.8215	725.67	0.3854
0.7327E-01	1615.2	0.8591	744.63	0.3961
0.8793E-01	1698.6	0.9047	764.49	0.4072
0.1026	1782.6	0.9506	790.75	0.4217
0.1172	1846.5	0.9858	819.59	0.4375
0.1319	1882.2	1.0058	851.11	0.4548
0.1465	1906.0	1.0194	882.48	0.4720
0.1612	1927.7	1.0318	918.31	0.4915
0.1759	1942.2	1.0402	954.78	0.5114
0.1905	1945.1	1.0423	990.89	0.5310
0.2052	1929.8	1.0346	1028.8	0.5516
0.2198	1894.4	1.0161	1061.1	0.5691
0.2345	1843.2	0.9889	1100.5	0.5905
0.2491	1786.3	0.9586	1115.9	0.5988
0.2638	1724.4	0.9256	1181.9	0.6344
0.2784	1652.2	0.8869	1251.3	0.6717

given as input to the program. The curves for the suction and pressure surface velocities would be made to come together at the trailing edge by reducing β_{te} in additional computer runs.

DESCRIPTION OF INPUT AND OUTPUT

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

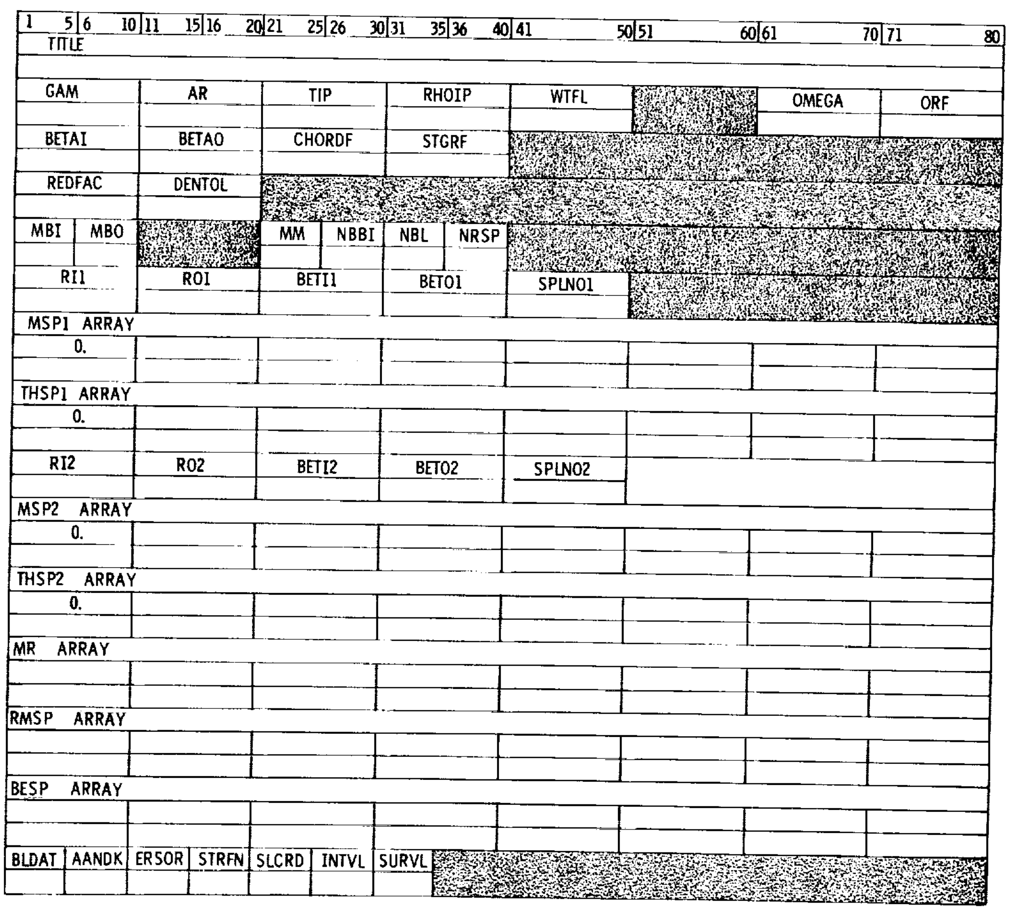


Figure 10. - Input form.

Input

Figure 10 shows the input variables as they are punched on the data cards. There are two types of variables, geometric and nongeometric. The geometric input variables are shown in figures 11 and 12. All input variables are described in this section.

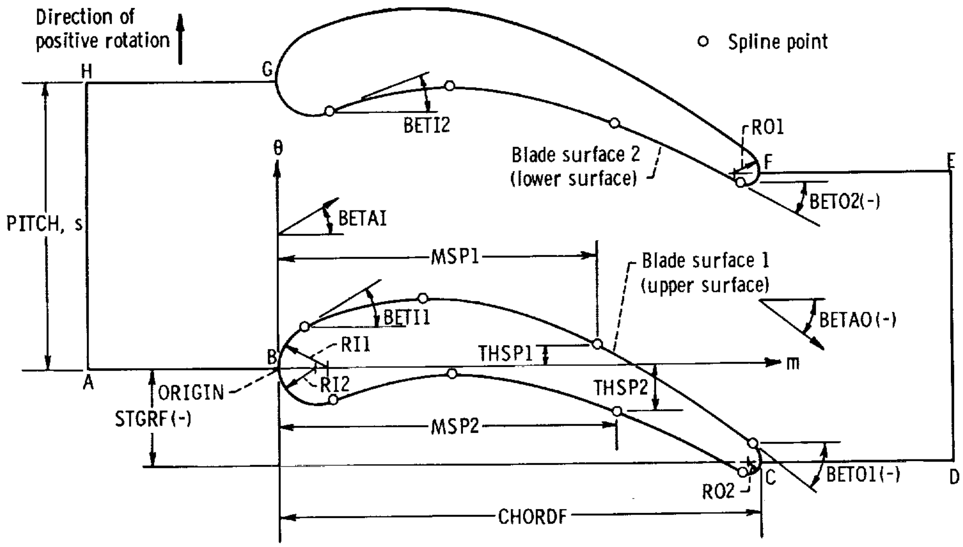


Figure 11. - Geometric input variables on blade-to-blade stream surface. Angles BETAI, BETAO, BETI1, BETI2, BETO1, and BETO2 must be given as true angle β in degrees, not the angle as measured in m-θ plane. Either use tan β = r dθ/dm to obtain β, or measure the true angle.

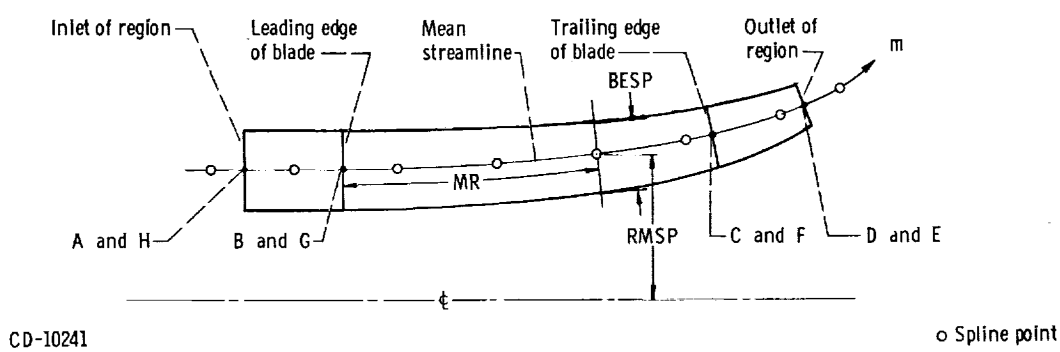


Figure 12. - Geometric input variables describing stream channel in meridional plane.

Further explanation of key variables is given in the section Instructions for Preparing Input.

The input variables are as follows:

GAM specific-heat ratio

AR	gas constant, J/(kg)(K)
TIP	inlet stagnation temperature, K
RHOIP	inlet stagnation density, kg/meter^3
WTFL	mass flow per blade for stream channel, kg/sec
OMEGA	rotational speed, ω, rad/sec (Note that ω is negative if rotation is in the opposite direction of that shown in fig. 1.)
ORF	value of overrelaxation factor to be used in the solution of the inner iteration simultaneous equations (If ORF = 0, the program calculates an estimated value for the overrelaxation factor. See p. 25 for discussion.)
BETAI	inlet flow angle β_{le} along BG with respect to m-direction, deg, see fig. 11
BETAO	outlet flow angle β_{te} along CF with respect to m-direction, deg, see fig. 11
CHORDF	overall length of blade in m-direction, meters, see fig. 11
STGRF	angular θ-coordinate for center of trailing-edge circle of blade with respect to the center of leading-edge circle of blade, radians, see fig. 11
REDFAC	factor by which weight flow (WTFL) must be reduced in order to assure subsonic flow throughout passage (REDFAC is usually between 0.5 and 0.9.)
DENTOL	tolerance on density change per iteration for reduced weight flow (DENTOL may be left blank, and the value 0.001 will be used. If trouble is experienced in obtaining convergence (i.e., the maximum relative change in density (item 14 of the output) does not get small enough), then a larger value of DENTOL may be used, or a smaller value of REDFAC may be used. The value of 0.001 for DENTOL is a tight tolerance, 0.01 would be a medium tolerance, and 0.1 would be a loose tolerance.)
MBI	number of vertical mesh lines from AH to BG inclusive, see fig. 13
MBO	number of vertical mesh lines from AH to CF inclusive, see fig. 13
MM	total number of vertical mesh lines in m-direction from AH to DE, maximum of 100, see fig. 13
NBBI	number of mesh spaces in θ-direction between AB and GH, maximum of 50, see fig. 13
NBL	number of blades
NRSP	number of spline points for stream-channel radius (RMSP) and thickness (BESP) coordinates, maximum of 50, see fig. 12

RI1,RI2	leading-edge radii of the two blade surfaces, meters, see fig. 11
RO1,RO2	trailing-edge radii of the two blade surfaces, meters, see fig. 11
BETI1,BETI2	angles (with respect to m-direction) at tangent points of leading-edge radii with the two blade surfaces, deg, see fig. 11 (These must be true angles in degrees. If angles are measured in the m-θ plane, (i.e., $d\theta/dm$), BETI1 and BETI2 can be obtained from the relation $\tan \beta = r(d\theta/dm)$.)
BETO1,BETO2	angles (with respect to m-direction) at tangent points of trailing-edge radii with the two blade surfaces, deg, see fig. 11 (These must also be true angles in degrees, like BETI1 and BETI2.)
SPLNO1,SPLNO2	number of blade spline points given for each surface as input, maximum of 50 (These include the first and last points (dummies) that are tangent to the leading- and trailing-edge radii (fig. 11).)
MSP1,MSP2	arrays of m-coordinates of spline points on the two blade surfaces, measured from the blade leading edge, meters, see fig. 11 (The first and last points in each of these arrays can be blank or have a dummy value, since these points are calculated by the program. If blanks are used, and the last point is on a new card, a blank card must be used.)
THSP1,THSP2	arrays of θ-coordinates of spline points corresponding to MSP1 and MSP2, radians, see fig. 11 (Dummy values are also used here in positions corresponding to those in MSP1 and MSP2.)
MR	array of m-coordinates of spline points for stream-channel radii and stream-channel thickness, meters, see fig. 12 (MR is measured from the leading edge of the blade. These coordinates should cover the entire distance from AH to DE, and may extend beyond these bounds. The total number of points is NRSP.)
RMSP	array of r-coordinates of spline points for the stream-channel radii, corresponding to the MR array, meters, see fig. 12
BESP	array of stream-channel normal thicknesses corresponding to the MR and RMSP arrays, meters, see fig. 12

The remaining variables, starting with BLDAT, are used to indicate what output is desired. A value of 0 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 iterations; 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:

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

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

ERSOR maximum change in stream function at any point for each iteration of SOR equation (eq. (A8), ref. 1)

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

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

INTVL velocity and flow angle at each interior mesh point for both reduced and actual weight flow

SURVL m-coordinate, surface velocity, flow angle, distance along surface, and W/W_{cr} based on meridional velocity components where each vertical mesh line meets each blade surface; m-coordinate, surface velocity, flow angle, distance along surface, and W/W_{cr} based on tangential velocity components where each horizontal mesh line meets each blade surface; plot of blade surface velocities against meridional streamline distance, meters

Instructions for Preparing Input

It is very unusual to have no errors of input the first time TSONIC is run. Therefore, it is recommended that the first attempt should allow only 1 minute of execution time and that BLDAT should be equal to 1. The resulting output should be checked carefully. Of particular interest are the second derivatives at input spline points. Any errors in blade geometry input will usually result in wild values for some of these second derivatives. All other preliminary output should be checked to see that it is reasonable.

Units of measurement. - The International System of Units (ref. 8) is used throughout this report. However, the program does not use any constants which depend on the system of units being used. Therefore, any consistent set of units may be used in preparing input for the program. For example, if force, length, temperature, and time are chosen independently, mass units are obtained from Force = Mass × Acceleration. The gas constant R must then have the units of force times length divided by mass times temperature (energy per unit mass per degree temperature). Density is mass per unit volume, and weight flow is mass per unit time. Output than gives velocity in the chosen units of length per unit time. Since any consistent set of units can be employed, the output is not labeled with any units.

Blade and stream-channel geometry. - The upper and lower surfaces of the blade are each defined by specifying three things: leading- and trailing-edge radii, angles at which these radii are tangent to the blade surfaces, and m- and θ-coordinates of several points along each surface. These angles and coordinates are used to define a cubic spline curve fit (ref. 9) to the surface. The standard sign convention is used for angles, as indicated in figure 11.

A cubic spline curve is a piecewise cubic polynomial which expresses mathematically the shape taken by an idealized spline passing through the given points. Reference 9 describes a method for determining the equation of the spline curve. Using this method, few points are required to specify most blade shapes accurately, usually no more than five or six in addition to the two end points. As a guide, enough points should be specified so that a physical spline passing through these points would accurately follow the blade shape. This means that the spline points should be closer where there is large curvature and farther apart where there is small curvature. As a check, the program should be run for 1 minute of execution time with BLDAT = 1 for any new geometry. Check the second derivatives at the spline points to see that they are reasonable. Also check blade curvatures at vertical mesh lines.

The coordinates for either surface of the blade are given with respect to the leading edge, with the leading edge of the blade being defined as the furthest point upstream (see fig. 11).

The mean stream surface of revolution (as seen in the meridional plane, fig. 12) and the stream-channel thickness are also fitted with cubic spline curves. The m-coordinates for the mean stream surface are independent of the m-coordinates for the blade surface.

Loss of relative stagnation pressure. - If desired, a simplified correction for losses can be made by assuming a loss in relative stagnation pressure. This type of loss can be accounted for by reducing each value in the BESP array by a percentage equal to the assumed percentage loss in relative stagnation pressure at that point.

Inlet and outlet flow angles. - The values of β_{le} and β_{te} are given as average values on BG and CF, respectively. If the flow is axial, these flow angles are constant upstream or downstream of the blades. If flow is radial or mixed, and these angles are not known on BG and CF, β_{le} and β_{te} must be calculated by equation (B14) of reference 1 or (B15) of reference 7.

Defining the mesh. - A finite-difference mesh is used for the solution of the basic differential equation. A typical mesh pattern is shown in figure 13. The mesh spacing and the extent of the upstream and downstream regions are determined by the values of MBI, MBO, and MM of the input. The mesh spacing must be chosen so that there are not more than 2500 unknown mesh points.

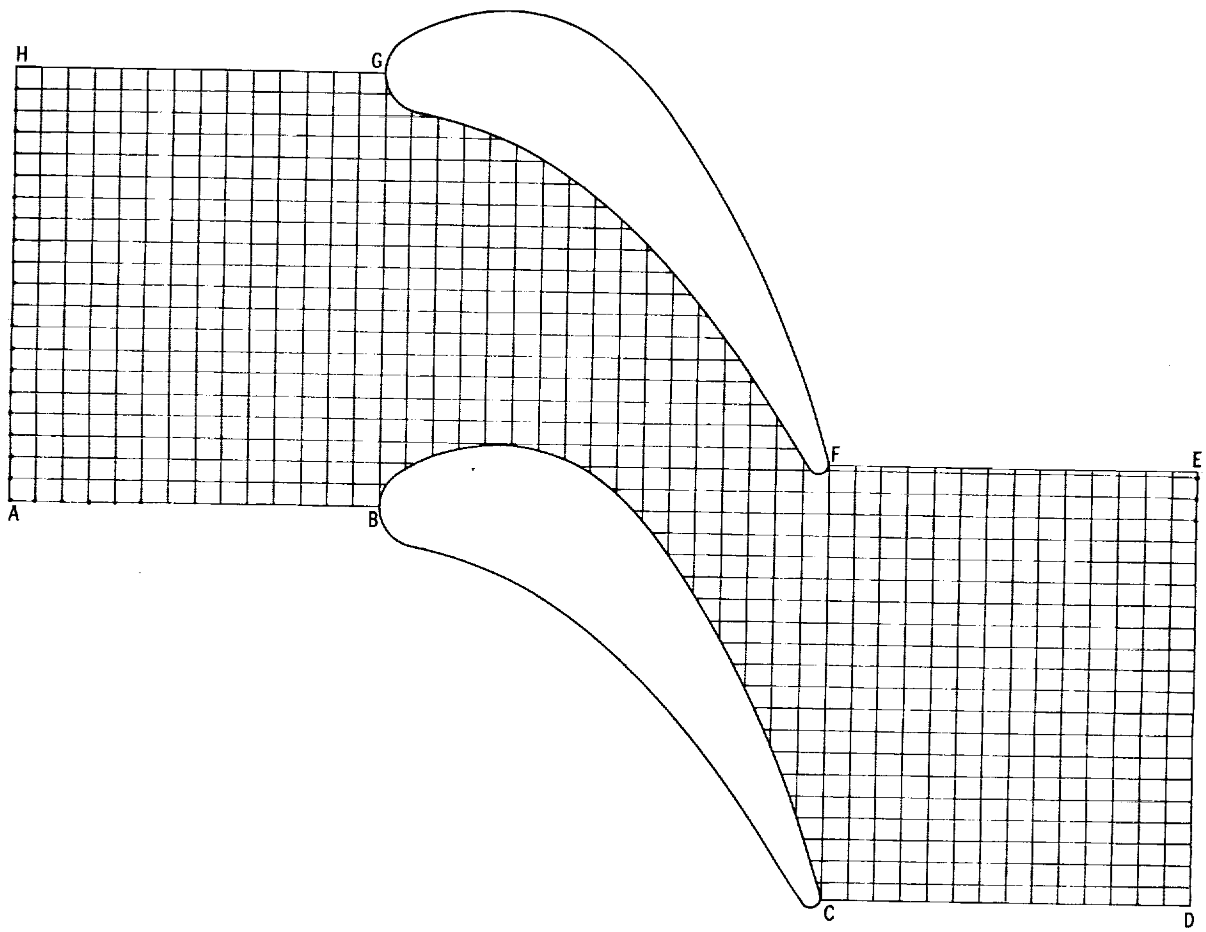


Figure 13. - Typical mesh in blade-to-blade solution region.

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

Overrelaxation factor. - ORF is the overrelaxation factor used in each inner iteration in the solution of the simultaneous finite-difference equations (see ref. 2, p. 102). ORF may be set to 0, or some value between 1 and 2. ORF should be 0 for the initial run of a given blade geometry and mesh spacing (MBI, NBBI, etc.). In this case the program uses extra time and calculates an optimum value for ORF. It does this by means of an iterative process, and on each iteration the current estimate of the optimum value for ORF is printed. The final estimate is the one used by the program for ORF. If the user does not change the mesh indexes MBI, MBO, MM, and NBBI between runs, even though blade geometry or other input does change, he may use this final estimate of ORF in the input, saving the time used in its computation. In all cases, if ORF is not 0, it should have a value greater than 1 and less than 2.

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

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

Incompressible flow. - While the program is written for compressible flow, it can be easily used for incompressible flow. To do so specify GAM = 1.5, AR = 1000, TIP = 10^6, and REDFAC = 1 as input. This results in a single outer iteration of the program to obtain the stream-function solution. And, of course, the velocity-gradient solution will yield nothing new.

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

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

This radius should then be used to compute the θ-coordinates required as input (THSP1, THSP2, and STGRF).

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

Output

Sample output is given in table V for the axial flow stator example of reference 1. The blade shape is shown in figure 13. Since the complete output would be lengthy, only the first few lines of each section of output are reproduced here. Most of the output is optional, and is controlled by the final input card, as already described. In some instances output lables are simply internal variable names.

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

(1) The first output is a listing of the input data. All items are labeled as on the input from (fig. 10).

(2) This is the output corresponding to BLDAT (see the list of input variables and the descriptions of internal variables for the subroutines of the program).

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

(4) These are calculated program constants, including the pitch from blade to blade, the mesh spacing in all solution regions, the minimum and maximum values of IT in the solution region (ITMIN and ITMAX), and the value of the prerotation λ (eq. (B8), ref. 1) for both full and reduced weight flow.

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

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

(7) This is the output corresponding to AANDK.

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

TABLE V. - SAMPLE OUTPUT FOR AXIAL FLOW TURBINE STATOR CASE

1

```
  AXIAL FLOW STATOR - MEAN SECTION
   GAM              AR               TIP              RHOIP            WTFL                                         OMEGA           ORF
 1.4000000        287.05300        288.15000        1.2250000        0.3619600        -0               -0               -0
   BETAI            BETAO            CHORDF           STGRF
-0               -67.000000        0.4265000E-01   -0.1116150
  REDFAC            DENTOL
 0.8000000         0.1000000E-02
MBI  MBO            MM   NEBI  NBL  NRSP
15   49   -0   -0   63    20   50    2

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

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

   MR  ARRAY
-1.0000000        1.0000000
   RMSP  ARRAY
 0.3302000        0.3302000
   BESP  ARRAY
 0.1016000        0.1016000

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

TABLE V. - Continued. SAMPLE OUTPUT FOR AXIAL FLOW TURBINE STATOR CASE

BLADE DATA AT INPUT SPLINE POINTS

2

BLADE SURFACE 1

M	THETA	DERIVATIVE	2ND DERIV.
0.20037E-02	0.10159E-01	1.53066	-129.936
0.85750E-02	0.17690E-01	0.60358	-182.661
0.17150E-01	0.15360E-01	-1.23217	-245.503
0.25725E-01	-0.53100E-02	-3.72155	-335.112
0.34300E-01	-0.46540E-01	-5.54459	-90.0846
0.38588E-01	-0.74000E-01	-7.92944	-1022.25
0.42606E-01	-0.11080	-9.54694	217.603

BLADE SURFACE 2

M	THETA	DERIVATIVE	2ND DERIV.
0.28754E-02	0.11448	-0.76632	57.4409
0.85750E-02	0.11004	-0.96494	-127.138
0.17150E-01	0.97124E-01	-2.04512	-124.799
0.25725E-01	0.74964E-01	-3.12744	-127.637
0.34300E-01	0.43164E-01	-4.32324	-151.265
0.41023E-01	0.12547E-01	-4.50684	96.6480

3

	FREESTREAM VELOCITY	MAXIMUM VALUE FOR RHO*W	CRITICAL VELOCITY
LEADING EDGE B-G	71.6668	241.239	310.645
TRAILING EDGE C-F	225.925	241.239	310.645

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

(BASED ON REDUCED WEIGHT FLOW)

4

CALCULATED PROGRAM CONSTANTS

PITCH	HT	HM1
0.1256637	0.6283185E-02	0.1254412E-02

ITMIN	ITMAX
-17	19

LAMBDA	LAMBDA AT REDUCED WEIGHT FLOW
-0	-0

5

NUMBER OF INTERIOR MESH POINTS = 1069

6

SURFACE BOUNDARY VALUES

SURFACE	BV
1	0.
2	1.00000

2

BLADE DATA AT INTERSECTIONS OF VERTICAL MESH LINES WITH BLADES

	BLADE SURFACE 1			BLADE SURFACE 2		
M	TV	DTDMV	CURV	TV	DTDMV	CURV
0	0	0.10000E 11	0	0.12566	-0.10000E 11	0
0.12544E-02	0.85578E-02	2.73889	-262.467	0.11711	-2.73889	262.467
0.25088E-02	0.10966E-01	1.56401	-31.0335	0.11482	-1.10043	262.467
0.37632E-02	0.12820E-01	1.38962	-35.7132	0.11392	-0.72809	8.70714
0.50176E-02	0.14447E-01	1.20260	-40.6549	0.11291	-0.71758	-3.63098

2

STREAM SHEET COORDINATES AND THICKNESS TABLE

IM	M	R	SAL	B	DB/DM
1	-0.17562E-01	0.33020	-0	0.10160	-0
2	-0.16307E-01	0.33020	-0	0.10160	-0
3	-0.15053E-01	0.33020	-0	0.10160	-0
4	-0.13799E-01	0.33020	-0	0.10160	-0
5	-0.12544E-01	0.33020	-0	0.10160	-0

2

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

2

M COORDINATES OF INTERSECTIONS OF HORIZONTAL MESH LINES WITH BLADE

MH ARRAY - BLADE SURFACE 1

MH	RMH	BEH	BETAH	DTDMH
0	0.3302	0.1016	90.000	0.1000E 11
0.6144E-03	0.3302	0.1016	57.007	4.6646
0.3582E-02	0.3302	0.1016	25.052	1.4155
0.1906E-01	0.3302	0.1016	-29.597	-1.7202
0.2201E-01	0.3302	0.1016	-40.095	-2.5498

2

THETA COORDINATES OF HORIZONTAL MESH LINES

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

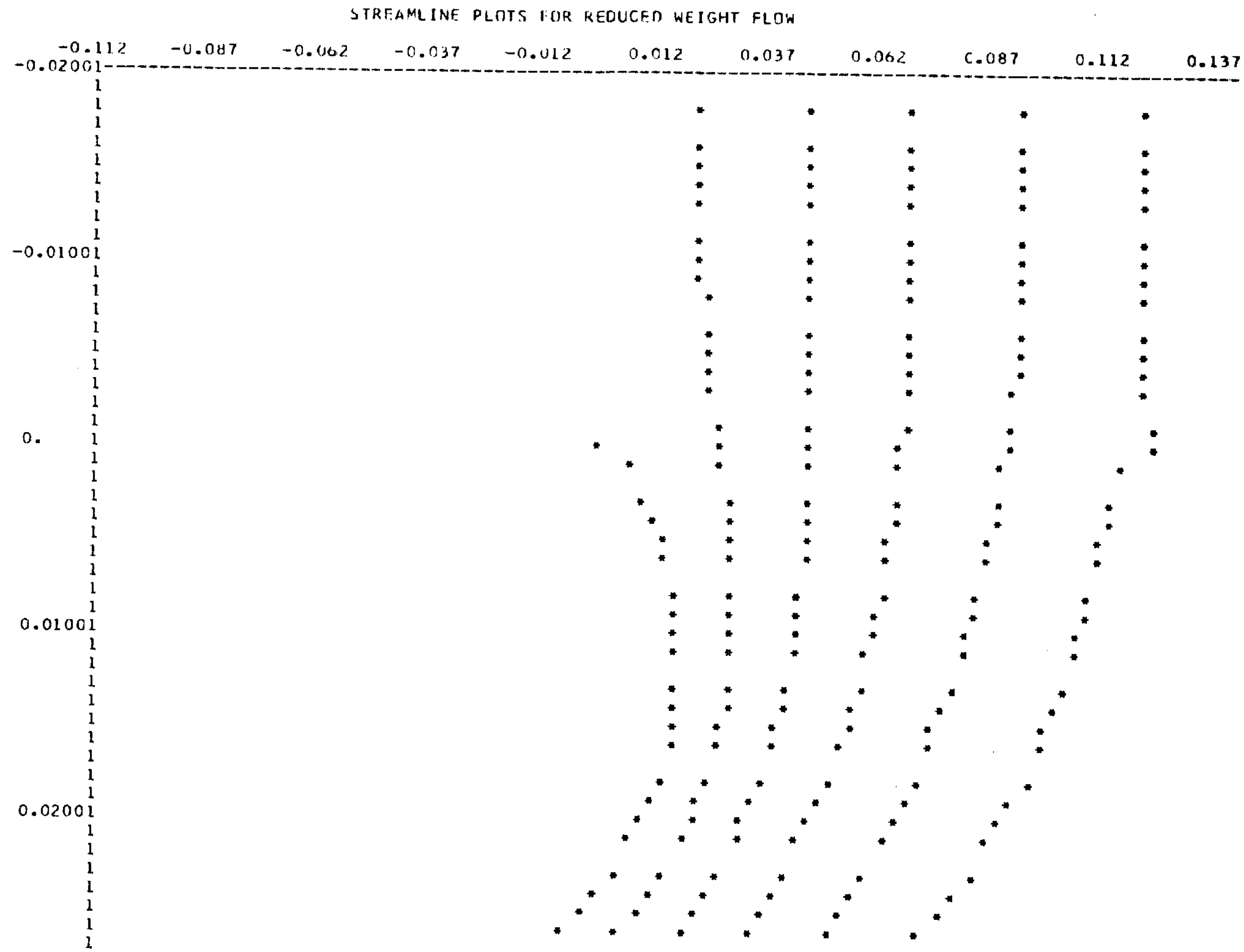
TABLE V. - Continued. SAMPLE OUTPUT FOR AXIAL FLOW TURBINE STATOR CASE

7

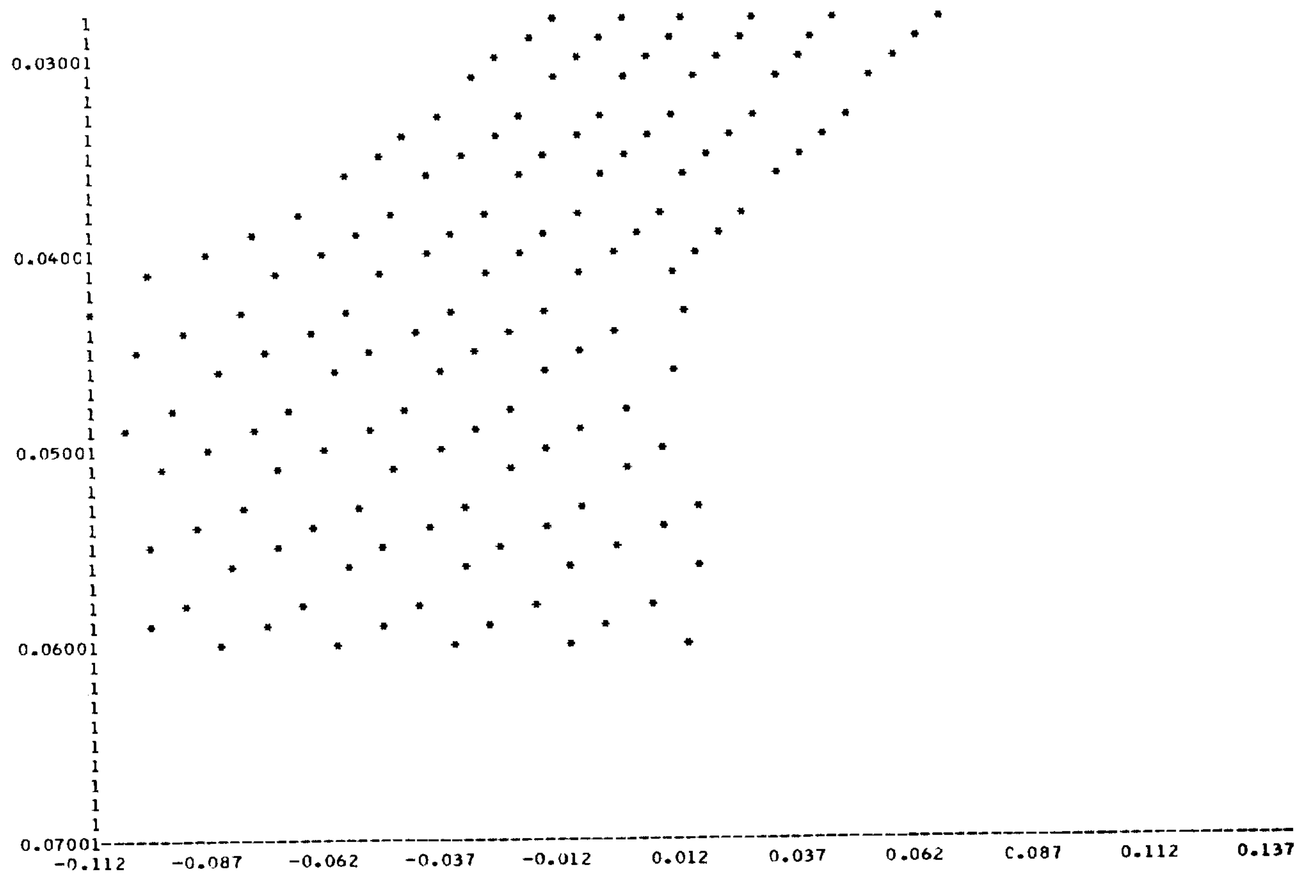
IT	IP	IP1	IP2	IP3	IP4	A(1)	A(2)	A(3)	A(4)	K
IM =	1	IT1 =	0							
0	1	20	2	-0	21	0.	0.	0.	1.00000	-0.
1	2	1	3	1	22	0.	0.	0.	1.00000	-0.
2	3	2	4	2	23	0.	0.	0.	1.00000	-0.
3	4	3	5	3	24	0.	0.	0.	1.00000	-0.
4	5	4	6	4	25	0.	0.	0.	1.00000	-0.
5	6	5	7	5	26	0.	0.	0.	1.00000	-0.
6	7	6	8	6	27	0.	0.	0.	1.00000	-0.
7	8	7	9	7	28	0.	0.	0.	1.00000	-0.
8	9	8	10	8	29	0.	0.	0.	1.00000	-0.
9	10	9	11	9	30	0.	0.	0.	1.00000	-0.
10	11	10	12	10	31	0.	0.	0.	1.00000	-0.
11	12	11	13	11	32	0.	0.	0.	1.00000	-0.
12	13	12	14	12	33	0.	0.	0.	1.00000	-0.
13	14	13	15	13	34	0.	0.	0.	1.00000	-0.
14	15	14	16	14	35	0.	0.	0.	1.00000	-0.
15	16	15	17	15	36	0.	0.	0.	1.00000	-0.
16	17	16	18	16	37	0.	0.	0.	1.00000	-0.
17	18	17	19	17	38	0.	0.	0.	1.00000	-0.
18	19	18	20	18	39	0.	0.	0.	1.00000	-0.
19	20	19	1	19	40	0.	0.	0.	1.00000	-0.
IM =	2	IT1 =	0							
0	21	40	22	1	41	0.13385	0.13385	0.36615	0.36615	-0.13385
1	22	21	23	2	42	0.13385	0.13385	0.36615	0.36615	-0.
2	23	22	24	3	43	0.13385	0.13385	0.36615	0.36615	-0.
3	24	23	25	4	44	0.13385	0.13385	0.36615	0.36615	-0.
4	25	24	26	5	45	0.13385	0.13385	0.36615	0.36615	-0.
5	26	25	27	6	46	0.13385	0.13385	0.36615	0.36615	-0.
6	27	26	28	7	47	0.13385	0.13385	0.36615	0.36615	-0.
7	28	27	29	8	48	0.13385	0.13385	0.36615	0.36615	-0.
8	29	28	30	9	49	0.13385	0.13385	0.36615	0.36615	-0.
9	30	29	31	10	50	0.13385	0.13385	0.36615	0.36615	-0.
10	31	30	32	11	51	0.13385	0.13385	0.36615	0.36615	-0.
11	32	31	33	12	52	0.13385	0.13385	0.36615	0.36615	-0.
12	33	32	34	13	53	0.13385	0.13385	0.36615	0.36615	-0.

```
 8  ESTIMATED OPTIMUM ORF = 2.000000
    ESTIMATED OPTIMUM ORF = 2.000000
    ESTIMATED OPTIMUM ORF = 2.000000
    ESTIMATED OPTIMUM ORF = 1.999827
    ESTIMATED OPTIMUM ORF = 1.999701

 9  ERROR = 1.85730337
    ERROR = 1.59505035
    ERROR = 1.35216291
    ERROR = 1.20889181
    ERROR = 1.12262167

10           STREAM FUNCTION VALUES FOR REDUCED WEIGHT FLOW

    IM =   1       ITI =     0
       0.02371997  0.07150990  0.11993275  0.16916088  0.21925756  0.27017924  0.32179615  0.37391961  0.42633016  0.47880118
       0.53111587  0.58308036  0.63453156  0.68534455  0.73543853  0.78478383  0.83340984  0.88141387  0.92896585  0.97630496

    IM =   2       ITI =     0
       0.02372004  0.07150999  0.11993286  0.16916101  0.21925766  0.27017941  0.32179631  0.37391974  0.42633040  0.47880144
       0.53111625  0.58308076  0.63453192  0.68534483  0.73543873  0.78478402  0.83341005  0.88141400  0.92896591  0.97630507

    IM =   3       ITI =     0
       0.02358316  0.07127872  0.11963863  0.16884360  0.21895614  0.26992545  0.32161131  0.37381501  0.42630862  0.47885899
       0.53124482  0.58326879  0.63476554  0.68560791  0.73571258  0.78504706  0.83363762  0.88157937  0.92904383  0.97627742

11  TIME =  2.6808 MIN.

12                      STREAMLINE COORDINATES FOR REDUCED WEIGHT FLOW

    M COORDINATE     STREAM FN.      THETA           STREAM FN.      THETA           STREAM FN.      THETA

    -0.1756176E-01   0.2000000       0.2273020E-01   0.4000000       0.4711152E-01   0.6000000       0.7117351E-01
                     0.8000000       0.9620439E-01   1.0000000       0.1225224       0.2000000       0.2273020E-01
    -0.1630735E-01   0.2000000       0.2273019E-01   0.4000000       0.4711150E-01   0.6000000       0.7117346E-01
                     0.8000000       0.9620436E-01   1.0000000       0.1225224       0.2000000       0.2273019E-01
```

TABLE V. - Continued. SAMPLE OUTPUT FOR AXIAL FLOW TURBINE STATOR CASE



12



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

TABLE V. - Continued. SAMPLE OUTPUT FOR AXIAL FLOW TURBINE STATOR CASE

13

VELOCITIES AT INTERIOR MESH POINTS
FOR REDUCED WEIGHT FLOW

IM= 1

VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)
53.935	-0.14	54.625	-0.23	55.460	-0.28	56.467	-0.30	57.475	-0.28
58.380	-0.23	59.094	-0.17	59.563	-0.10	59.767	-0.02	59.707	0.05
59.404	0.12	58.891	0.17	58.208	0.21	57.403	0.25	56.534	0.26
55.664	0.25	54.866	0.22	54.228	0.16	53.821	0.08	53.762	-0.03

IM= 2

VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)
53.935	0.14	54.625	0.23	55.460	0.28	56.467	0.30	57.475	0.28
58.380	0.23	59.094	0.17	59.563	0.09	59.767	0.02	59.707	-0.05
59.405	-0.12	58.891	-0.17	58.208	-0.22	57.403	-0.25	56.534	-0.26
55.664	-0.25	54.866	-0.22	54.228	-0.16	53.821	-0.08	53.762	0.03

14 ITERATION NO. 1 MAXIMUM RELATIVE CHANGE IN DENSITY = 0.2135

15

SURFACE VELOCITIES BASED ON MERIDIONAL COMPONENTS - REDUCED WEIGHT FLOW

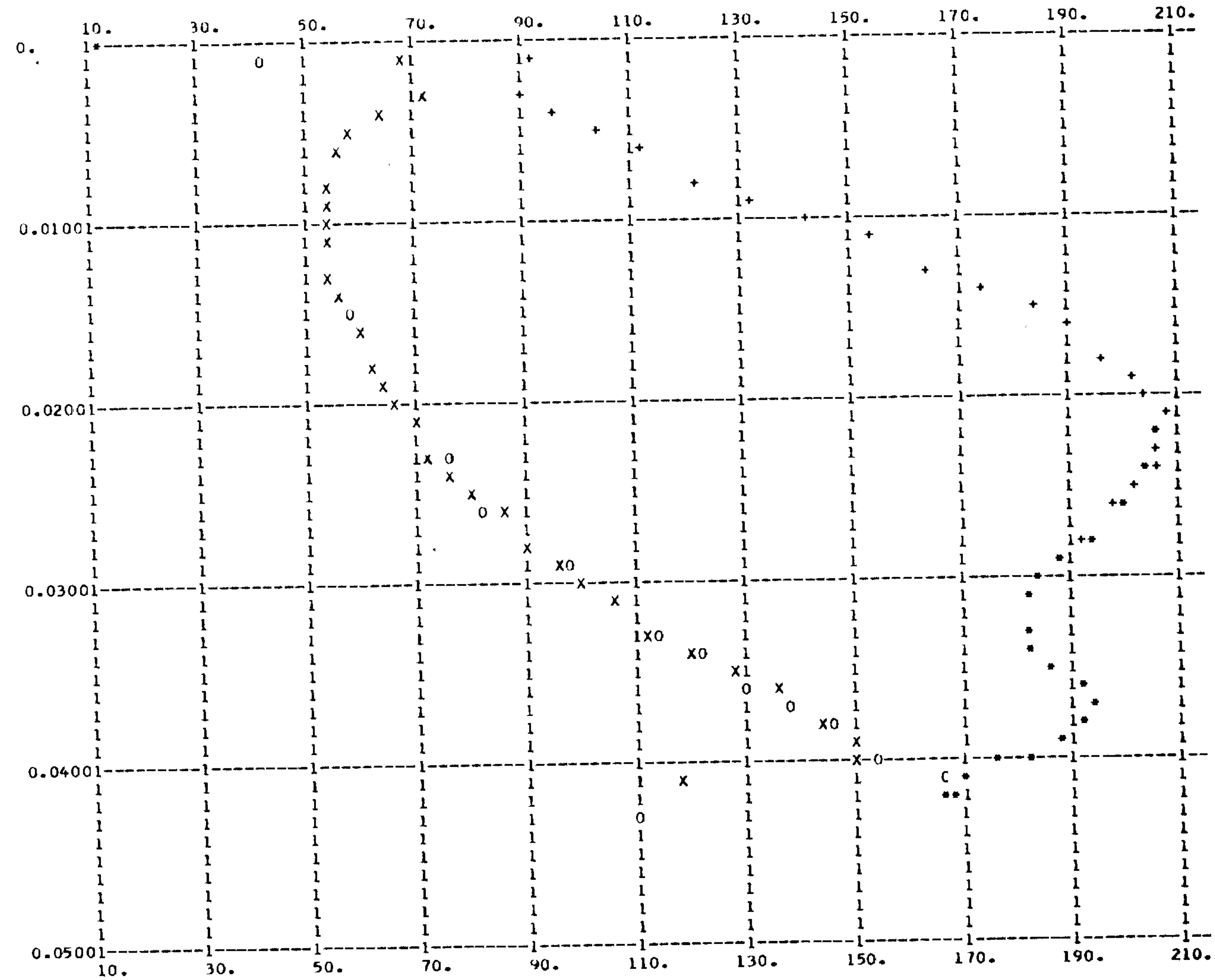
	BLADE SURFACE 1				BLADE SURFACE 2			
	VELOCITY	ANGLE(DEG)	SURF. LENGTH	W/WCR	VELOCITY	ANGLE(DEG)	SURF. LENGTH	W/WCR
0	0	90.00	0	0	0	-90.00	0	0
0.1254E-02	92.767	42.13	0.3092E-02	0.2986	68.633	-42.13	0.3092E-02	0.2209
0.2509E-02	90.907	27.31	0.4577E-02	0.2926	71.774	-19.97	0.4556E-02	0.2310
0.3763E-02	96.390	24.65	0.5973E-02	0.3103	63.451	-13.52	0.5853E-02	0.2043
0.5018E-02	104.07	21.66	0.7337E-02	0.3350	58.434	-13.33	0.7142E-02	0.1881

15

SURFACE VELOCITIES BASED ON TANGENTIAL COMPONENTS
REDUCED WEIGHT FLOW

	BLADE SURFACE 1		
	VELOCITY	ANGLE(DEG)	W/WCR
0	11.360	90.00	0.3657E-01
0.6144E-03	67.245	57.01	0.2165
0.3582E-02	93.585	25.05	0.3013
0.1906E-01	203.15	-29.60	0.6540
0.2201E-01	206.00	-40.10	0.6631

BLADE SURFACE VELOCITIES FOR REDUCED WEIGHT FLOW



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

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

15

TABLE V. - Concluded. SAMPLE OUTPUT FOR AXIAL FLOW TURBINE STATOR CASE

16{ TIME = 3.1942 MIN.

17

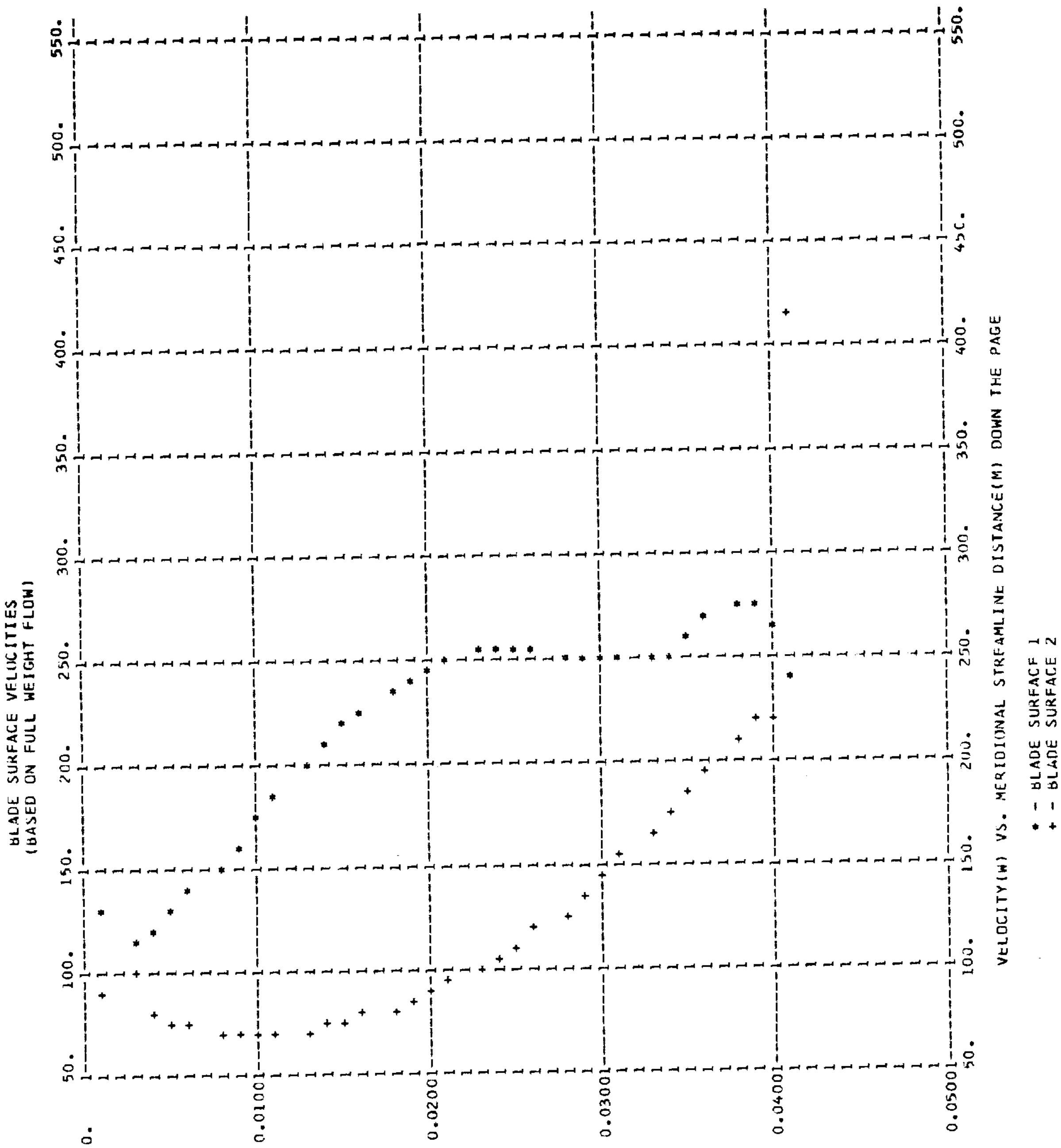
VELOCITIES AT INTERIOR MESH POINTS
(BASED ON FULL WEIGHT FLOW)

IM = 1	ITI = 0								
68.261	68.942	69.972	71.079	72.265	73.345	74.214	74.804	75.086	75.055
74.731	74.150	73.359	72.414	71.383	70.343	69.386	68.610	68.115	67.987
IM = 2	ITI = 0								
68.269	68.942	69.922	71.079	72.265	73.345	74.214	74.804	75.086	75.055
74.731	74.150	73.359	72.414	71.383	70.343	69.386	68.610	68.115	67.987
IM = 3	ITI = 0								
68.127	68.833	69.865	71.080	72.316	73.432	74.323	74.923	75.204	75.165
74.829	74.231	73.420	72.451	71.391	70.317	69.321	68.505	67.977	67.831

18

SURFACE VELOCITIES BASED ON MERIDIONAL COMPONENTS - FULL WEIGHT FLOW

	BLADE SURFACE 1		BLADE SURFACE 2	
M	VELOCITY	W/WCR	VELOCITY	W/WCR
0.1254E-02	131.29	0.4226	91.073	0.2932
0.2509E-02	112.81	0.3632	101.71	0.3274
0.3763E-02	119.33	0.3841	79.076	0.2546
0.5018E-02	128.84	0.4147	75.463	0.2429
0.6272E-02	139.08	0.4477	72.647	0.2339
0.7526E-02	150.03	0.4830	70.493	0.2269
0.8781E-02	161.65	0.5204	69.206	0.2228
0.1004E-01	173.75	0.5593	69.349	0.2232
0.1129E-01	186.06	0.5990	70.102	0.2257
0.1254E-01	197.91	0.6371	71.480	0.2301
0.1380E-01	208.60	0.6715	73.405	0.2363
0.1505E-01	218.05	0.7019	75.772	0.2439
0.1631E-01	226.50	0.7291	78.581	0.2530
0.1756E-01	234.32	0.7543	81.884	0.2636
0.1882E-01	241.79	0.7783	85.602	0.2756
0.2007E-01	245.95	0.7917	89.865	0.2893
0.2132E-01	250.86	0.8076	94.623	0.3046
0.2258E-01	252.71	0.8135	99.933	0.3217
0.2383E-01	256.40	0.8254	105.77	0.3405
0.2509E-01	256.21	0.8248	112.19	0.3612
0.2634E-01	254.64	0.8197	119.18	0.3836
0.2760E-01	251.72	0.8103	126.70	0.4079
0.2885E-01	251.15	0.8085	134.66	0.4335
0.3011E-01	248.53	0.8000	143.39	0.4616
0.3136E-01	247.89	0.7980	152.73	0.4917
0.3261E-01	248.42	0.7997	162.76	0.5239
0.3387E-01	251.03	0.8081	173.16	0.5574
0.3512E-01	260.78	0.8395	185.03	0.5956
0.3638E-01	270.18	0.8697	196.93	0.6339
0.3763E-01	276.10	0.8888	209.72	0.6751
0.3889E-01	273.96	0.8819	218.35	0.7029
0.4014E-01	263.68	0.8488	217.54	0.7003
0.4140E-01	240.48	0.7741	413.83	1.3322


BLADE SURFACE VELOCITIES
(BASED ON FULL WEIGHT FLOW)
50.
100.
150.
200.
250.
300.
350.
400.
450.
500.
550.
0.
0.0100
0.0200
0.0300
0.0400
0.0500
VELOCITY(W) VS. MERIDIONAL STREAMLINE DISTANCE(M) DOWN THE PAGE
* - BLADE SURFACE 1
+ - BLADE SURFACE 2

18

printed value of the estimate optimum ORF is the value of the overrelaxation factor (ORF) used by the program.

(9) This is the output corresponding to ERSOR.

(10) This is the output corresponding to STRFN.

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

(12) This is the output corresponding to SLCRD.

(13) This is the output corresponding to INTVL for the reduced weight flow.

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

(15) This is the output corresponding to SURVL for the reduced weight flow.

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

Most of the previously described output has been for the reduced weight flow. The following output is for the actual weight flow.

(17) This is the output corresponding to INTVL for the full weight flow.

(18) This is the output corresponding to SURVL for the full weight flow.

ERROR CONDITIONS

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

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

2. BLCD CALL NO. XX

M COORDINATE IS NOT WITHIN BLADE

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

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

3. ROOT CALL NO. XX

ROOT HAS FAILED TO OBTAIN A VALID ROOT

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

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

4. DENSTY CALL NO. XX

NER(1) = XX

RHO*W IS X.XXXX TIMES THE MAXIMUM VALUE FOR RHO*W

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

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

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

5. MM, NBBI, NRSP, OR SOME SPLNO IS TOO LARGE

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

6. INPUT WEIGHT FLOW (WTFL) IS TOO LARGE AT BLADE LEADING EDGE

This message is printed if WTFL is greater than the choking mass flow for the vertical line BG, and the program is stopped. If this happens, there is probably an error in the input. The following items should be checked carefully: RHOIP, WTFL, BETAI, NBL, RMSP, and BESP.

7. REDUCED WEIGHT FLOW IS STILL TOO LARGE

This message is printed if difficulty is encountered in calculating β_{in} or β_{out} for the reduced weight flow. If this happens, REDFAC should be reduced.

8. ONE OF THE MH ARRAYS IS TOO LARGE

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

9. THE NUMBER OF INTERIOR MESH POINTS EXCEEDS 2500

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

10. SEARCH CANNOT FIND M IN THE MH ARRAY

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

11. A VELOCITY-GRADIENT SOLUTION CANNOT BE OBTAINED FOR VERTICAL LINE IM = XX

This message is printed if difficulty is encountered in solving the velocity-gradient equation for some vertical line.

12. A VELOCITY-GRADIENT SOLUTION COULD NOT BE OBTAINED IN 50 ITERATIONS FOR VERTICAL LINE IM = XX

This message is printed after 50 attempts to find a velocity-gradient equation which results either in the specified weight flow (WTFL) or in a choked flow.

13. WTFL EXCEEDS CHOKING WEIGHT FLOW FOR IM = XX

CHOKING WEIGHT FLOW = XXXXX FOR IM = XX

This message is printed if the vertical line IM will not pass the specified weight flow (WTFL). WTFL should be reduced in this case.

PROGRAM PROCEDURE

The first part of the program is very similar to TURBLE (ref. 3). The program description for TURBLE is given in reference 1. The main difference in this part of the TSONIC is the calculation of coefficients A and B of equations (4) to (6) by subroutine TANG. Also, PRECAL has been considerably changed to calculate certain constants at reduced weight flow. In addition, a new segment was added to solve the velocity-gradient equation. The main subroutine of this new segment is TVELCY. PRECAL, TANG, TVELCY and the subroutines in the new segment of the program are described later in this section. All the subroutines and their relation are shown in figure 14.

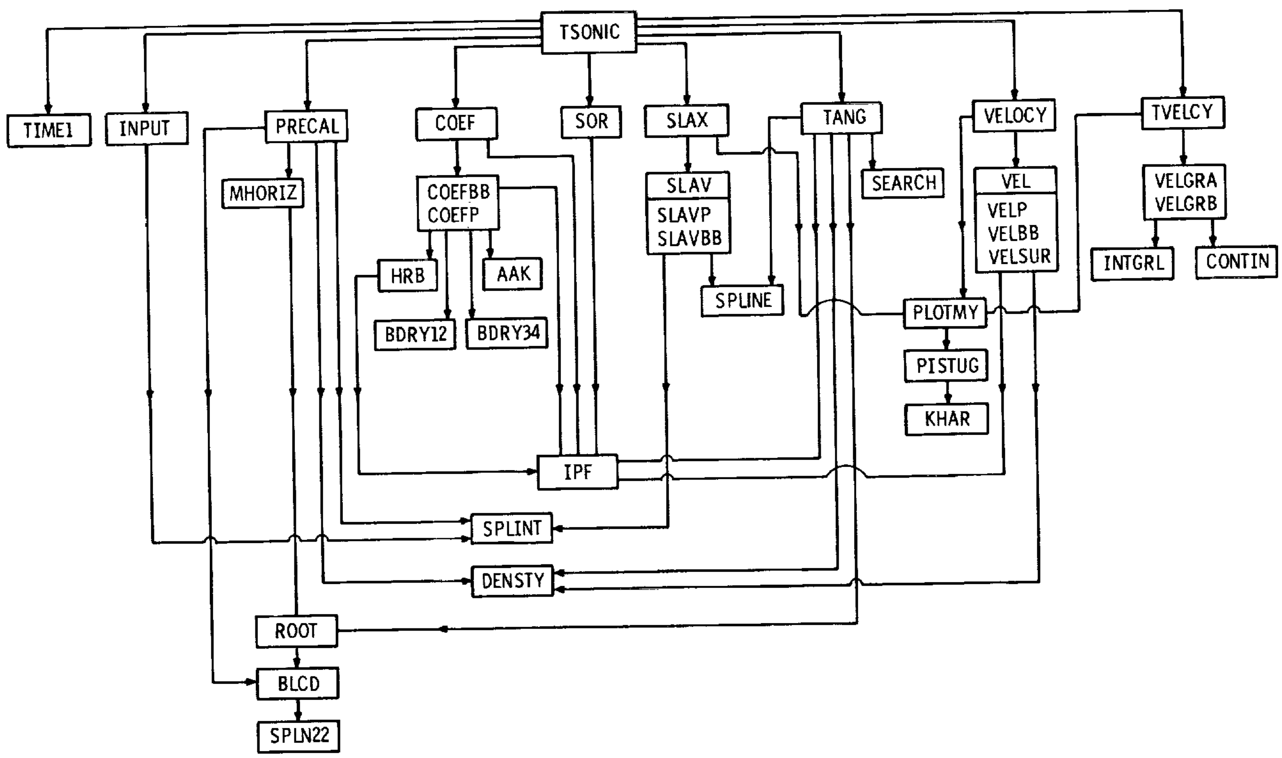


Figure 14. - Calling relation of subroutines.

The dictionary defines all new variables. Any variables not defined here are defined in reference 1.

The program can handle up to 2500 mesh points on the IBM 7094-2/7044 direct coupled system with a 32 768-word core. To be able to handle 2500 mesh points, an overlay arrangement is used, as shown in figure 15. All subroutines not shown are in the main link. If there is a storage problem on the user's computer, the maximum

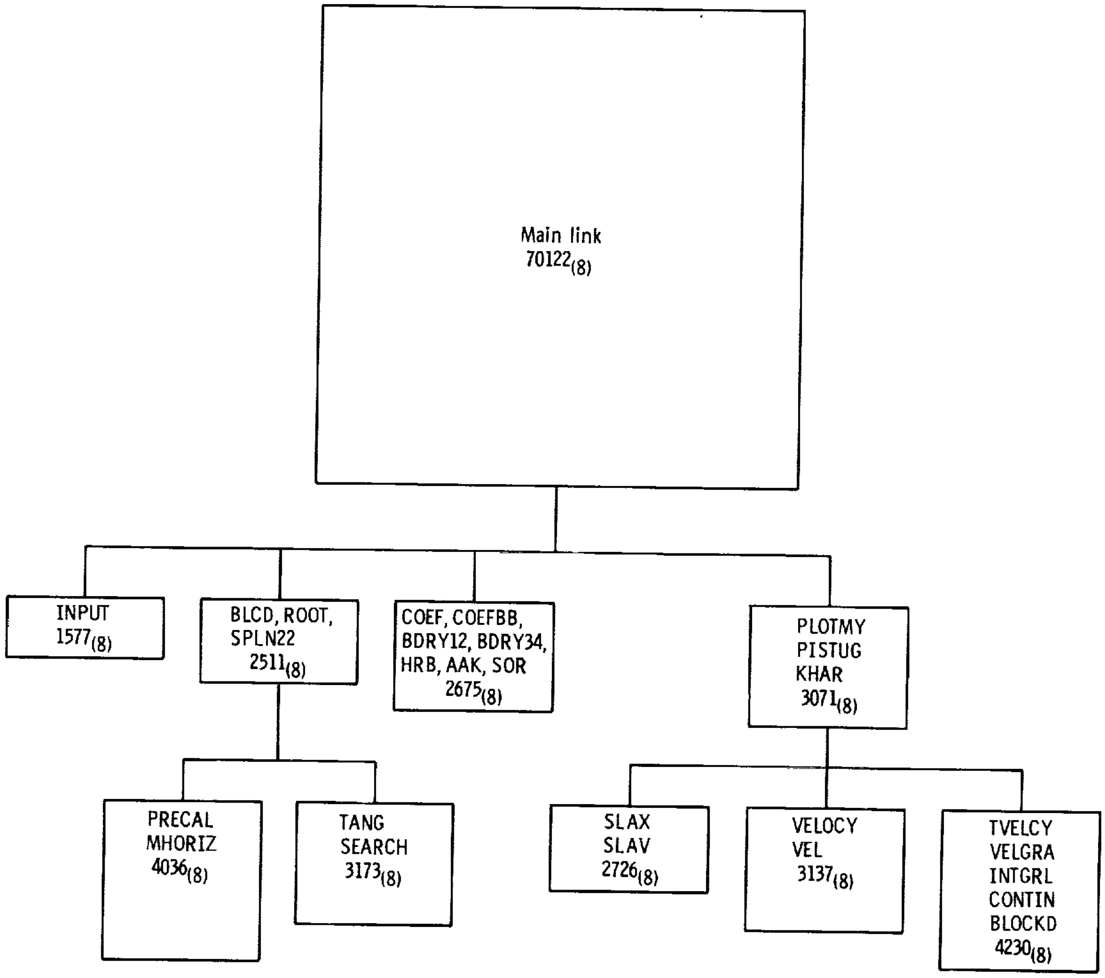


Figure 15. - Arrangement for overlay, showing octal storage requirements.

number of mesh points should be reduced. The following program changes are required to change the maximum number of mesh points:

(1) Change the dimension of A, U, K, and RHO in the COMMON/AUKRHO/ statement. This statement occurs in most subroutines.

(2) In subroutine INPUT, change the number of values of U, K, and RHO to be initialized (the bound on the DO loop near statement 60).

(3) In subroutine PRECAL, change statement 340 and format statement 1150 to reflect the maximum allowable number of mesh points. Statement 340 will cause the program to stop if there are too many mesh points.

(4) Change the dimensions of W, BETA, DUDT, DUDTT, AAP, and BBP in SLAX, SLAV, TANG, VELOCY, VEL, TVELCY, and VELGRA.

The conventions used in the program and most of the labeled COMMON blocks are described in reference 1. The following are new COMMON blocks in TSONIC:

(1) /SURVEL/ is used for arrays of surface velocities and plotting arrays.

(2) /D2TDM2/ is used for an array of second derivatives $d^2\theta/dm^2$ along the blade surfaces.

(3) /WWCRM/ is used for an array of critical velocity ratios and an array of labels to indicate choking.

Subroutine PRECAL

Calculation of λ, W_{le}, and W_{te}. - The prerotation λ and the average velocity W_{le} at the blade leading edge are calculated for the full weight flow. They are calculated iteratively by equations (B7) to (B9) of reference 1. If the input weight flow (WTFL) is too large, a solution cannot be obtained. In this case, error message (6) is printed, and the program is stopped. Otherwise, W_{te} is calculated using β_{te} by satisfying one-dimensional continuity at line CF.

Calculation of W_{cr} and maximum values of mass flow parameter ρW. - Calculation of all these quantities at the leading and trailing edges is done using equations (B10) to (B12) of reference 1.

Calculation of w, ω, and λ for reduced weight flow. - First, the values of w, ω, and λ for the full weight flow are stored. Then values of w and ω for the reduced weight flow are calculated by multiplying by REDFAC. The value of λ for the reduced weight flow is then calculated by the same procedure as for the full weight flow.

Calculate β_{in} and β_{out}. - These quantities are necessary as boundary conditions. The input, however, gives β_{le} (BETAI) and β_{te} (BETAO). The desired values of β_{in} and β_{out} are calculated from the input values of β_{le} and β_{te}, respectively, by using equation (B14) of reference 1. The reduced weight flow is used in these calculations for reasons explained in appendix B.

The remaining calculations in PRECAL are the same as for TURBLE, and are described in reference 1.

Subroutine TANG

Most of this subroutine is the same as in TURBLE, and is described in reference 1. After these calculations are complete for a given horizontal mesh line segment, there is a check to see if there is convergence of the outer iteration (by checking the value of IEND). If convergence has been achieved, the coefficients A and B of equation (4) are calculated. This requires first the calculation of $\partial^2 u/\partial m\,\partial\theta$ and $\partial W/\partial m$. The calculation of $\partial^2 u/\partial m\,\partial\theta$ is done by using subroutine SPLINE to calculate the partial derivative with respect to m of $\partial u/\partial\theta$. The values of $\partial u/\partial\theta$ have been previously calculated in SLAV. The values of $\partial W/\partial m$ are also calculated by subroutine SPLINE. The values of A and B are now calculated at each point by using equations (5) and (6). These values are stored in the AAP and BBP arrays.

Subroutine TVELCY

TVELCY calls VELGRA and VELGRB to solve the velocity-gradient equation along each vertical line. After this is done, the blade surface velocity is printed at each vertical mesh line. Then these velocities are plotted.

Subroutine VELGRA

VELGRA has a second entry point. The main entry point is for vertical mesh lines upstream or downstream of the blade. The second entry point is called VELGRB and is used for vertical mesh lines between the blades. Most calculations are the same for either entry point. The variable AORB is used as a switch to indicate the entry point, and is used where there is some difference in the calculation, such as at the surface of a blade.

After calculating some constants the values of A and B of equation (4) are placed in the A2 and B2 arrays from the AAP and BBP arrays, respectively, for the interior points. The values of A and B on the blade surface are calculated by equations (5a) and (6).

After all the values of A and B are calculated, the velocity-gradient equation (4) is solved. An initial estimate of W on the lower boundary is available from the reduced weight flow solution. The initial velocity is obtained by dividing the reduced weight flow value by REDFAC. A numerical solution to equation (4) is calculated by a Runge-Kutta method as follows (ref. 10, p. 233). If W_j is known at the j^{th} point, W_{j+1} is calculated by the following algorithm. Let

$$\left.\begin{aligned} W_{j+1}^{*} &= W_j + (A_jW_j + B_j)(\theta_{j+1} - \theta_j) \\ W_{j+1}^{**} &= W_j + (A_{j+1}W_{j+1}^{*} + B_{j+1})(\theta_{j+1} - \theta_j) \end{aligned}\right\} \qquad (8)$$

Then

$$W_{j+1} = \frac{W_{j+1}^{*} + W_{j+1}^{**}}{2}$$

After the second boundary is reached, the solution is checked by calculating the weight flow. The weight flow can be calculated by

$$w_{est} = \int_{\theta_1}^{\theta_2} \rho W \cos \beta \, br \, d\theta \tag{9}$$

using the values of W just calculated. The density ρ for each W is calculated, and the value of β from the reduced weight flow solution is used. The stream-channel thickness b and the radius are constants which are available in the BE and RM arrays, respectively. The integral in equation (9) is calculated numerically by subroutine INTGRL. After w_{est} (WTFLES in program) is calculated, subroutine CONTIN is called. CONTIN will give a new initial value for W. The entire procedure is then repeated, with CONTIN giving a new initial value for W each time, until one of four occurrences:

(1) WTFLES = $w \pm w/10^5$.

(2) A maximum value of WTFLES $< w$ is found. This indicates choking. In this case, error message 13 is printed.

(3) In the calculations W becomes so large that no corresponding density can be found. In this case, error message 11 is printed, and the program goes to the next vertical line.

(4) None of the above conditions are met for 50 iterations. In this case, error message 12 is printed, and the program goes to the next vertical line.

After the calculations are completed, the interior velocities are printed, and surface velocities are stored to be printed later.

Subroutine BLOCKD

This initializes the LABEL array with blanks.

Main Dictionary

Most of the FORTRAN variables are the same as those given in reference 1 and will not be repeated here. The following list defines all new variables in the previously discussed subroutines.

A2 — array of values of coefficient A (eq. (4)) along a vertical mesh line

AAP — array of values of coefficient A (eq. (4)) at all interior mesh points

ACTLAM — value of λ based on input value of weight flow w (WTFL) (LAMBDA is based on the reduced weight flow for the reduced weight flow calculations.)

ACTOMG — input value of OMEGA (OMEGA is reduced for the reduced weight flow calculations.)

ACTWT — input value of WTFL (WTFL is reduced for the reduced weight flow calculations.)

AORB — switch in VELGRA to indicate entry point (AORB = 1 for main entry point and AORB = 2 for entry point VELGRB.)

B2 — array of values of coefficient B (eq. (4)) along a vertical mesh line

BBP — array of values of coefficient B (eq. (4)) at all interior mesh points

CBETA — array of values of $\cos\beta$ along a vertical mesh line

CHOKED — variable storing the word "choked" in Hollerith

D2TDM2 — array of values of $d^2\theta/dm^2$ for each blade surface at vertical mesh lines

DDT — array of value of $\partial u/\partial\theta$ along a horizontal line

DELMAX — maximum permitted change of estimated initial value of W ($DELMAX = W_{cr}/10$)

DENTOL — see input

DUDMM — array of values of $\partial^2 u/\partial m^2$ along a horizontal mesh line

DUDTM — array of values of $\partial^2 u/\partial\theta\,\partial m$ along a horizontal line

DUDTT — array of values of $\partial^2 u/\partial\theta^2$ at all interior mesh points

DWDM — array of values of $\partial W/\partial m$ along a horizontal mesh line

I2 — temporary index variable in TVELCY

IND — integer variable controlling logical sequence in CONTIN

LABEL — array of labels for A format output to indicate a particular blade surface velocity was based on choked weight flow

NERT — temporary storage of a value of NER(1)

REDFAC — see input

RWCB — array of values of $\rho W\cos\beta$ along a vertical mesh line

SBETA — $\sin\beta$

SBETA1 — $\sin\beta$ on the upper blade surface at a given vertical line

SBETAN — $\sin\beta$ on the lower blade surface at a given vertical line

THETA — array of θ-coordinates along a vertical mesh line

TOLERC	velocity tolerance for convergence in calculating choking weight flow (TOLERC = W/100)
TORSAL	$2\omega r \sin \alpha$
VT	temporary velocity
WAS	W^{*}_{j+1}, eq. (8)
WASS	W^{**}_{j+1}, eq. (8)
WGRAD	array of velocities W calculated by eq. (8)
WIP	array of velocities W along a horizontal mesh line in TANG
WTFLES	w_{est} calculated by eq. (9)

Program Listing For Subroutines Using Main Dictionary

```
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,REDFAC,
     1  DENTOL,MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2  BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1  HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2  TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3  TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4  BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5  SAL(100),AAA(100)
      COMMON /GEOMIN/ CHORD(2),STGR(2),MLE(2),THLE(2),RMI(2),RMO(2),
     1  RI(2),RO(2),BETI(2),BETO(2),NSPI(2),MSP(50,2),THSP(50,2)
      COMMON /RHOS/RHOHB(100,2),RHOVB(100,2)
      COMMON /BLCDCM/ EM(50,2),INIT(2)
      COMMON /SURVEL/ WTB(100,2),WMB(100,2),XDOWN(400),YACROS(400)
      COMMON /D2TDM2/ D2TDM2(100,2)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1  UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      CALL TIME1(T1)
   10 IEND = -1
      ITER = 0
      INIT(1) = 0
      INIT(2) = 0
      CALL INPUT
      CALL PRECAL
   30 CALL COEF
      CALL SOR
      CALL TIME1(T2)
      TIME= (T2-T1)/3600.
      WRITE(6,1000) TIME
      CALL SLAX
      CALL TANG
```

```
      CALL VELOCY
      CALL TIME1(T2)
      TIME= (T2-T1)/3600.
      WRITE(6,1000) TIME
      IF(NER(2).GT.0) GO TO 10
      IF(IEND.LE.0) GO TO 30
      CALL TVELCY
      GO TO 10
 1000 FORMAT (8HLTIME = ,F7.4,5H MIN.)
      END
```

```
      SUBROUTINE INPUT
C
C  INPUT READS AND PRINTS ALL INPUT DATA CARDS AND CALCULATES HORIZONTAL
C  SPACING (MV ARRAY)
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,REDFAC,
     1   DENTOL,MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2   TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3   TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4   BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5   SAL(100),AAA(100)
      COMMON /GEOMIN/ CHORD(2),STGR(2),MLE(2),THLE(2),RMI(2),RMO(2),
     1   RI(2),RO(2),BETI(2),BETO(2),NSPI(2),MSP(50,2),THSP(50,2)
      COMMON /RHOS/RHOHB(100,2),RHOVB(100,2)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
C
C  READ AND PRINT ALL INPUT DATA
C
      WRITE(6,1000)
      READ(5,1100)
      WRITE(6,1100)
      WRITE(6,1110)
      READ (5,1030) GAM,AR,TIP,RHOIP,WTFL,BLANK,OMEGA,ORF
      WRITE(6,1040) GAM,AR,TIP,RHOIP,WTFL,BLANK,OMEGA,ORF
      WRITE(6,1120)
      READ (5,1030)BETAI,BETAO,CHORD(1),STGR(1)
      WRITE(6,1040)BETAI,BETAO,CHORD(1),STGR(1)
      WRITE (6,1125)
      READ  (5,1030) REDFAC,DENTOL
      IF(DENTOL.LE.0.) DENTOL = .001
      WRITE (6,1040) REDFAC,DENTOL
      WRITE(6,1130)
      READ (5,1010) MBI,MBO,BLANK,BLANK,MM,NBBI,NBL,NRSP
      WRITE(6,1010) MBI,MBO,BLANK,BLANK,MM,NBBI,NBL,NRSP
      DO 10 J=1,2
      IF (J.EQ.1) WRITE(6,1140)
```

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

```
C
C  CALCULATE GEOMETRICAL CONSTANTS
C
      CHORD(2) = CHORD(1)
      STGR(2) = STGR(1)
      MLE(1) = 0.
      MLE(2) = 0.
      THLE(1) = 0.
      THLE(2) = PITCH
      RMI(1) = RM(MBI)
      RMI(2) = RM(MBI)
      RMO(1) = RM(MBO)
      RMO(2) = RM(MBO)
C
C  INITIALIZE ARRAYS
C
      DO 60 I=1,2500
      L(I) = 1.
      K(I) = 0.
   60 RHO(I) = RHOIP
      DO 70 IM=1,100
      DO 70 SURF=1,2
      RHOHB(IM,SURF) = RHOIP
   70 RHOVB(IM,SURF) = RHOIP
      RETURN
 1000 FORMAT (1H1)
 1010 FORMAT (16I5)
 1020 FORMAT (1X,16I7)
 1030 FORMAT (8F10.5)
 1040 FORMAT (1X,8G16.7)
 1100 FORMAT (80H
     1                                                               )
 1110 FORMAT (7X,3HGAM,14X,2HAR,13X,3HTIP,12X,5HRHOIP,12X,4HWTFL,11X,6H
     1      ,10X,5HOMEGA,12X,3HORF)
 1120 FORMAT (6X,5HBETAI,10X,5HBETAO,11X,6HCHORDF,11X,5HSTGRF)
 1125 FORMAT (6X,6HREDFAC,10X,6HDENTOL)
 1130 FORMAT (41H    MBI   MBO                  MM   NBBI   NBL NRSP)
 1140 FORMAT (39HL     BLADE SURFACE 1  --  UPPER SURFACE)
 1150 FORMAT (39HL     BLADE SURFACE 2  --  LOWER SURFACE)
 1180 FORMAT (7X,2HRI,I1,12X,2HRO,I1,12X,4HBETI,I1,11X,4HBETO,I1,11X,5HS
     1FLNO,I1)
 1190 FORMAT (7X,3HMSP,I1,2X,5HARRAY)
 1200 FORMAT (7X,4HTHSP,I1,2X,5HARRAY)
 1210 FORMAT (16HL          MR  ARRAY)
 1220 FORMAT (7X,11HRMSP   ARRAY)
 1230 FORMAT (7X,11HBESP   ARRAY)
 1240 FORMAT (52HL     BLDAT   AANDK   ERSOR   STRFN   SLCRD   INTVL   SURVL)
 1250 FORMAT (41H1 MM,NBBI,NRSP,OR SOME SPLNO IS TOO LARGE)
      END
```

```
      SUBROUTINE PRECAL
C
C   PRECAL CALCULATES ALL REQUIRED FIXED CONSTANTS
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,REDFAC,
     1   DENTOL,MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2   TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3   TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4   BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5   SAL(100),AAA(100)
      COMMON /D2TDM2/ D2TDM2(100,2)
      DIMENSION CURV(100,2)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLF,MR,MSL,MSP,MV,MVIM1
      EXTERNAL BL1,BL2
C
C   CALCULATE LAMBDA, VI AND VO
C
      BETAI = BETAI/57.295779
      BETAO = BETAO/57.295779
      TBI = SIN(BETAI)/COS(BETAI)
      TBO = SIN(BETAO)/COS(BETAO)
   10 RHOT = RHOIP
      RHOVI = WTFL/BE(MBI)/PITCH/COS(BETAI)/RM(MBI)
   20 VI = RHOVI/RHOT
      LAMBDA = RM(MBI)*(VI*SIN(BETAI)+OMEGA*RM(MBI))
      TTIP = 1.-(VI**2+2.*OMEGA*LAMBDA-(OMEGA*RM(MBI))**2)/CPTIP
      IF (TTIP.LE.0.) GO TO 30
      RHOMBI = RHOIP*TTIP**EXPON
      IF(ABS(RHOMBI-RHOT)/RHOIP.LT..000001) GO TO 40
      RHOT = RHOMBI
      GO TO 20
   30 WRITE(6,1020) WTFL
      STOP
   40 VI = RHOVI/RHOMBI
      LAMBDA = RM(MBI)*(VI*SIN(BETAI)+OMEGA*RM(MBI))
      TWL = 2.*OMEGA*LAMBDA
      RHOVO = WTFL/BE(MBO )/PITCH/COS(BETAO)/RM(MBO)
      RHOMB2 = RHOIP
      TWLMR = TWL-(OMEGA*RM(MBO ))**2
      LER(1)=1
C     DENSTY CALL NO. 1
      CALL DENSTY(RHOVO,RHOMB2,VO,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP)
C
C   CALCULATE W-CRITICAL, AND MAXIMUM VALUES FOR RHO*W AT LEADING AND
C   TRAILING EDGE
C
      AA = (TWL-(OMEGA*RM(MBI))**2)/CPTIP
      TPP = TIP*(1.-AA)
      BB = TGROG*TPP
      WCRI = SQRT(BB)
      TTIP = 1.-BB/CPTIP-AA
      RHOWMI = RHOIP*TTIP**EXPON*WCRI
```

```
      AA = (TWL-(OMEGA*RM(MBO ))**2)/CPTIP
      TPP = TIP*(1.-AA)
      BB = TGROG*TPP
      WCRO = SQRT(BB)
      TTIP = 1.-BB/CPTIP-AA
      RHOWMO = RHOIP*TTIP**EXPON*WCRO
C
C     STORE ACTUAL VALUES OF WTFL, OMEGA AND LAMBDA - CALCULATE REDUCED
C     VALUES OF WTFL AND LAMBDA
C
      ACTWT = WTFL
      ACTOMG = OMEGA
      ACTLAM = LAMBDA
      WTFL  =  WTFL*REDFAC
      OMEGA = OMEGA*REDFAC
C
C     CALCULATE LAMBDA FOR REDUCED WEIGHT FLOW
C
      RHOT = RHOIP
      RHOVI = WTFL/BE(MBI)/PITCH/COS(BETAI)/RM(MBI)
   50 VT = RHOVI/RHOT
      LAMBDA = RM(MBI)*(VT*SIN(BETAI)+OMEGA*RM(MBI))
      TTIP = 1.-(VT**2+2.*OMEGA*LAMBDA-(OMEGA*RM(MBI))**2)/CPTIP
      RHOMBI = RHOIP*TTIP**EXPON
      IF(ABS(RHOMBI-RHOT)/RHOIP.LT..000001) GO TO 60
      RHOT = RHOMBI
      GO TO 50
   60 VT = RHOVI/RHOMBI
      LAMBDA = RM(MBI)*(VT*SIN(BETAI)+OMEGA*RM(MBI))
      TWL = 2.*OMEGA*LAMBDA
C
C     CALCULATE BETA CORRECTED TO BOUNDARY A-N AND G-H USING REDUCED
C     WEIGHT FLOW
C
      NERT = NER(1)
      TWLMR = TWL-(OMEGA*RM(1))**2
      RHO1 = RHOMBI
      TBI1 = 1.E20
   70 TBIT = (TBI/BE(MBI)*RHO1/RHOMBI+OMEGA*(RM(MBI)**2-RM(1)**2)*RHO1
     1    /WTFL*PITCH)*BE(1)
      IF(ABS(TBI1-TBIT).LT..00001) GO TO 80
      TBI1 = TBIT
      RHOVI = WTFL/PITCH*SQRT(1.+TBI1**2)/BE(1)/RM(1)
      LER(1)=2
C     DENSTY CALL NO. 2
      CALL DENSTY (RHOVI,RHO1,AA,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP)
      GO TO 70
   80 TBI = TBIT
      RHOVO = WTFL/BE(MBO )/PITCH/COS(BETAO)/RM(MBO)
      RHOMB2 = RHOIP
      TWLMR = TWL-(OMEGA*RM(MBO ))**2
      LER(1)=3
C     DENSTY CALL NO. 3
      CALL DENSTY(RHOVO,RHOMB2,AA,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP)
      BTAIN = ATAN(TBI)*57.295779
      TWLMR = TWL-(OMEGA*RM(MM))**2
      RHOMM = RHOMB2
      TBOM = 1.E20
   90 TBOT = (TBO/BE(MBO )*RHOMM/RHOMB2+OMEGA*(RM(MBO )**2-RM(MM)**2)*
```

```
     1    RHOMM/WTFL*PITCH)*BE(MM)
      IF (ABS(TBOM-TBOT).LT..00001) GO TO 100
      TBOM = TBOT
      RHOVO = WTFL/PITCH*SQRT(1.+TBOM**2)/BE(MM)/RM(MM)
      LER(1)=4
C     DENSTY CALL NO. 4
      CALL DENSTY (RHOVO,RHOMM,AA,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP)
      GO TO 90
  100 TBO = TBOT
      BTAOUT = ATAN(TBO)*57.295779
      IF(NER(1).EQ.NERT) GO TO 110
      WRITE (6,1025)
      STOP
C
C  CALCULATE TV, ITV, IV, DTDMV, AND BETAV ARRAYS
C
  110 ITMIN = 0
      ITMAX = NBBI-1
C  TV, ITV, AND DTDMV ON BLADE
      DO 120 IM=MBI,MBO
      LER(2)=1
C     BLCD CALL NO. 1
      CALL BL1(MV(IM),TV(IM,1),DTDMV(IM,1),INF)
      ITV(IM,1)= INT((TV(IM,1)+DTLR)/HT)
      IF (TV(IM,1).GT.-DTLR) ITV(IM,1)=ITV(IM,1)+1
      ITMIN= MINO(ITMIN,ITV(IM,1))
      LER(2)=2
C     BLCD CALL NO. 2
      CALL BL2(MV(IM),TV(IM,2),DTDMV(IM,2),INF)
      ITV(IM,2)= INT((TV(IM,2)-DTLR)/HT)
      IF (TV(IM,2).LT.DTLR) ITV(IM,2)=ITV(IM,2)-1
  120 ITMAX= MAXO(ITMAX,ITV(IM,2))
C   ITV AND IV UPSTREAM OF BLADE
      FIRST = 0
      LAST = NBBI-1
      DO 130 IM=1,MBIM1
      ITV(IM,1)= FIRST
  130 ITV(IM,2)= LAST
C   ITV DOWNSTREAM OF BLADE
  140 LAST= ITV(MBO,2)
      FIRST= LAST+1-NBBI
      DO 150 IM=MBOP1,MM
      ITV(IM,1) = FIRST
  150 ITV(IM,2) = LAST
      ITMIN = MINO(ITMIN,ITV(MM,1))
C   CALCULATE IV ARRAY
      IV(1) = 1
      DO 160 IM=1,MM
  160 IV(IM+1) = IV(IM)+ITV(IM,2)-ITV(IM,1)+1
C  BETAV ARRAY
      DO 200 SURF=1,2
      DO 200 IM=MBI,MBO
      CURV(IM,SURF) = (RM(IM)*D2TDM2(IM,SURF)+SAL(IM)*DTDMV(IM,SURF)) /
     1    (1.+(RM(IM)*DTDMV(IM,SURF))**2)**1.5
  200 BETAV(IM,SURF) = ATAN(DTDMV(IM,SURF)*RM(IM))*57.295779
      NIP = IV(MM)+NBBI-1
      WRITE(6,1030) VI,RHOWMI,WCRI,BTAIN,VO,RHOWMO,WCRO,BTAOUT
      WRITE(6,1040) PITCH,HT,HM1
      WRITE(6,1050) ITMIN,ITMAX,ACTLAM,LAMBDA,NIP
      WRITE(6,1060) (SURF,BV(SURF),SURF=1,2)
```

```
      IF(BLDAT.LE.0) GO TO 230
      WRITE (6,1070)
      WRITE (6,1080) (MV(IM),TV(IM,1),DTDMV(IM,1),CURV(IM,1),TV(IM,2),
     1   DTDMV(IM,2),CURV(IM,2),IM=MBI,MBO)
      WRITE (6,1090) (IM,MV(IM),RM(IM),SAL(IM),BE(IM),DBDM(IM),IM=1,MM)
  230 CONTINUE
C
C  CALCULATE MH AND DTDMH ARRAYS
C
      ITO = ITV(1,1)
      MRTS = 1
      IMS(1) = 1
      MH(1,1) = 0.
      DTDMH(1,1) = 1.E10
      LER(2) = 3
C     BLCD AND ROOT (VIA MHORIZ) CALL NO. 3
      CALL MHORIZ(MV,ITV(1,1),BL1,MBI,MBO,ITO,HT,DTLR,0,IMS(1),MH(1,1),
     1   DTDMH(1,1),MRTS)
      IF (ITV(MBO,1)-ITV(MBO,2)+NBBI.NE.2) GO TO 240
      IMSL = IMS(1)+1
      MH(IMSL,1) = MV(MBO)
      DTDMH(IMSL,1) = -1.E10
      IMS(1) = IMSL
  240 IMS(2) = 0
      MRTS = 1
      LER(2) = 4
C     BLCD AND ROOT (VIA MHORIZ) CALL NO. 4
      CALL MHORIZ(MV,ITV(1,2),BL2,MBI,MBO,ITO,HT,DTLR,1,IMS(2),MH(1,2),
     1   DTDMH(1,2),MRTS)
      I = MAXO(IMS(1),IMS(2))
      IF(I.LE.100) GO TO 290
      WRITE(6,1100) I
      STOP
  290 IF(BLDAT.LE.0) GO TO 300
      WRITE (6,1110)  (IM,IV(IM),(ITV(IM,SURF),SURF=1,2),IM=1,MM)
C
C  CALCULATE RMH, BEH, AND BETAH ARRAYS
C
  300 IF(BLDAT.GT.0) WRITE(6,1120)
      DO 320 SURF=1,2
      CALL SPLINT(MR,RMSP,NRSP,MH(1,SURF),IMS(SURF),RMH(1,SURF),AAA)
      CALL SPLINT(MR,BESP,NRSP,MH(1,SURF),IMS(SURF),BEH(1,SURF),AAA)
      IMSS = IMS(SURF)
      IF(IMSS.LT.1) GO TO 320
      DO 310 IHS = 1,IMSS
  310 BETAH(IHS,SURF) = ATAN(DTDMH(IHS,SURF)*RMH(IHS,SURF))*57.295779
      IF (BLDAT.GT.0) WRITE(6,1130) SURF,(MH(IM,SURF),RMH(IM,SURF),
     1   BEH(IM,SURF),BETAH(IM,SURF),DTDMH(IM,SURF),IM=1,IMSS)
  320 CONTINUE
      IF (BLDAT.LE.0) GO TO 340
      WRITE (6,1140)
      IT = ITMIN
  330 IF (IT.GT.ITMAX) GO TO 340
      TH = FLOAT(IT)*HT
      WRITE (6,1010) IT,TH
      IT = IT+1
      GO TO 330
  340 IF(NIP.LE.2500) GO TO 350
      WRITE(6,1150)
      STOP
```

```
 350 WRITE (6,1000)
     RETURN
1000 FORMAT (1H1)
1010 FORMAT (4X,I4,G16.5)
1020 FORMAT(60HLINPUT WEIGHT FLOW (WTFL) IS TOO LARGE AT BLADE LEADING
    1EDGE)
1025 FORMAT(40HL REDUCED WEIGHT FLOW IS STILL TOO LARGE)
1030 FORMAT (1H1/24X,10HFREESTREAM,8X,13HMAXIMUM VALUE,
    17X,8HCRITICAL,30X,14HBETA CORRECTED/25X,8HVELOCITY,10X,9HFOR RHO*W
    2,10X,8HVELOCITY,31X,11HTO BOUNDARY/1X,17HLEADING EDGE  B-G,3G18.5,
    312X,12HBOUNDARY A-H,G18.5/1X,17HTRAILING EDGE C-F,3G18.5,12X,
    412HBOUNDARY D-E,G18.5/86X,30H(BASED ON REDUCED WEIGHT FLOW))
1040 FORMAT(33HL    CALCULATED PROGRAM CONSTANTS//5X,5HPITCH,13X,
    1    2HHT,13X,3HHM1/1X,5G16.7)
1050 FORMAT (/5X,5HITMIN,10X,5HITMAX/4X,I5,10X,I5//5X,6HLAMBDA,12X,
    1    29HLAMBDA AT REDUCED WEIGHT FLOW/1X,G16.7,12X,G16.7/
    2    38HL     NUMBER OF INTERIOR MESH POINTS = ,I5)
1060 FORMAT(28HL    SURFACE BOUNDARY VALUES//5X,7HSURFACE,7X,2HBV
    1/(5X,I4,4X,F10.5))
1070 FORMAT (1H1,6X,62HBLADE DATA AT INTERSECTIONS OF VERTICAL MESH LIN
    1ES WITH BLADES)
1080 FORMAT (1HL,22X,15HBLADE SURFACE 1,30X,15HBLADE SURFACE 2/7X,
    1    1HM,14X,2HTV,11X,5HDTDMV,11X,4HCURV,12X,2HTV,11X,5HDTDMV,11X,
    2    4HCURV/(7G15.5))
1090 FORMAT (1H1,13X,44HSTREAM SHEET COORDINATES AND THICKNESS TABLE /
    1    2X,2HIM,7X,1HM,14X,1HR,13X,3HSAL,13X,1HB,12X,5HDB/DM/(1X,I3,
    2    5G15.5))
1100 FORMAT(34HLONE OF THE MH ARRAYS IS TOO LARGE/7H IT HAS,I5, 8H  POI
    1NTS)
1110 FORMAT (4H1 IM,9X,8HIV ARRAY,25X,9HITV ARRAY/38X,5HBLADE/37X,7HSUR
    1FACE,3X,1H1,5X,1H2/39X,3HNO./(1X,I3,5X,I10,25X,2(I4,2X)))
1120 FORMAT (67H1M COORDINATES OF INTERSECTIONS OF HORIZONTAL MESH LINE
    1S WITH BLADE)
1130 FORMAT (25HLMH ARRAY - BLADE SURFACE,I2//15X,2HMH,19X,3HRMH,19X,
    1    3HBEH,18X,5HBETAH,17X,5HDTDMH/(5G22.4))
1140 FORMAT (43H1THETA COORDINATES OF HORIZONTAL MESH LINES//6X,2HIT,
    15X,5HTHETA)
1150 FORMAT(48HLTHE NUMBER OF INTERIOR MESH POINTS EXCEEDS 2500)
     END



      SUBROUTINE COEF
C
C  COEF CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANTS, K,
C  AT ALL UNKNOWN MESH POINTS FOR THE ENTIRE REGION
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,REDFAC,
     1    DENTOL,MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2    BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1    HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2    TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3    TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4    BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5    SAL(100),AAA(100)
```

```
      COMMON /HRBAAK/H(4),R(4),B(4),KAK(4),KA(4),RZ,BZ,IH(4)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
C  INITIALIZE ARRAYS
      ITER = ITER+1
      IH(1) = 1
      IH(2) = 0
C  INCOMPRESSIBLE CASE
      IF(GAM.NE.1.5.OR.AR.NE.1000..OR.TIP.NE.1.E6) GO TO 20
      IEND = 1
      GO TO 40
C  ADJUSTMENT OF PRINTING CONTROL VARIABLES
   20 IF(ITER.NE.1.AND.ITER.NE.2) GO TO 30
      AANDK = AANDK-1
      ERSOR = ERSOR-1
      STRFN = STRFN-1
      SLCRD = SLCRD-1
      INTVL = INTVL-1
      SURVL = SURVL-1
   30 IF(IEND.NE.0) GO TO 40
      AANDK = AANDK+2
      ERSOR = ERSOR+2
      STRFN = STRFN+2
      SLCRD = SLCRD+2
      INTVL = INTVL+2
      SURVL = SURVL+2
C
C  FIRST VERTICAL MESH LINE
C
   40 DO 50 IP=1,NBBI
      A(IP,1) = 0.
      A(IP,2) = 0.
      A(IP,3) = 0.
      A(IP,4) = 1.
   50 K(IP) = HMI*TBI/PITCH/((RM(1)+RM(2))/2.)
C
C   UPSTREAM OF BLADE, EXCEPT FOR FIRST VERTICAL MESH LINE
C
      IF(2.GT.MBIM1) GO TO 70
      DO 60 IM=2,MBIM1
   60 CALL COEFP(IM)
C
C   BETWEEN BLADES
C
   70 DO 80 IM=MBI,MBO
   80 CALL COEFBB(IM)
C
C  DOWNSTREAM OF BLADES EXCEPT FOR FINAL MESH LINE
C
  150 IF(MBOP1.GT.MMM1) GO TO 170
      DO 160 IM=MBOP1,MMM1
  160 CALL COEFP(IM)
C
C  FINAL VERTICAL MESH LINE
C
  170 IVMM = IV(MM)
      DO 180 IP=IVMM,NIP
      A(IP,1) = 0.
      A(IP,2) = 0.
```

```
      A(IP,3) = 1.
      A(IP,4) = 0.
  180 K(IP) = -HM1*TBO/PITCH/RM(MM)
C
C  TAKE CARE OF POINTS ADJACENT TO B, AND CASES WHEN POINTS J,C,E, OR F
C  ARE GRID POINTS
C
C  POINT B
      IP = IV(MBIM1)
      A(IP,4) = 0.
C  POINT C
      IF(ITV(MBO,1)-ITV(MBO,2)+NBBI.NE.2) RETURN
      IT = ITV(MBO,1)-1
      IP = IPF(MBOP1,IT)
      A(IP,3) = 0.
      RETURN
      END
```

```
      SUBROUTINE COEFBB(IM)
C
C  COEFBB CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANTS, K
C  ALONG ALL VERTICAL MESH LINES WHICH INTERSECT BLADES
C
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,REDFAC,
     1   DENTOL,MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2   TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3   TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4   BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5   SAL(100),AAA(100)
      COMMON /HRBAAK/H(4),R(4),B(4),KAK(4),KA(4),RZ,BZ,IH(4)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      IF(ITV(IM,1).GT.ITV(IM,2)) RETURN
      ITVU = ITV(IM,1)
      IT   = ITVU - 1
      ITVL = ITV(IM,2)
      IPU = IPF(IM,ITVU)
      IPL = IPU+ITVL-ITVU
      DO 90 IP=IPU,IPL
      IT = IT+1
      CALL HRB(IM,IT,IP)
      DO 10 I=1,4
      KAK(I) = 0.
   10 KA(I) = 0
C  FIX HRB VALUES FOR CASES WHERE MESH LINES INTERSECT BLADES
   60 IF(IT.EQ.ITV(IM,1)) CALL BDRY12(1,IM,IT)
      IF(IT.EQ.ITV(IM,2)) CALL BDRY12(2,IM,IT)
      ITVM1 = ITV(IM-1,1)
      ITVP1 = ITV(IM+1,1)
      IF(IT.LT.ITVM1) CALL BDRY34(3,IM,1)
```

```
      IF(IT.LT.ITVP1) CALL BDRY34(4,IM,1)
      IF(IT.GT.ITV(IM-1,2)) CALL BDRY34(3,IM,2)
      IF(IT.GT.ITV(IM+1,2)) CALL BDRY34(4,IM,2)
   70 IF(IM.EQ.MBO.AND.LOWER.EQ.2) GO TO 80
C  COMPUTE A AND K COEFFICIENTS
   80 CALL AAK(IM,IP)
      DO 90 I=1,4
      K(IP) = K(IP)+KAK(I)*A(IP,I)
   90 IF(KA(I).EQ.1) A(IP,I) = 0.
      RETURN
C
C  COEFP CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANTS, K,
C  ALONG ALL VERTICAL MESH LINES WHICH DO NOT INTERSECT BLADES
C
      ENTRY COEFP(IM)
      ITVU = ITV(IM,1)
      IT = ITVU-1
      ITVL = ITV(IM,2)
      IPL = IV(IM+1)-1
      IPU = IV(IM)
      DO 100 IP=IPU,IPL
      IT = IT+1
      CALL HRB(IM,IT,IP)
      IF (IT.EQ.ITVU) R(1) = RHO(IPL)
      IF (IT.EQ.ITVL) R(2) = RHO(IPU)
  100 CALL AAK(IM,IP)
      K(IPL) = K(IPL)+A(IPL,2)
      K(IPU) = K(IPU)-A(IPU,1)
      RETURN
      END




      SUBROUTINE HRB(IM,IT,IP)
C
C  HRB CALCULATES MESH SPACING, H, DENSITIES, RZ AND R, AT GIVEN AND
C  ADJACENT POINTS, AND STREAM SHEET THICKNESSES, BZ AND B, AT GIVEN
C  AND ADJACENT POINTS
C
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2   TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3   TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4   BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5   SAL(100),AAA(100)
      COMMON /HRBAAK/H(4),R(4),B(4),KAK(4),KA(4),RZ,BZ,IH(4)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      H(1)= HT*RM(IM)
      H(2)= HT*RM(IM)
      H(3)= MV(IM) - MV(IM-1)
      H(4) = MV(IM+1)-MV(IM)
      RZ = RHO(IP)
      IP3 = IPF(IM-1,IT)
      IP4 = IPF(IM+1,IT)
```

```
      R(1) = RHO(IP-1)
      R(2) = RHO(IP+1)
      R(3) = RHO(IP3)
      R(4) = RHO(IP4)
      BZ= BE(IM)
      B(3)= BE(IM-1)
      B(4)= BE(IM+1)
      RETURN
      END
```

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

```
      SUBROUTINE BORY12(I,IM,IT)
C
C  BDRY12 CORRECTS VALUES COMPUTED BY HRB WHEN A VERTICAL MESH LINE
C  INTERSECTS A BLADE
C
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2   TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3   TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4   BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5   SAL(100),AAA(100)
      COMMON /RHOS/RHOHB(100,2),RHOVB(100,2)
```

```
      COMMON /HRBAAK/H(4),R(4),B(4),KAK(4),KA(4),RZ,BZ,IH(4)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      H(I) = ABS(FLOAT(IT)*HT-TV(IM,I))*RM(IM)
      R(I)= RHOVB(IM,I)
      KAK(I) = BV(I)
      KA(I)=1
      RETURN
      END
```

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

```
      SUBROUTINE SOR
C
C  SOR SOLVES THE SET OF SIMULTANEOUS EQUATIONS FOR THE STREAM FUNCTION
C  USING THE METHOD OF SUCCESSIVE OVER-RELAXATION
C
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,REDFAC,
     1   DENTOL,MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2   TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3   TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4   BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
```

```
     5    SAL(100),AAA(100)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1    UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      AATEMP = AANDK
      IF (ORF.GE.2.) ORF=0.
      IF(ORF.GT.1.) GO TO 50
      CRF = 1.
      CRFOPT= 2.
   40 CRFTEM=ORFOPT
      LMAX = 0.
   50 IF(AATEMP.GT.0) WRITE(6,1010)
      ERROR = 0.
C
C  SOLVE MATRIX EQUATION BY SOR, OR CALCULATE OPTIMUM OVERRELAXATION
C  FACTOR
C
      IP = 0
      CO 120 IM=1,MM
      IPU = IV(IM)
      IPL = IV(IM+1)-1
      IT = ITV(IM,1)
      IF(AATEMP.GT.0) WRITE (6,1020) IM,IT
      CO 120 IP=IPU,IPL
      IP1 = IP-1
      IP2 = IP+1
C  CORRECT IP1 AND IP2 ALONG PERIODIC BOUNDARIES
      IF(IM.GE.MBI.AND.IM.LE.MBO) GO TO 60
      IF(IT.EQ.ITV(IM,1)) IP1 = IP1+NBBI
      IF(IT.EQ.ITV(IM,2)) IP2 = IP2-NBBI
   60 IT3 = IT
      IT4 = IT
  100 IP3 = IPF(IM-1,IT3)
      IP4 = IPF(IM+1,IT4)
      IF(CRF.GT.1.) GO TO 110
C  CALCULATE NEW ESTIMATE FOR LMAX
      UNEW = A(IP,1)*U(IP1)+A(IP,2)*U(IP2)+A(IP,3)*U(IP3)+A(IP,4)*U(IP4)
      IF (UNEW.LT.1.E-25) U(IP) = 0.
      IF (U(IP).FQ.0.) GO TO 115
      RATIO = UNEW/U(IP)
      LMAX= AMAX1(RATIO,LMAX)
      U(IP) = UNEW
      GO TO 115
C  CALCULATE NEW ESTIMATE FOR STREAM FUNCTION BY SOR
  110 CHANGE = ORF*(K(IP)-U(IP)+A(IP,1)*U(IP1)+A(IP,2)*U(IP2)+A(IP,3)*
     1    U(IP3)+A(IP,4)*U(IP4))
      ERROR= AMAX1(ERROR,ABS(CHANGE))
      U(IP) = U(IP)+CHANGE
  115 IF(AATEMP.LE.0) GO TO 120
      WRITE (6,1030) IT,IP,IP1,IP2,IP3,IP4,(A(IP,I),I=1,4),K(IP)
  120 IT = IT+1
      AATEMP = 0
      IF(CRF.GT.1.) GO TO 130
      CRFOPT= 2./(1.+SQRT(ABS(1.-LMAX)))
      WRITE(6,1000) ORFOPT
      IF(ORFTEM-ORFOPT.GT..00001.OR.ORFOPT.GT.1.999) GO TO 40
      WRITE (6,1070)
      CRF = ORFOPT
      GO TO 50
```

```
  130 IF(ERSOR.GT.0) WRITE(6,1040) ERROR
      IF(ERROR.GT..000001) GO TO 50
      IF(STRFN.LE.0) RETURN
C
C  PRINT STREAM FUNCTION VALUES FOR THIS ITERATION
C
      WRITE (6,1050)
      DO 140 IM=1,MM
      IPU = IV(IM)
      IPL = IV(IM+1)-1
      ITVU = ITV(IM,1)
      WRITE (6,1020) IM,ITVU
  140 WRITE (6,1060) (U(IP),IP=IPU,IPL)
      RETURN
 1000 FORMAT(24H ESTIMATED OPTIMUM ORF =,F9.6)
 1010 FORMAT (82H1  IT    IP     IP1    IP2    IP3    IP4     A(1)       A(2)
     1     A(3)       A(4)        K)
 1020 FORMAT(5HKIM =,I4,6X,6HITI1 = ,I4)
 1030 FORMAT(1X,I4,5I6,5F10.5)
 1040 FORMAT(8H ERROR =,F11.8)
 1050 FORMAT(1H1,10X,46HSTREAM FUNCTION VALUES FOR REDUCED WEIGHT FLOW)
 1060 FORMAT (2X,10F13.8)
 1070 FORMAT (1H1)
      END




      SUBROUTINE SLAX
C
C  SLAX CALLS SUBROUTINES TO CALCULATE RHO*W-SUB-M THROUGHOUT THE REGION
C  AND ON THE BLADE SURFACES, AND TO CALCULATE AND PLOT THE
C  STREAMLINE LOCATIONS
C
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,REDFAC,
     1   DENTOL,MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2   TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3   TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4   BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5   SAL(100),AAA(100)
      COMMON /SLA/TSL(600),UINT(6)
      DIMENSION MSL(100),KKK(14),P(4)
      COMMON /SURVEL/ WTB(100,2),WMB(100,2),XDOWN(400),YACROS(400)
      DIMENSION W(2500),BETA(2500),DUDT(2500),DUDTT(2500),AAP(2500),
     1   BBP(2500)
      EQUIVALENCE (A,W),(A(1,2),BETA),(A(1,3),DUDT),(A(1,4),DUDTT),
     1   (K,AAP),(RHO,BBP)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      DATA (KKK(J),J=4,14,2)/6*1H*/
C
C  CALL SLAVP AND SLAVBB THROUGHOUT THE REGION
C
```

```
      ITVU= ITV(1,1)
      ITVL= ITV(1,2)
      CO 10 IM=1,MBIM1
   10 CALL SLAVP(IM,ITVU,ITVL)
      CO 20 IM=MBI,MBO
      I= 0
   20 CALL SLAVBB(IM)
   90 ITVU = ITV(MBOP1,1)
      ITVL = ITV(MBOP1,2)
      CO 100 IM=MBOP1,MM
  100 CALL SLAVP(IM,ITVU,ITVL)
C
C  PLOT STREAMLINES
C
      IF (SLCRD.LE.0) RETURN
      CO 110 IM=1,MM
  110 MSL(IM) = MV(IM)
      KKK(1) = 7
      KKK(2) = 6
      KKK(3) = MM
      P(1) = 1.
      P(3) = 0.
      P(4) = 0.
      WRITE(6,1000)
      CALL PLOTMY(MSL,TSL,KKK,P)
      WRITE(6,1010)
      RETURN
 1000 FORMAT (2HPT,50X,40HSTREAMLINE PLOTS FOR REDUCED WEIGHT FLOW)
 1010 FORMAT (2HPL,40X,70HSTREAMLINES ARE PLOTTED WITH THETA ACROSS THE
     1PAGE AND M DOWN THE PAGE)
      END




      SUBROUTINE SLAV
C
C  SLAV CALCULATES RHO*W-SUB-M THROUGHOUT THE REGION AND ON THE BLADE
C  SURFACES, AND CALCULATES THE STREAMLINE LOCATIONS
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,REDFAC,
     1   DENTOL,MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1    HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2    TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3    TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4    BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5    SAL(100),AAA(100)
      COMMON /SLA/TSL(600),UINT(6)
      DIMENSION TSP(50),USP(50),TINT(6),DDT(50)
      COMMON /SURVEL/ WTB(100,2),WMB(100,2),XDOWN(400),YACROS(400)
      DIMENSION W(2500),BETA(2500),DUDT(2500),DUDTT(2500),AAP(2500),
     1    BBP(2500)
      EQUIVALENCE (A,W),(A(1,2),BETA),(A(1,3),DUDT),(A(1,4),DUDTT),
     1    (K,AAP),(RHO,BBP)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
```

```
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
C
C  SLAVP CALCULATES ALONG VERTICAL MESH LINES WHICH DO NOT
C  INTERSECT BLADES
C
      ENTRY SLAVP(IM,ITVU,ITVL)
      LOC= 0
      NSP= ITVL-ITVU+2
      IP = IV(IM)-1
      DO 10 IT=1,NSP
      IP = IP+1
      TSP(IT) = FLOAT(IT+ITVU-1)*HT
   10 USP(IT)= U(IP)
      USP(NSP) = USP(1)+1.
      IP = IV(IM)
      INTU = INT(U(IP)*5.)
      IF (U(IP).GT.0.) INTU=INTU+1
      DO 20 J=1,5
      UINT(J) = FLOAT(INTU)/5.
   20 INTU = INTU+1
      UINT(6) = UINT(1)
      GO TO 100
C
C  SLAVBB CALCULATES ALONG VERTICAL MESHLINES WHICH INTERSECT BLADES
C
      ENTRY SLAVBB(IM)
      LOC= 1
      ITVUP1 = ITV(IM,1)
      ITVLM1 = ITV(IM,2)
      ITVU = ITVUP1-1
      ITVL = ITVLM1+1
      NSP = ITVL-ITVU+1
      TSP(1) = TV(IM,1)
      TSP(NSP) = TV(IM,2)
      USP(1) = BV(1)
      USP(NSP) = BV(2)
      IP = IV(IM)-1
      NSPM1 = NSP-1
      IF(2.GT.NSPM1) GO TO 70
      DO 60 IT=2,NSPM1
      IP = IP+1
      TSP(IT) = FLOAT(IT+ITVU-1)*HT
   60 USP(IT) = U(IP)
   70 DO 80 I=1,6
   80 UINT(I) = FLOAT(I-1)/5.
C
C  FOR BOTH SLAVP AND SLAVBB, CALCULATE RHO*W-SUB-M IN THE REGION, AND
C  RHO*W AT VERTICAL MESH LINE INTERSECTIONS ON THE BLADE SURFACES
C
  100 CONTINUE
      CALL SPLINE(TSP,USP,NSP,DDT,AAA)
      IT = LOC
      IPU = IV(IM)
      IPL = IV(IM+1)-1
      DO 110 IP=IPU,IPL
      IT = IT+1
      DUDT(IP) = DDT(IT)
  110 DUDTT(IP) = AAA(IT)
  120 IF (LOC.EQ.0) GO TO 130
```

```
      WMB(IM,1) =  DDT(  1)*WTFL/BE(IM)/RM(IM)
      WMB(IM,2) =  DDT(NSP)*WTFL/BE(IM)/RM(IM)
      RMDTU2 = (RM(IM)*DTDMV(IM,1))**2
      RMDTL2 = (RM(IM)*DTDMV(IM,2))**2
      IF (RMDTU2.GT.10000.) WMB(IM,1) = 0.
      IF (RMDTL2.GT.10000.) WMB(IM,2) = 0.
      WMB(IM,1) = ABS(WMB(IM,1))*SQRT(1.+RMDTU2)
      WMB(IM,2) = ABS(WMB(IM,2))*SQRT(1.+RMDTL2)
  130 IF (SLCRD.LE.0) RETURN
      NI = 6
      CALL SPLINT(USP,TSP,NSP,UINT,NI,TINT,AAA)
      DO 140 J=1,6
      L= (J-1)*MM+IM
  140 TSL(L)= TINT(J)
      IF (IM.EQ.1) WRITE(6,1000)
      WRITE(6,1010) MV(IM),(UINT(J),TINT(J),J=1,6)
      RETURN
 1000 FORMAT(1H130X,46HSTREAMLINE COORDINATES FOR REDUCED WEIGHT FLOW/
     1   1HL,4X,12HM COORDINATE,3(7X,10HSTREAM FN.,10X,5HTHETA,4X)//)
 1010 FORMAT(1X,7G18.7/(19X,6G18.7))
      END




      SUBROUTINE TANG
C
C  TANG CALCULATES RHO*W-SUB-THETA AND THEN RHO*W THROUGHOUT THE REGION
C  AND ON THE BLADE SURFACES, AND CALCULATES THE VELOCITY ANGLE, BETA,
C  THROUGHOUT THE REGION
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,REDFAC,
     1   DENTOL,MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2   TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3   TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4   BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5   SAL(100),AAA(100)
      COMMON /SURVEL/ WTB(100,2),WMB(100,2),XDOWN(400),YACROS(400)
      COMMON /RHOS/RHOHB(100,2),RHOVB(100,2)
      DIMENSION W(2500),BETA(2500),DUDT(2500),DUDTT(2500),AAP(2500),
     1   BBP(2500)
      EQUIVALENCE (A,W),(A(1,2),BETA),(A(1,3),DUDT),(A(1,4),DUDTT),
     1   (K,AAP),(RHO,BBP)
      DIMENSION SPM(100),USP(100),DDT(100),DUDM(100),DUDMM(100),
     1   DUDTM(100)
      DIMENSION DWDM(100),WIP(100)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      EXTERNAL BL1,BL2
C
C  PERFORM CALCULATIONS ALONG ONE HORIZONTAL LINE AT A TIME
C
```

```
      IT = ITMIN
   10 IF (IT.GT.ITMAX) RETURN
      S1 = 0
C
C  ON THE GIVEN HORIZONTAL MESH LINE, FIND A FIRST POINT IN THE REGION
C
      IF(IT.GE.0.AND.IT.LT.NBBI) GO TO 60
      IM = MBIM1
   20 IM= IM+1
      IF(IM.GT.MBO) GO TO 200
      SURF = 1
      IF(IT.GE.ITV(IM,1).AND.IT.LT.ITV(IM-1,1)) GO TO 70
      IF(IM.EQ.MBOP1.AND.IT.EQ.ITV(MBO,1)-1.AND.ITV(MBO,1)-ITV(MBO,2)
     1   +NBBI.EQ.2) GO TO 70
      SURF = 2
      IF(IT.LE.ITV(IM,2).AND.IT.GT.ITV(IM-1,2)) GO TO 70
      GO TO 20
C
C  FIRST POINT IS ON BOUNDARY A-H
C
   60 IM1= 1
      IM = 1
      SPM(1) = MV(1)
      LSP (1) = U(IT+1)
      GO TO 90
C
C  FIRST POINT IS ON A BLADE SURFACE
C
   70 S1 = SURF
      IM1 = IM-1
      IM2 = IM
      TH = FLOAT(IT)*HT
      MVIM1 = MV(IM1)
      IF (IM.EQ.MBIP1) MVIM1 = MVIM1+(MV(IM2)-MVIM1)/1000.
      LER(2) = 5
C     BLCD (VIA ROOT) CALL NO. 5
      IF(S1.EQ.1.AND.IM1.NE.MBO)CALL ROOT(MVIM1   ,MV(IM2),TH,BL1,DTLR,
     1   ANS,AAA)
      LER(2) = 6
C     BLCD (VIA ROOT) CALL NO. 6
      IF(S1.EQ.2)CALL ROOT(MVIM1   ,MV(IM2),TH,BL2,DTLR,ANS,AAA)
      IF(S1.EQ.1.AND.IM1.EQ.MBO) ANS = MV(MBO)
      SPM(IM1) = ANS
      LSP(IM1)= BV(S1)
C
C  MOVE ALONG HORIZONTAL MESH LINE UNTIL MESH LINE INTERSECTS BOUNDARY
C
   90 IF(IM.LT.MBI.OR.IM.GT.MBO) GO TO 120
      SURF = 1
      IF(IT.LT.ITV(IM,SURF).AND.IT.GE.ITV(IM-1,SURF)) GO TO 140
      SURF = 2
      IF(IT.GT.ITV(IM,SURF).AND.IT.LE.ITV(IM-1,SURF)) GO TO 140
  120 SPM(IM) = MV(IM)
      IP = IPF(IM,IT)
      LSP(IM) = U(IP)
      IF (IM.EQ.MM) GO TO 130
      IM= IM+1
      GO TO 90
```

```
C
C    FINAL POINT IS ON BOUNDARY D-E
C
  130 IMT = MM
      GO TO 150
C
C   FINAL POINT IS ON A BLADE SURFACE
C
  140 ST = SURF
      IMT=IM
      IMTM1= IMT-1
      TH = FLOAT(IT)*HT
      MVIM1 = MV(IMTM1)
      IF (IM.EQ.MBIP1) MVIM1 = MVIM1+(MV(IMT)-MVIM1)/1000.
      LER(2) = 7
C     BLCD (VIA ROOT) CALL NO. 7
      IF(ST.EQ.1.AND.IMT.NE.MBI)CALL ROOT(MVIM1     ,MV(IMT),TH,BL1,
     1   DTLR,ANS,AAA)
      LER(2) = 8
C     BLCD (VIA ROOT) CALL NO. 8
      IF(ST.EQ.2)CALL ROOT(MVIM1     ,MV(IMT),TH,BL2,DTLR,ANS,AAA)
      IF(ST.EQ.1.AND.IMT.EQ.MBI) ANS = MV(MBI)
      SPM(IMT) = ANS
      USP(IMT)= BV(ST)
  150 NSP= IMT-IM1+1
      CALL SPLINE(SPM(IM1),USP(IM1),NSP,DUDM(IM1),DUDMM(IM1))
C
C   CALCULATE RHO*W ON THE BLADE SURFACES
C
      FIRST=1
      LAST= MM
      IF(IM1.EQ.1) GO TO 160
      FIRST = IM2
      CALL SEARCH (SPM(IM1),S1,IHS)
      ANS =-DUDM(IM1)*WTFL/BEH(IHS,S1)
      WTB(IHS,S1) = ABS(ANS)*SQRT(1.+1./(RMH(IHS,S1)*DTDMH(IHS,S1))**2)
      DDT(IM1) =-DUDM(IM1)/DTDMH(IHS,S1)
      WIP(IM1) = WTB(IHS,S1) / RHOHB(IHS,S1)
  160 IF(IMT.EQ.MM) GO TO 170
      LAST = IMTM1
      CALL SEARCH (SPM(IMT),ST,IHS)
      ANS =-DUDM(IMT)*WTFL/BEH(IHS,ST)
      WTB(IHS,ST) = ABS(ANS)*SQRT(1.+1./(RMH(IHS,ST)*DTDMH(IHS,ST))**2)
      DDT(IMT) =-DUDM(IMT)/DTDMH(IHS,ST)
      WIP(IMT) = WTB(IHS,ST) / RHOHB(IHS,ST)
C
C   CALCULATE RHO*W-SUB-THETA AND THEN RHO*W AND BETA IN THE REGION
C
  170 IF(FIRST.GT.LAST) GO TO 190
      DO 180 I=FIRST,LAST
      IP = IPF(I,IT)
      DDT(I) = DUDT(IP)
      RWM = DDT(I)/RM(I)
      RWT =-DUDM(I)
      W(IP) = SQRT(RWT**2+RWM**2)/BE(I)*WTFL
      TWLMR = 2.*OMEGA*LAMBDA-(OMEGA*RM(I  ))**2
      LER(1) = 5
C     DENSTY CALL NO. 5
      CALL DENSTY(W(IP),RHO(IP),ANS,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP)
      W(IP) = ANS
```

```
      WIP(I) = W(IP)
      BETA(IP) = ATAN2(RWT,RWM)*57.295779
  180 CONTINUE
      IF(IEND.LT.0) GO TO 190
      CALL SPLINE (SPM(IM1),CDT(IM1),NSP,DUDTM(IM1),AAA(IM1))
      CALL SPLINE (SPM(IM1),WIP(IM1),NSP,DWDM(IM1),AAA(IM1))
      DO 185 I=FIRST,LAST
      IP = IPF(I,IT)
      SBETA = SIN(BETA(IP)/57.295779)
      CBETA = SQRT(1.-SBETA**2)
      AAP(IP) = SBETA**2*(2.*DUDTM(I)/DUDM(I)-DUDT(IP)/DUDM(I)**2*
     1   DUDMM(I)-DUDTT(IP)/DUDT(IP))+SAL(I)*SBETA/CBETA*(1.+CBETA**2)
      BBP(IP) = RM(I)/CBETA*(2.*ACTOMG*SAL(I)+SBETA*DWDM(I)/REDFAC)
  185 CONTINUE
  190 CONTINUE
      IF(IMT.NE.MM) GO TO 20
  200 IT = IT+1
      GO TO 10
      END
```

```
      SUBROUTINE SEARCH (DIST,SURF,IS)
C
C   SEARCH LOCATES THE POSITION OF A GIVEN VALUE OF M IN THE MH ARRAY
C
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2   TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3   TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4   BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5   SAL(100),AAA(100)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      DO 10 I=1,100
      IF (ABS(MH(I,SURF)-DIST).GT.DMLR) GO TO 10
      IS = I
      RETURN
   10 CONTINUE
      WRITE (6,1000) DIST,SURF
      STOP
 1000 FORMAT (38HL SEARCH CANNCT FIND M IN THE MH ARRAY/7H DIST =,G14.6,
     110X,6HSURF =,G14.6)
      END
```

```
      SUBROUTINE VELOCY
C
C   VELOCY CALLS SUBROUTINES TO CALCULATE DENSITIES AND VELOCITIES
C   THROUGHOUT THE REGION AND ON THE BLADE SURFACES, AND IT PLOTS
C   THE SURFACE VELOCITIES
C
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,REDFAC,
     1   DENTOL,MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2   TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3   TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4   BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5   SAL(100),AAA(100)
      DIMENSION KKK(14)
      COMMON /SURVEL/ WTB(100,2),WMB(100,2),XDOWN(400),YACROS(400)
      DIMENSION W(2500),BETA(2500),DUDT(2500),DUDTT(2500),AAP(2500),
     1   BBP(2500)
      EQUIVALENCE (A,W),(A(1,2),BETA),(A(1,3),DUDT),(A(1,4),DUDTT),
     1   (K,AAP),(RHO,BBP)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,SI,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      DATA KKK(4)/1H*/,KKK(6)/1HO/,KKK(8)/1H+/,KKK(10)/1HX/
C
C   CALL VELP, VELBB, AND VELSUR THROUGHOUT THE REGION
C
      IF(INTVL.GT.0)CALL VELP(1,MBIM1)
      CALL VELBB(MBI,MBO)
   20 IF(INTVL.GT.0)CALL VELP(MBOP1,MM)
      CALL VELSUR
C
C   PREPARE INPUT ARRAYS FOR PLOT OF VELOCITIES
C
      IF(SURVL.LE.0) RETURN
      NP2 = 0
C    TANGENTIAL COMPONENTS
      DO 50 SURF=1,2
      NP1 = NP2
      IMSS = IMS(SURF)
      IF(IMSS.LT.1) GO TO 40
      DO 30 IHS=1,IMSS
      IF (ABS(DTDMH(IHS,SURF)*RMH(IHS,SURF)).LT..57735) GO TO 30
      NP1 = NP1+1
      YACROS(NP1) = WTB(IHS,SURF)
      XDOWN(NP1) = MH(IHS,SURF)
   30 CONTINUE
   40 KKK(2*SURF+1) = NP1-NP2
   50 NP2 = NP1
C    MERIDIONAL COMPONENTS
      DO 80 SURF=1,2
      NP1 = NP2
      DO 60 IM=MBIP1,MBOM1
      IF (ABS(DTDMV(IM,SURF)*RM(IM)).GT.1.7321) GO TO 60
      NP1 = NP1+1
      YACROS(NP1) = WMB(IM,SURF)
      XDOWN(NP1) = MV(IM)
   60 CONTINUE
```

```
   70 KKK(2*SURF+5) = NP1-NP2
   80 NP2 = NP1
C
C  PLOT VELOCITIES
C
      KKK(1) = 1
      KKK(2) = 4
      P = 5.
      WRITE(6,1000)
      CALL PLOTMY(XDOWN,YACROS,KKK,P)
      WRITE(6,1010)
      RETURN
 1000 FORMAT(2HPT,50X,48HBLADE SURFACE VELOCITIES FOR REDUCED WEIGHT FLO
     1W)
 1010 FORMAT (2HPL,37X,63HVELOCITY(W) VS. MERIDIONAL STREAMLINE DISTANCE
     1(M) DOWN THE PAGE  /2HPL/
     2   2HPL,50X,50H+ - BLADE SURFACE 1, BASED ON MERIDIONAL COMPONENT/
     3   2HPL,50X,50H* - BLADE SURFACE 1, BASED ON TANGENTIAL COMPONENT/
     4   2HPL,50X,50HX - BLADE SURFACE 2, BASED ON MERIDIONAL COMPONENT/
     5   2HPL,50X,50HO - BLADE SURFACE 2, BASED ON TANGENTIAL COMPONENT)
      END




      SUBROUTINE VEL
C
C  VEL CALCULATES DENSITIES AND VELOCITIES FROM THE PRODUCT OF
C  DENSITY TIMES VELOCITY
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,REDFAC,
     1   DENTOL,MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2   TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3   TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4   BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5   SAL(100),AAA(100)
      COMMON /RHOS/RHOHB(100,2),RHOVB(100,2)
      DIMENSION WWCRM(100,2),WWCRT(100,2),SURFL(100,2)
      COMMON /SURVEL/ WTB(100,2),WMB(100,2),XDOWN(400),YACROS(400)
      DIMENSION W(2500),BETA(2500),DUDT(2500),DUDTT(2500),AAP(2500),
     1   BBP(2500)
      EQUIVALENCE (A,W),(A(1,2),BETA),(A(1,3),DUDT),(A(1,4),DUDTT),
     1   (K,AAP),(RHO,BBP)
C
C  VELP CALCULATES ALONG VERTICAL MESH LINES WHICH DO NOT
C  INTERSECT BLADES
C
      ENTRY VELP(FIRST,LAST)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLF,MR,MSL,MSP,MV,MVIM1
      IF(FIRST.GT.LAST) RETURN
      IF (FIRST.EQ.1) WRITE (6,1000)
      DO 20 IM=FIRST,LAST
```

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

```
      TWLMR = 2.*OMEGA*LAMBDA-(OMEGA*RMH(IHS,SURF))**2
      WCR = SQRT(TGROG*TIP*(1.-TWLMR/CPTIP))
      RHOB = RHOHB(IHS,SURF)
      LER(1)=8
C     DENSTY CALL NO. 8
      CALL DENSTY(WTB(IHS,SURF),RHOHB(IHS,SURF),ANS,TWLMR,CPTIP,
     1   EXPON,RHOIP,GAM,AR,TIP)
      WTB(IHS,SURF) = ANS
      WWCRT(IHS,SURF) = WTB(IHS,SURF)/WCR
   80 RELER = AMAX1(RELER,ABS((RHOB-RHOHB(IHS,SURF))/RHOHB(IHS,SURF)))
   90 CONTINUE
      IF(RELER.LT.DENTOL) IEND = IEND+1
      WRITE(6,1080) ITER,RELER
C
C   WRITE ALL BLADE SURFACE VELOCITIES
C
      IF (SURVL.LE.0) RETURN
      WRITE(6,1020)
      WRITE(6,1040) (MV(IM),WMB(IM,1),BETAV(IM,1),SURFL(IM,1),
     1   WWCRM(IM,1),WMB(IM,2),BETAV(IM,2),SURFL(IM,2),WWCRM(IM,2),
     2   IM=MBI,MBO)
      WRITE(6,1050)
      DO 100 SURF=1,2
      IMSS = IMS(SURF)
      IF(IMSS.LT.1) GO TO 100
      WRITE(6,1060) SURF
      WRITE(6,1070) (MH(IHS,SURF),WTB(IHS,SURF),BETAH(IHS,SURF),
     1   WWCRT(IHS,SURF), IHS=1,IMSS)
  100 CONTINUE
      RETURN
 1000 FORMAT(1H1/40X,34HVELOCITIES AT INTERIOR MESH POINTS/45X,
     1   23HFOR REDUCED WEIGHT FLOW)
 1010 FORMAT(1HL,3HIM=,I3,5(24H    VELOCITY    ANGLE(DEG))/
     1(5X,5(G15.4,F9.2)))
 1020 FORMAT(1H1/16X,1H*,18X,71HSURFACE VELOCITIES BASED ON MERIDIONAL C
     1CMPONENTS - REDUCED WEIGHT FLOW,18X,1H*/16X,1H*,53X,1H*53X,1H*/
     2   16X,1H*,19X,15HBLADE SURFACE 1,19X,1H*,20X,15HBLADE SURFACE 2,
     3   18X,1H*/7X,1HM,8X,1H*,2(3X,8HVELOCITY,3X,23HANGLE(DEG) SURF. LE
     4NGTH,5X,5HW/WCR,6X,1H*))
 1040 FORMAT((1H ,G13.4,3H  *,2(G12.4,F9.2,2G15.4,3H  *)))
 1050 FORMAT(1H1/3X,49HSURFACE VELOCITIES BASED ON TANGENTIAL COMPONENTS
     1   /18X,19HREDUCED WEIGHT FLOW)
 1060 FORMAT(//22X,15HBLADE SURFACE  ,I1/7X,1HM,10X,8HVELOCITY,3X,10HANG
     1LE(DEG),3X,5HW/WCR)
 1070 FORMAT(1H ,2G13.4,F9.2,G15.4)
 1080 FORMAT(14HLITERATION NO.,I3,3X,36HMAXIMUM RELATIVE CHANGE IN DENSI
     1TY =,G11.4)
      END
```

```
      SUBROUTINE TVELCY
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,REDFAC,
     1   DENTOL,MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2   TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3   TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4   BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5   SAL(100),AAA(100)
      COMMON /WWCRM/WWCRM(100,2),LABEL(100)
      COMMON /SURVEL/ WTB(100,2),WMB(100,2),XDOWN(400),YACROS(400)
      DIMENSION W(2500),BETA(2500),DUDT(2500),DUDTT(2500),AAP(2500),
     1   BBP(2500)
      EQUIVALENCE (A,W),(A(1,2),BETA),(A(1,3),DUDT),(A(1,4),DUDTT),
     1   (K,AAP),(RHO,BBP)
      DIMENSION KKK(14)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,SURFBV,
     1FIRST,UPPER,UPPRBV,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      LAMBDA = ACTLAM
      IF(INTVL.GT.0) WRITE(6,1000)
      IF(1.GT.MBIM1) GO TO 20
      DO 10 IM=1,MBIM1
   10 CALL VELGRA(IM)
   20 DO 30 IM=MBIP1,MBOM1
   30 CALL VELGRB(IM)
      IF(MBOP1.GT.MM) GO TO 50
      DO 40 IM=MBOP1,MM
   40 CALL VELGRA(IM)
      IF(SURVL.LE.0) RETURN
   50 WRITE(6,1010)
      WRITE(6,1020)(MV(IM),WMB(IM,1),WWCRM(IM,1),LABEL(IM),WMB(IM,2),
     1   WWCRM(IM,2),LABEL(IM),IM=MBIP1,MBOM1)
C    PREPARE ARRAYS FOR PLOT OF VELOCITIES
      DO 60 IM = MBIP1,MBOM1
      I = IM - MBI
      I2 = I + MBOM1 - MBI
      XDOWN(I) = MV(IM)
      YACROS(I) = WMB(IM,1)
   60 YACROS(I2) = WMB(IM,2)
      KKK(1) = 0
      KKK(2) = 2
      KKK(3) = MBOM1 - MBI
      P = 1.
C    PLOT VELOCITIES
      WRITE (6,1030)
      CALL PLOTMY (XDOWN,YACROS,KKK,P)
      WRITE (6,1040)
      RETURN
 1000 FORMAT(1H1/40X,34HVELOCITIES AT INTERIOR MESH POINTS/44X,
     1    27H(BASED ON FULL WEIGHT FLOW))
 1010 FORMAT(1H1/16X,1H*,13X,68HSURFACE VELOCITIES BASED ON MERIDIONAL C
     1OMPONENTS - FULL WEIGHT FLOW,30X,1H*/16X,1H*,55X,1H*,55X,1H*/16X,
     21H*,20X,15HBLADE SURFACE 1,20X,1H*,20X,15HBLADE SURFACE 2,20X,1H*/
     37X,1HM,8X,1H*,2(3X,8HVELOCITY,6X,5HW/WCR,33X,1H*))
 1020 FORMAT((1X,G13.4,3H  *,2(2G13.4,A9,20X,1H*)))
```

```
1030 FORMAT (2HPT,50X,24HBLADE SURFACE VELOCITIES/2HPT,49X,27H(BASED ON
    1 FULL WEIGHT FLOW))
1040 FORMAT (2HPL,37X,63HVELOCITY(W) VS. MERIDIONAL STREAMLINE DISTANCE
    1(M) DOWN THE PAGE/2HPL/2HPL,50X,19H* - BLADE SURFACE 1/2HPL,50X,19
    2H+ - BLADE SURFACE 2)
     END




      SUBROUTINE VELGRA(IM)
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,REDFAC,
     1   DENTOL,MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2   TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3   TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4   BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5   SAL(100),AAA(100)
      COMMON /D2TDM2/ D2TDM2(100,2)
      COMMON /WWCRM/WWCRM(100,2),LABEL(100)
      DIMENSION WGRAD(50),THETA(50),RWCB(50),CBETA(50),A2(50),B2(50)
      COMMON /SURVEL/ WTB(100,2),WMB(100,2),XDOWN(400),YACROS(400)
      DIMENSION W(2500),BETA(2500),DUDT(2500),DUDTT(2500),AAP(2500),
     1   BBP(2500)
      EQUIVALENCE (A,W),(A(1,2),BETA),(A(1,3),DUDT),(A(1,4),DUDTT),
     1   (K,AAP),(RHO,BBP)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,SURFBV,
     1FIRST,UPPER,UPPRBV,S1,ST,SRW
      INTEGER CHOKED
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      DATA CHOKED/6HCHOKED/
      AORB = 1.
      IP = IV(IM)
      WGRAD(1) = W(IP)/REDFAC
      IT1 = ITV(IM,1)
      IT = IT1
      NSP = NBBI+1
      GO TO 10
      ENTRY VELGRB(IM)
      IP = IV(IM)-1
      WGRAD(1) = WMB(IM,1)/REDFAC
      NSP = IV(IM+1)-IV(IM)+2
      AORB = 2.
      IT1 = ITV(IM,1)
      IT = IT1-1
   10 NSPM1 = NSP-1
      TORSAL = 2.*ACTOMG*RM(IM)*SAL(IM)
      TWLMR = 2.*ACTOMG*LAMBDA-(ACTOMG*RM(IM))**2
      WCR = SQRT(TGROG*TIP*(1.-TWLMR/CPTIP))
      DELMAX = WCR/ 10.
      TOLERC = WCR/ 100.
      DO 20 I=1,NSPM1
      CBETA(I) = COS(BETA(IP)/57.295779)
      THETA(I) = HT*FLOAT(IT)
```

```
      A2(I) = AAP(IP)
      B2(I) = BBP(IP)
      IT = IT+1
   20 IP = IP+1
      CBETA(NSP) = CBETA(1)
      A2(NSP) = A2(1)
      B2(NSP) = B2(1)
      THETA(NSP) = HT*FLOAT(IT)
      IF(AORB.LE.1.) GO TO 30
      CBETA(1) = 1./SQRT(1.+(RM(IM)*DTDMV(IM,1))**2)
      SBETA1    = SQRT(1.-CBETA(1)**2)*SIGN(1.,DTDMV(IM,1))
      A2(1) = (RM(IM)*CBETA(1))**2*D2TDM2(IM,1)+SAL(IM)*SBETA1/
     1   CBETA(1)*(1.+CBETA(1)**2)
      B2(1) = B2(2)+TORSAL*(1./CBETA(1)-1./CBETA(2))
      THETA(1) = TV(IM,1)
      CBETA(NSP) = 1./SQRT(1.+(RM(IM)*DTDMV(IM,2))**2)
      SBETAN      = SQRT(1.-CBETA(NSP)**2)*SIGN(1.,DTDMV(IM,2))
      A2(NSP) = (RM(IM)*CBETA(NSP))**2*D2TDM2(IM,2)+SAL(IM)*SBETAN/
     1   CBETA(NSP)*(1.+CBETA(NSP)**2)
      B2(NSP) = B2(NSPM1)+TORSAL*(1./CBETA(NSP)-1./CBETA(NSPM1))
      THETA(NSP) = TV(IM,2)
   30 IND = 1
   40 CONTINUE
      DO 50 I=2,NSP
      WAS = WGRAD(I-1)+(A2(I-1)*WGRAD(I-1)+B2(I-1))*(THETA(I)-
     1   THETA(I-1))
      WASS = WGRAD(I-1)+(A2(I)*WAS+B2(I))*(THETA(I)-THETA(I-1))
   50 WGRAD(I) = (WAS+WASS)/2.
      DO 60 I=1,NSP
      TTIP = 1.-(WGRAD(I)**2+TWLMR)/CPTIP
      IF(TTIP.GE..0) GO TO 55
      WRITE (6,1010) IM
      WGRAD(1) = 0.
      WGRAD(NSP) = 0.
      GO TO 80
   55 RHOT = RHOIP*TTIP**EXPON
   60 RWCB(I) = RHOT*WGRAD(I)*CBETA(I)
      CALL INTGRL (THETA,RWCB,NSP,AAA)
      WTFLES = BE(IM)*RM(IM)*AAA(NSP)
      IF (ABS(ACTWT-WTFLES).LE.ACTWT/100000.) GO TO 70
      CALL CONTIN (WGRAD(1),WTFLES,IND,IM,ACTWT,DELMAX,TOLERC)
      IF(IND.LT.6) GO TO 40
      IF (IND.EQ.6) GO TO 65
      WRITE (6,1020) IM
      IF (AORB.GT.1.) WGRAD(1) = 0.
      WGRAD(NSP) = 0.
      GO TO 70
   65 LABEL(IM) = CHOKED
   70 CONTINUE
      FIRST = 1
      IF(AORB.GT.1.) FIRST = 2
      LAST = NSPM1
      IF(INTVL.GT.0) WRITE (6,1000) IM,IT1,(WGRAD(I),I=FIRST,LAST)
   80 IF(AORB.LE.1.) RETURN
      WMB(IM,1) = WGRAD(1)
      WMB(IM,2) = WGRAD(NSP)
      WWCRM(IM,1) = WMB(IM,1)/WCR
      WWCRM(IM,2) = WMB(IM,2)/WCR
```

```
      RETURN
 1000 FORMAT(5HKIM =,I3,10X,5HIT1 =,I3/(2X,10G13.4))
 1010 FORMAT (73HK  A VELOCITY GRADIENT SOLUTION CANNOT BE OBTAINED FOR
     1VERTICAL LINE IM =,I3)
 1020 FORMAT(92HK A VELOCITY GRADIENT SOLUTION COULD NOT BE OBTAINED IN
     150 ITERATIONS FOR VERTICAL LINE IM =,I3)
      END
```

```
      BLOCK DATA
      COMMON /WWCRM/WWCRM(100,2),LABEL(100)
      DATA LABEL/100*6H
      END
```

```
      SUBROUTINE BLCD
C
C  BLCD CALCULATES BLADE THETA COORDINATE AS A FUNCTION OF M
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,REDFAC,
     1   DENTOL,MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/ACTWT,ACTCMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,
     2   TWL,ITMIN,ITMAX,NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),
     3   TV(100,2),DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),
     4   BETAH(100,2),RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100),
     5   SAL(100),AAA(100)
      COMMON /GEOMIN/ CHORD(2),STGR(2),MLE(2),THLE(2),RMI(2),RMO(2),
     1   RI(2),RO(2),BETI(2),BETO(2),NSPI(2),MSP(50,2),THSP(50,2)
      COMMON /BLCDCM/ EM(50,2),INIT(2)
      COMMON /D2TDM2/ D2TDM2(100,2)
      ENTRY  BL1(M,THETA,DTDM,INF)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      REAL M,MMLE,MSPMM,MMMSP
      SURF= 1
      SIGN= 1.
      GO TO 10
      ENTRY  BL2(M,THETA,DTDM,INF)
      SURF= 2
      SIGN=-1.
   10 INF = 0
      IM = 1
      DO 15 I=MBI,MBO
   15 IF(ABS(MV(I)-M).LE.DMLR) IM=I
      NSP= NSPI(SURF)
      IF (INIT(SURF).EQ.13) GO TO 30
      INIT(SURF)= 13
C
C  INITIAL CALCULATION OF FIRST AND LAST SPLINE POINTS ON BLADE
C
```

```
      AA = BETI(SURF)/57.295779
      AA = SIN(AA)
      MSP(1,SURF) =  RI(SURF)*(1.-SIGN*AA)
      BB = SQRT(1.-AA**2)
      THSP(1,SURF) = SIGN*BB*RI(SURF)/RMI(SURF)
      BETI(SURF) = AA/BB/RMI(SURF)
      AA = BETO(SURF)/57.295779
      AA = SIN(AA)
      MSP(NSP,SURF) = CHORD(SURF)-RO(SURF)*(1.+SIGN*AA)
      BB = SQRT(1.-AA**2)
      THSP(NSP,SURF) = STGR(SURF)+SIGN*BB*RO(SURF)/RMO(SURF)
      BETO(SURF) = AA/BB/RMO(SURF)
      DO 20 IA=1,NSP
      MSP(IA,SURF)= MSP(IA,SURF)+MLE(SURF)
   20 THSP(IA,SURF)= THSP(IA,SURF)+THLE(SURF)
      CALL SPLN22(MSP(1,SURF),THSP(1,SURF),BETI(SURF),BETO(SURF),NSP,
     1 AAA,EM(1,SURF))
      IF(BLDAT.LE.0) GO TO 30
      IF (SURF.EQ.1) WRITE(6,1000)
      WRITE(6,1010) SURF
      WRITE (6,1020) (MSP(IA,SURF),THSP(IA,SURF),AAA(IA),EM(IA,SURF),
     1    IA=1,NSP)
C
C   BLADE COORDINATE CALCULATION
C
   30 KK = 2
      IF (M.GT.MSP(1,SURF)) GO TO 50
C
C   AT LEADING EDGE RADIUS
C
      MMLE= M-MLE(SURF)
      IF (MMLE.LT.-DMLR) GO TO 90
      MMLE= AMAX1(0.,MMLE)
      THETA= SQRT(MMLE*(2.*RI(SURF)-MMLE))*SIGN
      IF (THETA.EQ.0.) GO TO 40
      RMM= RI(SURF)-MMLE
      DTDM= RMM/THETA/RMI(SURF)
      THETA = THETA/RMI(SURF)
      D2TDM2(IM,SURF) = (-THETA-RMM*DTDM)/(RMI(SURF)*THETA)**2
      THETA = THETA+THLE(SURF)
      RETURN
   40 INF= 1
      DTDM = 1.E10*SIGN
      THETA= THLE(SURF)
      D2TDM2(IM,SURF) = 0.
      RETURN
C
C   ALONG SPLINE CURVE
C
   50 IF (M.LE.MSP(KK,SURF)) GO TO 60
      IF (KK.GE.NSP) GO TO 70
      KK = KK+1
      GO TO 50
   60 S= MSP(KK,SURF)-MSP(KK-1,SURF)
      EMKM1= EM(KK-1,SURF)
      EMK= EM(KK,SURF)
      MSPMM= MSP(KK,SURF)-M
      MMMSP= M-MSP(KK-1,SURF)
      THK= THSP(KK,SURF)/S
      THKM1= THSP(KK-1,SURF)/S
```

```
      THETA= EMKM1*MSPMM**3/6./S + EMK*MMMSP**3/6./S + (THK-EMK*S/6.)*
     1 MMMSP + (THKM1-EMKM1*S/6.)*MSPMM
      DTDM= -EMKM1*MSPMM**2/2./S + EMK*MMMSP**2/2./S + THK-THKM1-(EMK-
     1 EMKM1)*S/6.
      D2TDM2(IM,SURF) = EMKM1*MSPMM/S+EMK*MMMSP/S
      RETURN
C
C   AT TRAILING EDGE RADIUS
C
   70 CMM= CHORD(SURF)+MLE(SURF)-M
      IF (CMM.LT.-DMLR) GO TO 90
      CMM= AMAX1(0.,CMM)
      THETA= SQRT(CMM*(2.*RO(SURF)-CMM))*SIGN
      IF (THETA.EQ.0.) GO TO 80
      RMM= RO(SURF)-CMM
      DTDM = -RMM/THETA/RMO(SURF)
      THETA = THETA/RMO(SURF)
      D2TDM2(IM,SURF) = (-THETA+RMM*DTDM)/(RMO(SURF)*THETA)**2
      THETA = THETA+STGR(SURF)+THLE(SURF)
      RETURN
   80 INF= 1
      DTDM = -1.E10*SIGN
      THETA= THLE(SURF)+STGR(SURF)
      D2TDM2(IM,SURF) = 0.
      RETURN
C
C   ERROR RETURN
C
   90 WRITE(6,1030) LER(2),M,SURF
      STOP
 1000 FORMAT (1H1,13X,33HBLADE DATA AT INPUT SPLINE POINTS)
 1010 FORMAT(1HL,17X,16HBLADE    SURFACE,I4)
 1020 FORMAT (7X ,1HM,10X,5HTHETA,10X,10HDERIVATIVE,5X,10H2ND DERIV. /
     1   (4G15.5) )
 1030 FORMAT (14HLBLCD CALL NO.,I3/33H M COORDINATE IS NOT WITHIN BLADE/
     14H M =,G14.6,10X,6HSURF =,G14.6)
      END
```

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

Subroutine CONTIN

CONTIN calculates a new estimate for the initial value of W for equation (4). This is based on satisfying the continuity equation (7). If the input value of w (WTFL) is too large, there may be no solution of equation (4) satisfying equation (7). In this case, the choking weight flow will be found.

An initial estimate of the velocity at the lower boundary is furnished by VELGRA, say W_1. The corresponding weight flow w_1 is also calculated by VELGRA. CONTIN furnishes the next estimate W_2, by linear interpolation or extrapolation from the origin (see fig. 16). Subsequent estimates are obtained by linear interpolation or extrapolation from the two previous estimates (see W_3 in fig. 16). This is essentially the method of false position (regula falsi).

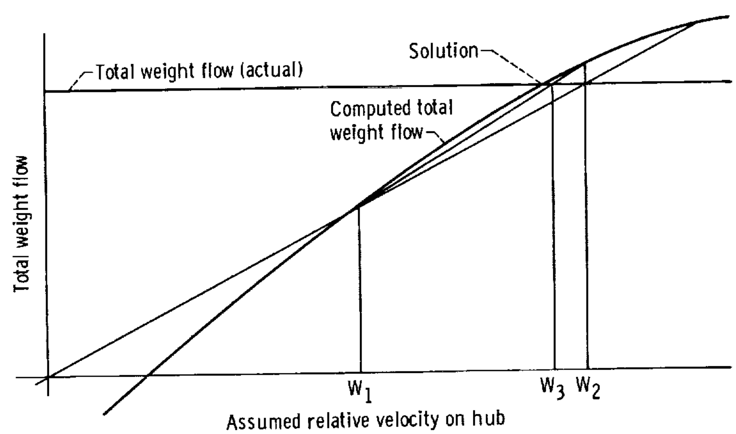


Figure 16. - Method used by subroutine CONTIN to determine relative hub velocity.

If there is choked flow, so that a solution does not exist, information from three iterations is stored. This information is used to predict the next estimate of W on the lower boundary such that the weight flow will be a maximum.

The input arguments for CONTIN are as follows:

WA	last value of W on lower boundary used in solving eq. (4)
WTFL	weight flow calculated by eq. (9) based on the input value of WA
IND	controls sequence of calculation in CONTIN; VELGRA sets IND = 1 to indicate start of velocity-gradient solution for a new vertical mesh line
I	value of IM for vertical mesh line
WT	input weight flow, w

DELMAX maximum permitted change in estimated velocity WA per iteration

TOLERC tolerance on velocity for calculating choking weight flow

The output arguments for CONTIN are as follows:

WA value of W on lower boundary to be used in solving eq. (4)

IND used to control next iteration in CONTIN, and to indicate when a choked flow solution has been found

The internal variables for CONTIN are as follows:

DELTA predicted correction to WA

NCALL number of iterations of CONTIN for a vertical mesh line

SPEED array of values of WA from up to three previous iterations

WEIGHT array of values of WTFL from up to three previous iterations

```
      SUBROUTINE CONTIN (WA,WTFL,IND,I,WT,DELMAX,TOLERC)
      DIMENSION SPEED(3),WEIGHT(3)
      NCALL = NCALL + 1
      IF (IND.NE.1.AND.NCALL.GT.50) GO TO 400
  135 GO TO (140,150,210,270,370),IND
  140 SPEED(1) = WA
      WEIGHT(1) = WTFL
      DELTA = WT/WTFL*WA-WA
      IF(ABS(DELTA).GT.DELMAX) DELTA = SIGN(DELMAX,DELTA)
      IF(WTFL.LT.0.) DELTA = DELMAX
      WA = DELTA+WA
      IND = 2
      NCALL = 1
      RETURN
  150 IF ((WTFL-WEIGHT(1))/(WA-SPEED(1))) 180,180,160
  160 SPEED(2) = WA
      DELTA = (WT-WTFL)/(WTFL-WEIGHT(1))*(WA-SPEED(1))
      IF(ABS(DELTA).GT.DELMAX) DELTA = SIGN(DELMAX,DELTA)
      WA = DELTA+WA
  166 SPEED(1) = SPEED(2)
      WEIGHT(1) = WTFL
      RETURN
  170 WRITE (6,1000) I,WTFL, I
      IND = 6
      RETURN
  180 IND = 3
      IF (WTFL.GE.WT) GO TO 140
      IF (SPEED(1)-WA) 190,200,200
  190 SPEED(2) = SPEED(1)
      SPEED(1) = 2.0*SPEED(1)-WA
      SPEED(3) = WA
      WEIGHT(2) = WEIGHT(1)
      WEIGHT(3) = WTFL
      WA = SPEED(1)
      RETURN
  200 SPEED(2) = WA
```

```
      SPEED(3) = SPEED(1)
      SPEED(1) = 2.0*WA-SPEED(1)
      WEIGHT(2) = WTFL
      WEIGHT(3) = WEIGHT(1)
      WA = SPEED(1)
      RETURN
  210 WEIGHT(1) = WTFL
      IF (WTFL.GE.WT) GO TO 140
      IF (WEIGHT(1)-WEIGHT(2)) 230,380,220
  220 WEIGHT(3) = WEIGHT(2)
      WEIGHT(2) = WEIGHT(1)
      SPEED(3) = SPEED(2)
      SPEED(2) = SPEED(1)
      SPEED(1) = 2.0*SPEED(2)-SPEED(3)
      WA = SPEED(1)
      RETURN
  230 IF(SPEED(3)-SPEED(1)-TOLERC) 170, 170, 240
  240 IND = 4
  245 IF (WEIGHT(3)-WEIGHT(1)) 260,260,250
  250 WA = (SPEED(1)+SPEED(2))/2.0
      RETURN
  260 WA = (SPEED(3)+SPEED(2))/2.0
      RETURN
  270 IF(SPEED(3)-SPEED(1)-TOLERC) 170, 170, 280
  280 IF (WTFL-WEIGHT(2)) 320,350,290
  290 IF (WA-SPEED(2)) 310,300,300
  300 SPEED(1) = SPEED(2)
      SPEED(2) = WA
      WEIGHT(1) = WEIGHT(2)
      WEIGHT(2) = WTFL
      GO TO 245
  310 SPEED(3) = SPEED(2)
      SPEED(2) = WA
      WEIGHT(3) = WEIGHT(2)
      WEIGHT(2) = WTFL
      GO TO 245
  320 IF (WA-SPEED(2)) 340,330,330
  330 WEIGHT(3) = WTFL
      SPEED(3) = WA
      GO TO 245
  340 WEIGHT(1) = WTFL
      SPEED(1) = WA
      GO TO 245
  350 IND = 5
      IF (WA-SPEED(2)) 380,360,360
  360 SPEED(1) = SPEED(2)
      WEIGHT(1) = WEIGHT(2)
      SPEED(2) = (SPEED(1)+SPEED(3))/2.0
      WA = SPEED(2)
      RETURN
  370 IND = 4
      WEIGHT(2) = WTFL
      WA = (SPEED(1)+SPEED(2))/2.0
      RETURN
  380 IND = 5
  390 WEIGHT(3) = WEIGHT(2)
      SPEED(3) = SPEED(2)
      SPEED(2) = (SPEED(1)+SPEED(3))/2.
      WA = SPEED(2)
      RETURN
```

```
C    NO SOLUTION FOUND IN 50 ITERATIONS
  400 IND = 7
      RETURN
 1000 FORMAT(43HLACTWT EXCEEDS CHOKING WEIGHT FLOW FOR IM =,I3/
     1    22HKCHOKING WEIGHT FLOW =,G15.6,9H FOR IM =,I3)
      END
```

Subroutine INTGRL

INTGRL calculator is the integral of a function passing through a given set of points. This subroutine is based on the spline curve. INTGRL solves a tridiagonal matrix equation given in reference 9 to obtain the coefficients for the piecewise cubic polynomial function giving the spline fit curve. INTGRL is based on the end condition that the second derivative at either end point is one-half that at the next spline point.

The input variables are as follows:

X   array of ordinates

Y   array of function values

N   integer number of X and Y values given

The output variable is

SUM   array of values of integral of function, $SUM(J) = \int_{X(1)}^{X(J)} Y\,dX$

If SRW = 17 in COMMON, input and output data for INTGRL are printed. This is useful in debugging.

```
      SUBROUTINE INTGRL (X,Y,N,SUM)
C
C    INTGRL CALCULATES THE INTEGRAL OF A SPLINE CURVE PASSING THROUGH
C    A GIVEN SET OF POINTS
C    END CONDITION - SECOND DERIVATIVE AT EITHER END POINT IS ONE-HALF
C    THAT AT THE ADJACENT POINT
C
      COMMON SRW
      COMMON /BOX/ G(50), SB(50), EM(100)
      DIMENSION X(N), Y(N), SUM(N)
      INTEGER SRW
      SB(1) = -.5
      G(1) = 0
      NO=N-1
      IF(NO.LT.2) GO TO 20
      DO 10 I=2,NO
      A = (X(I)-X(I-1))/6.
      C = (X(I+1)-X(I))/6.
      W = 2.*(A+C)-A*SB(I-1)
      SB(I) = C/W
      F = (Y(I+1)-Y(I))/(X(I+1)-X(I))-(Y(I)-Y(I-1))/(X(I)-X(I-1))
```

```
   10 G(I) = (F-A*G(I-1))/W
   20 EM(N) = G(N-1)/(2.+SB(N-1))
      DO 30 I=2,N
      K = N+1-I
   30 EM(K) = G(K)-SB(K)*EM(K+1)
      SUM(1) =0.0
      DO 50 K=2,N
   50 SUM(K) = SUM(K-1)+(X(K)-X(K-1))* (Y(K)+Y(K-1))/2.0-(X(K)-X(K-1))
     1**3*(EM(K)+EM(K-1))/24.0
      IF(SRW.EQ.17) WRITE(6,1000) N,(X(I),Y(I),SUM(I),EM(I),I=1,N)
      RETURN
 1000 FORMAT (17HK NO. OF POINTS =I3/10X5HX     15X5HY     15X5HSUM
     1    13X10H2ND DERIV./(4E20.8))
      END
```

The remaining subroutines are essentially the same as described in reference 1. The subroutines in TSONIC are not interchangeable with those in TANDEM or TURBLE, since there are differences in COMMON blocks and some changes in coding. However, the description of these subroutines in reference 1 still applies, with the exception of SPLINE and ROOT. In SPLINE the end condition has been changed so that the second derivative is the same at an end point as at the adjacent point. ROOT has been changed to find the root by the bisection method instead of by Newton's method. This is less efficient, but more foolproof.

```
      SUBROUTINE SPLINE (X,Y,N,SLOPE,EM)
C
C  SPLINE CALCULATES FIRST AND SECOND DERIVATIVES AT SPLINE POINTS
C  END CONDITION - SECOND DERIVATIVES ARE THE SAME AT END POINT AND
C  ADJACENT POINT
C
      COMMON SRW
      COMMON /BOX/ G(100),SB(100)
      DIMENSION X(N),Y(N),EM(N),SLOPE(N)
      INTEGER SRW
      SB(1) = -1.0
      G(1) = 0
      NO=N-1
      IF(NO.LT.2) GO TO 20
      DO 10 I=2,NO
      A = (X(I)-X(I-1))/6.
      C = (X(I+1)-X(I))/6.
      W = 2.*(A+C)-A*SB(I-1)
      SB(I) = C/W
      F = (Y(I+1)-Y(I))/(X(I+1)-X(I))-(Y(I)-Y(I-1))/(X(I)-X(I-1))
   10 G(I) = (F-A*G(I-1))/W
   20 EM(N) = G(N-1)/(1.+SB(N-1))
      DO 30 I=2,N
      K = N+1-I
   30 EM(K) = G(K)-SB(K)*EM(K+1)
      SLOPE(1) = (X(1)-X(2))/6.*(2.*EM(1)+EM(2))+(Y(2)-Y(1))/(X(2)-X(1))
      DO 40 I=2,N
   40 SLOPE(I) = (X(I)-X(I-1))/6.*(2.*EM(I)+EM(I-1))+(Y(I)-Y(I-1))/
     1    (X(I)-X(I-1))
      IF(SRW.EQ.13) WRITE (6,1000) N,(X(I),Y(I),SLOPE(I),EM(I),I=1,N)
      RETURN
```

```
 1000 FORMAT (2X,15HNO. OF POINTS =,I3/10X,1HX,19X,1HY,19X,5HSLOPE,15X,
     12HEM/(4G20.8))
      END



      SUBROUTINE SPLN22 (X,Y,Y1P,YNP,N,SLOPE,EM)
C
C   SPLN22 CALCULATES FIRST AND SECOND DERIVATIVES AT SPLINE POINTS
C   END CONDITION - DERIVATIVES SPECIFIED AT END POINTS
C
      COMMON SRW
      COMMON /BOX/ G(100),SB(100)
      DIMENSION X(N),Y(N),EM(N),SLOPE(N)
      INTEGER SRW
      SB(1) = .5
      F = (Y(2)-Y(1))/(X(2)-X(1))-Y1P
      G(1) = F*3./(X(2)-X(1))
      NO=N-1
      IF(NO.LT.2) GO TO 20
      DO 10 I=2,NO
      A = (X(I)-X(I-1))/6.
      C = (X(I+1)-X(I))/6.
      W = 2.*(A+C)-A*SB(I-1)
      SB(I) = C/W
      F = (Y(I+1)-Y(I))/(X(I+1)-X(I))-(Y(I)-Y(I-1))/(X(I)-X(I-1))
   10 G(I) = (F-A*G(I-1))/W
   20 F = YNP-(Y(N)-Y(N-1))/(X(N)-X(N-1))
      W = (X(N)-X(N-1))/6.*(2.-SB(N-1))
      EM(N) = (F-(X(N)-X(N-1))*G(N-1)/6.)/W
      DO 30 I=2,N
      K = N+1-I
   30 EM(K) = G(K)-SB(K)*EM(K+1)
      SLOPE(1) = (X(1)-X(2))/6.*(2.*EM(1)+EM(2))+(Y(2)-Y(1))/(X(2)-X(1))
      DO 40 I=2,N
   40 SLOPE(I) = (X(I)-X(I-1))/6.*(2.*EM(I)+EM(I-1))+(Y(I)-Y(I-1))/
     1    (X(I)-X(I-1))
      IF(SRW.EQ.18) WRITE (6,1000) N,(X(I),Y(I),SLOPE(I),EM(I),I=1,N)
      RETURN
 1000 FORMAT (2X,15HNO. OF POINTS =,I3/10X,1HX,19X,1HY,19X,5HSLOPE,15X,
     12HEM/(4G20.8))
      END



      SUBROUTINE SPLINT (X,Y,N,Z,MAX,YINT,DYDX)
C
C   SPLINT CALCULATES INTERPOLATED POINTS AND DERIVATIVES
C   FOR A SPLINE CURVE
C    END CONDITION - SECOND DERIVATIVE AT EITHER END POINT IS ONE-HALF
C    THAT AT THE ADJACENT POINT
C
      COMMON SRW
      COMMON /BOX/ G(100),SB(100)
      DIMENSION X(N),Y(N),Z(MAX),YINT(MAX),DYDX(MAX)
      DIMENSION EM(100)
```

```
      EQUIVALENCE (SB,EM)
      INTEGER SRW
      IF(MAX.LE.0) RETURN
      III = SRW
      SB(1) = -.5
      G(1) = 0
      NO=N-1
      IF(NO.LT.2) GO TO 20
      DO 10 I=2,NO
      A = (X(I)-X(I-1))/6.
      C = (X(I+1)-X(I))/6.
      W = 2.*(A+C)-A*SB(I-1)
      SB(I) = C/W
      F = (Y(I+1)-Y(I))/(X(I+1)-X(I))-(Y(I)-Y(I-1))/(X(I)-X(I-1))
   10 G(I) = (F-A*G(I-1))/W
   20 EM(N) = G(N-1)/(2.+SB(N-1))
      DO 30 I=2,N
      K = N+1-I
   30 EM(K) = G(K)-SB(K)*EM(K+1)
      DO 140 I=1,MAX
      K=2
      IF(Z(I)-X(1)) 70,60,90
   60 YINT(I)=Y(1)
      SK = X(K)-X(K-1)
      GO TO 130
   70 IF(Z(I).GE.(1.1*X(1)-.1*X(2))) GO TO 120
      WRITE (6,1000) Z(I)
      SRW = 16
      GO TO 120
   80 K=N
      IF(Z(I).LE.(1.1*X(N)-.1*X(N-1))) GO TO 120
      WRITE (6,1000) Z(I)
      SRW = 16
      GO TO 120
   90 IF(Z(I)-X(K)) 120,100,110
  100 YINT(I)=Y(K)
      SK = X(K)-X(K-1)
      GO TO 130
  110 K=K+1
      IF(K-N) 90,90,80
  120 CONTINUE
      SK = X(K)-X(K-1)
      YINT(I) = EM(K-1)*(X(K)-Z(I))**3/6./SK  +EM(K)*(Z(I)-X(K-1))**3/6.
     1  /SK+(Y(K)/SK  -EM(K)*SK  /6.)*(Z(I)-X(K-1))+(Y(K-1)/SK  -EM(K-1)
     2  *SK/6.)*(X(K)-Z(I))
  130 DYDX(I)=-EM(K-1)*(X(K)-Z(I))**2/2.0/SK  +EM(K)*(X(K-1)-Z(I))**2/2.
     1   /SK+(Y(K)-Y(K-1))/SK  -(EM(K)-EM(K-1))*SK/6.
  140 CONTINUE
      MXA = MAX0(N,MAX)
      IF(SRW.EQ.16) WRITE(6,1010) N,MAX,(X(I),Y(I),Z(I),YINT(I),DYDX(I),
     1   I=1,MXA)
      SRW = III
      RETURN
 1000 FORMAT (54H SPLINT USED FOR EXTRAPOLATION.  EXTRAPOLATED VALUE = ,
     1G14.6)
 1010 FORMAT (2X,21HNO. OF POINTS GIVEN =,I3,30H, NO. OF INTERPOLATED PO
     1INTS =,I3/10X,1HX,19X,1HY,16X,11HX-INTERPOL.,9X,11HY-INTERPOL.,
     28X,14HDYDX-INTERPOL./(5E20.8))
      END
```

```
      SUBROUTINE MHORIZ(MV,ITV,BL,MBI,MBO,ITO,HT,DTLR,KODE,J,MH,DTDMH,
     1MRTS)
C
C  MHORIZ CALCULATES M COORDINATES OF INTERSECTIONS OF ALL HORIZONTAL
C  MESH LINES WITH A BLADE SURFACE
C  KODE = 0 FOR UPPER BLADE SURFACE
C  KODE = 1 FOR LOWER BLADE SURFACE
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      DIMENSION MV(100),ITV(100),MH(100),DTDMH(100)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,SI,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      REAL MVIM
      EXTERNAL BL
      IF (MBI.GE.MBO) RETURN
      IM= MBI
   10 ITIND= 0
   20 IF (ITV(IM+1)-ITV(IM)-ITIND) 30,40,50
   30 J= J+1
      TI= FLOAT(ITV(IM+1)-ITO-ITIND+KODE)*HT
      ITIND= ITIND-1
      MVIM = MV(IM)
      IF (MRTS.EQ.1) MVIM = MVIM+(MV(IM+1)-MVIM)/1000.
      CALL ROOT (MVIM,MV(IM+1),TI,BL,DTLR,MH(J),DTDMH(J))
      GO TO 20
   40 IM= IM+1
      MRTS = 0
      IF (IM.EQ.MBO) RETURN
      GO TO 10
   50 J= J+1
      TI= FLOAT(ITV(IM)-ITO+ITIND+KODE)*HT
      ITIND= ITIND+1
      MVIM = MV(IM)
      IF (MRTS.EQ.1) MVIM = MVIM+(MV(IM+1)-MVIM)/1000.
      CALL ROOT(MVIM  ,MV(IM+1),TI,BL,DTLR,MH(J),DTDMH(J))
      GO TO 20
      END


      SUBROUTINE DENSTY(RHOW,RHO,VEL,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP)
C
C  DENSTY CALCULATES DENSITY AND VELOCITY FROM THE WEIGHT FLOW PARAMETER
C  DENSITY TIMES VELOCITY
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      VEL = RHOW/RHO
      IF (VEL.NE.0.) GO TO 10
      RHO = RHOIP
      RETURN
   10 TTIP = 1.-(VEL**2+TWLMR)/CPTIP
      IF(TTIP.LT.0.) GO TO 30
      TEMP = TTIP**(EXPON-1.)
      RHOT = RHOIP*TEMP*TTIP
      RHOWP = -VEL**2/GAM*RHOIP/AR*TEMP/TIP+RHOT
      IF(RHOWP.LE.0.) GO TO 30
      VELNEW = VEL+(RHOW-RHOT*VEL)/RHOWP
      IF(ABS(VELNEW-VEL)/VELNEW.LT..0001) GO TO 20
      VEL = VELNEW
```

```
      GO TO 10
   20 VEL = VELNEW
      RHO = RHOW/VEL
      RETURN
   30 TGROG = 2.*GAM*AR/(GAM+1.)
      VEL = SQRT(TGROG*TIP*(1.-TWLMR/CPTIP))
      RHO = RHOIP*(1.-(VEL**2+TWLMR)/CPTIP)**EXPON
      RWMURW = RHOW/RHO/VEL
      NER(1) = NER(1)+1
      WRITE(6,1000) LER(1),NER(1),RWMORW
      IF(NER(1).EQ.50) STOP
      RETURN
 1000 FORMAT(16HLDENSTY CALL NO.,I3/9H NER(1) =,I3/10H RHO*W IS ,F7.4,
     134H TIMES THE MAXIMUM VALUE FOR RHO*W)
      END


      SUBROUTINE ROOT(A,B,Y,FUNCT,TOLERY,X,DFX)
C
C  ROOT FINDS A ROOT FOR (FUNCT MINUS Y) IN THE INTERVAL (A,B)
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      INTEGER SRW
      IF (SRW.EQ.21) WRITE(6,1000) A,B,Y,TOLERY
      X1 = A
      CALL FUNCT(X1,FX1,DFX,INF)
      IF(SRW.EQ.21) WRITE(6,1010) X1,FX1,DFX,INF
      X2 = B
   10 DO 30 I=1,20
      X = (X1+X2)/2.
      CALL FUNCT(X,FX,DFX,INF)
      IF(SRW.EQ.21) WRITE(6,1010) X,FX,DFX,INF
      IF((FX1-Y)*(FX-Y).GT.0.) GO TO 20
      X2 = X
      GO TO 30
   20 X1 = X
      FX1 = FX
   30 CONTINUE
      IF(ABS(Y-FX).LT.TOLERY) RETURN
      WRITE(6,1020) LER(2),A,B,Y
      STOP
 1000 FORMAT (32H1INPUT ARGUMENTS FOR ROOT -- A =G13.5,3X,3HB =,G13.5,
     1    3X,3HY =,G13.5,3X,8HTOLERY =,G13.5/16X,1HX,17X,2HFX,15X,3HDFX,
     2    10X,3HINF)
 1010 FORMAT(8X,G16.5,2G18.5,I6)
 1020 FORMAT(14HLROOT CALL NO.,I3/37H ROOT HAS FAILED TO OBTAIN VALID RO
     1CT/4H A =,G14.6,10X,3HB =,G14.6,10X,3HY =,G14.6)
      END
```

Lewis Research Center,

National Aeronautics and Space Administration,

Cleveland, Ohio, June 27, 1969,

720-03-00-66-22.

APPENDIX A

DERIVATION OF VELOCITY-GRADIENT EQUATION

The velocity-gradient equation is an expression of the force equation. By a balance of force in the θ-direction, the following equation can be obtained:

$$\frac{\partial W}{\partial \theta} = \frac{1}{W} \frac{d(rV_\theta)}{dt} = \frac{1}{W} \frac{d(rW_\theta + \omega r^2)}{dt} \tag{A1}$$

The time derivative indicates the change in the quantity for a moving particle as a function of time. Equation (A1) is a special case of equation (B10) of reference 11. We make use of the following relations (see fig. 1):

$$W_\theta = W \sin \beta$$

$$W_m = W \cos \beta$$

$$W_r = W_m \sin \alpha$$

$$W_z = W_m \cos \alpha$$

$$\frac{dr}{dt} = W_r$$

$$r \frac{d\theta}{dt} = W_\theta$$

$$\frac{dm}{dt} = W_m$$

$$\frac{dS}{dt} = W$$

$$\frac{d\beta}{dt} = W \frac{d\beta}{dS}$$

$$\frac{dW}{dt} = W_\theta \frac{\partial W}{r \partial \theta} + W_m \frac{\partial W}{\partial m}$$

We can perform the indicated differentiation of the right side of equation (A1) by using the preceding relations, and then solve for $\partial W/\partial\theta$ (which appears on both sides of the equation) to obtain

$$\frac{\partial W}{\partial \theta} = W\left(\frac{r}{\cos\beta}\frac{d\beta}{dS} + \sin\alpha\tan\beta\right) + r\tan\beta\frac{\partial W}{\partial m} + 2\omega r\frac{\sin\alpha}{\cos\beta} \tag{A2}$$

We desire now to evaluate $d\beta/dS$ in terms of first and second derivatives of θ with respect to m along streamlines. The following relations hold for streamlines:

$$\tan\beta = \frac{r d\theta}{dm} \tag{A3}$$

$$\tan\beta = \frac{W_\theta}{W_m} = -\frac{\frac{r\partial u}{\partial m}}{\frac{\partial u}{\partial \theta}} \tag{A4}$$

Equation (A4) is obtained by using equations (2) and (3). Also along streamlines we have

$$\frac{dm}{dS} = \cos\beta \tag{A5}$$

$$\frac{dr}{dm} = \sin\alpha \tag{A6}$$

Now differentiate equation (A3) and use equations (A5) and (A6) to obtain

$$\frac{d\beta}{dS} = r\cos^3\beta\frac{d^2\theta}{dm^2} + \frac{\sin\alpha\sin\beta\cos^2\beta}{r} \tag{A7}$$

Along the surface of the blade $d^2\theta/dm^2$ can be easily calculated since θ is given explicitly as a function of m. However, in the passage $d^2\theta/dm^2$ is given indirectly by the stream function. Hence, we will need an expression for $d^2\theta/dm^2$ in terms of the partial derivatives of the stream function. First, from equations (A3) and (A4) we have

$$\frac{d\theta}{dm} = -\frac{\frac{\partial u}{\partial m}}{\frac{\partial u}{\partial \theta}} \tag{A8}$$

By differentiating equation (A8) we obtain

$$\frac{d^2\theta}{dm^2} = \frac{\partial}{\partial m}\left(\frac{d\theta}{dm}\right) + \frac{\partial}{\partial \theta}\left(\frac{d\theta}{dm}\right)\frac{d\theta}{dm} = \frac{2\,\frac{\partial u}{\partial \theta}\,\frac{\partial u}{\partial m}\,\frac{\partial^2 u}{\partial \theta\,\partial m} - \left(\frac{\partial u}{\partial \theta}\right)^2 \frac{\partial^2 u}{\partial m^2} - \left(\frac{\partial u}{\partial m}\right)^2 \frac{\partial^2 u}{\partial \theta^2}}{\left(\frac{\partial u}{\partial \theta}\right)^3} \tag{A9}$$

Finally, by using the fact (from eqs. (A3) and (A8)) that

$$\frac{r \cos \beta}{\frac{\partial u}{\partial \theta}} = -\frac{\sin \beta}{\frac{\partial u}{\partial m}}$$

we can obtain

$$r^2 \cos^2 \beta \frac{d^2\theta}{dm^2} = \sin^2 \beta \left[2\,\frac{\frac{\partial^2 u}{\partial \theta\,\partial m}}{\frac{\partial u}{\partial m}} - \frac{\frac{\partial u}{\partial \theta}}{\left(\frac{\partial u}{\partial m}\right)^2}\,\frac{\partial^2 u}{\partial m^2} - \frac{\frac{\partial^2 u}{\partial \theta^2}}{\frac{\partial u}{\partial \theta}}\right] \tag{A10}$$

By using equations (A7) and (A10), equation (A2) can be put in the following form:

$$\frac{\partial W}{\partial \theta} = AW + B \tag{A11}$$

where

$$A = r^2 \cos^2 \beta \frac{d^2\theta}{dm^2} + \sin \alpha \tan \beta(1 + \cos^2 \beta) \tag{A12a}$$

is used on blade surface,

$$A = \sin^2 \beta \left[2\,\frac{\frac{\partial^2 u}{\partial \theta\,\partial m}}{\frac{\partial u}{\partial m}} - \frac{\frac{\partial u}{\partial \theta}}{\left(\frac{\partial u}{\partial m}\right)^2}\,\frac{\partial^2 u}{\partial m^2} - \frac{\frac{\partial^2 u}{\partial \theta^2}}{\frac{\partial u}{\partial \theta}}\right] + \sin \alpha \tan \beta(1 + \cos^2 \beta) \tag{A12b}$$

is used at interior points, and

$$B = r \tan \beta \frac{\partial W}{\partial m} + 2\omega r \frac{\sin \alpha}{\cos \beta} \tag{A13}$$

Equations (A11) to (A13) are in the form used in the program.

APPENDIX B

DEFINING REDUCED WEIGHT FLOW PROBLEM

Since the final solution obtained by the velocity-gradient equation depends on the stream-function solution obtained with a reduced weight flow, it is important to establish the conditions which will give the most suitable stream-function solution. To accomplish this, the streamlines at the reduced weight flow should correspond in shape as closely as possible to those for the actual weight flow. Some of the factors affecting this correspondence are discussed in the following paragraphs.

The first condition is that equation (1) should not be changed. This condition is satisfied if the ratio ω/w is not changed. Therefore, ω is reduced in the same ratio as w, for the reduced weight flow solution. With this condition satisfied, there would be no change at all in the stream function, if the flow were incompressible. With compressible flow, the stream-function solution will change since the coefficients in equation (1) are functions of the density ρ, which in turn is a function of the relative velocity W. The relative velocity naturally will be lower if the weight flow is reduced.

Another consideration is the boundary conditions at the upstream and downstream boundaries, AH and DE (see fig. 4). We could use the same boundary condition as for the full weight flow. However, this is not the best approximation. The reason for this is that the input information is the mean value of β_{le} and β_{te} at BG and CF, respectively, instead of AH and DE. And we want to obtain a streamline pattern satisfying the input condition, but at a reduced weight flow. So the entire calculation of β_{in} and β_{out} for the stream-function solution is based on the reduced weight flow condition. Notice, also, that the calculation of β_{in} and β_{out} requires a value of prerotation λ, which in turn depends on ω and w. Hence, a value of λ based on the reduced weight flow is also calculated. The method of calculating λ and β_{in} and β_{out} is described in appendix B of reference 1.

We can summarize the quantities which must be calculated based on the reduced weight flow. They are w, ω, λ, β_{in}, and β_{out}. These quantities, based on the reduced weight flow, are used for calculating the reduced weight flow solution. Values of w, ω, and λ based on the actual weight flow are also calculated for later use in the velocity-gradient equations.

REFERENCES

1. Katsanis, Theodore; and McNally, William D.: FORTRAN Program for Calculating Velocities and Streamlines on a Blade-to-Blade Stream Surface of a Tandem Blade Turbomachine. NASA TN D-5044, 1969.

2. Katsanis, Theodore; and Dellner, Lois T.: A Quasi-Three-Dimensional Method for Calculating Blade Surface Velocities for an Axial Flow Turbine Blade. NASA TM X-1394, 1967.

3. Katsanis, Theodore; and McNally, William D.: Revised FORTRAN Program for Calculating Velocities and Streamlines on a Blade-to-Blade Stream Surface of a Turbomachine. NASA TM X-1764, 1969.

4. Shapiro, Ascher H.: The Dynamics and Thermodynamics of Compressible Fluid Flow. Vol. I. The Ronald Press Co., 1953.

5. Bers, Lipman.: Mathematical Aspects of Subsonic and Transonic Gas Dynamics. John Wiley & Sons, Inc., 1958.

6. 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.

7. Katsanis, Theodore: Computer Program for Calculating Velocities and Streamlines on a Blade-to-Blade Stream Surface of a Turbomachine. NASA TN D-4525, 1968.

8. Mechtly, E. A.: The International System of Units. Physical Constants and Conversion Factors. NASA SP-7012, 1964.

9. 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.

10. Hildebrand, F. B.: Introduction to Numerical Analysis. McGraw-Hill Book Co., Inc., 1956.

11. Katsanis, Theodore: Use of Arbitrary Quasi-Orthogonals for Calculating Flow Distribution in the Meridional Plane of a Turbomachine. NASA TN D-2546, 1964.

NASA-Langley, 1969 —— 1 E-5046