ANALYTICAL INVESTIGATION OF
SUPersonic TURBOMACHINERY BLADING

I - Computer Program for Blading Design

by Louis J. Goldman and Vincent J. Scullin

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MARCH 1968
ANALYTICAL INVESTIGATION OF SUPERSONIC TURBOMACHINERY BLADING

I - COMPUTER PROGRAM FOR BLADING DESIGN

By Louis J. Goldman and Vincent J. Scullin

Lewis Research Center
Cleveland, Ohio
CONTENTS

SUMMARY .......................................................... 1

INTRODUCTION ................................................... 1

SYMBOLS .......................................................... 2

METHOD OF ANALYSIS ........................................... 4
  Blade Description ............................................ 4
  Blade Design .................................................. 8
    Circular arcs ............................................. 8
    Lower transition arcs .................................... 10
    Upper transition arcs .................................... 14
  Geometric parameters ...................................... 14
  Design Limitations ......................................... 16
    Supersonic starting ...................................... 16
    Flow separation ......................................... 20

DESCRIPTION OF INPUT ....................................... 23

DESCRIPTION OF OUTPUT ...................................... 23

PROGRAM DESCRIPTION ....................................... 30
  Main Program ................................................ 30
  Subroutine ROOT ............................................ 36
  Subroutine START .......................................... 37
  Subroutine MSSTAR ......................................... 39
  Subroutine SIMPS1 .......................................... 39
  Function Subprograms ...................................... 40

PROGRAM LISTING .............................................. 42

REFERENCES .................................................... 61
ANALYTICAL INVESTIGATION OF SUPERSONIC TURBOMACHINERY BLADING

I - COMPUTER PROGRAM FOR BLADING DESIGN

by Louis J. Goldman and Vincent J. Scullin

Lewis Research Center

SUMMARY

A FORTRAN IV computer program for the design of supersonic blading based on establishing vortex flow within the blade passage is presented. The method of characteristics, as applied to the two-dimensional isentropic flow of a perfect gas, was utilized for the blade design. The equations necessary for the design are developed. The information required for the program consists of an inlet flow angle, specification of the inlet, outlet, and lower- and upper-circular-surface Mach numbers, and the specific-heat ratio. The program output consists of the blade coordinates and, if desired, a printer plot of the blade profile and flow passage. In addition, supersonic starting and flow separation calculations are performed by the program and obtained as output. An example is included to indicate the use of the program and the results obtainable.

INTRODUCTION

Supersonic compressors and turbines are employed in special circumstances because of their simplicity and low weight. A recent application for a supersonic turbine involves the hydrogen-fueled open-cycle auxiliary space power system described in reference 1. If the highest practical efficiency is to be obtained from supersonic compressors or turbines, proper design methods must be available.

A method for designing supersonic blade sections based on two-dimensional isentropic flow is given in reference 2. The method consists of converting the uniform parallel flow at the blade inlet into a vortex flow field, turning the vortex flow, and reconverting to a uniform parallel flow at the blade exit. The application of this design procedure involves specification of the inlet and outlet Mach numbers, the lower- (or concave) surface Mach number, the upper- (or convex) surface Mach number, the inlet flow
angle, and the specific-heat ratio of the working fluid. In general, a wide range of designs is possible by selection of these parameters. Guidance in the selection of a blade design is obtained by considering blade shape, solidity, and supersonic starting and flow separation problems. In reference 2 the effect of some of the design parameters for low Mach numbers and a specific-heat ratio of 1.4 is examined.

In view of the interest in hydrogen-fueled auxiliary space power systems, an analysis was conducted to gain a better understanding of the effects of the design parameters on the resulting blade geometry and to extend the results of reference 2 to levels of interest for such systems. In reference 3, the effect of surface Mach numbers, inlet flow angle, and specific-heat ratio on the geometric characteristics of supersonic impulse turbine-blade sections is investigated over an inlet Mach number range of 1.5 to 5.0. Blade design limitations resulting from supersonic starting and flow separation problems are also considered. In the present report, a description and a FORTRAN IV listing of a computer program for the design of blading applicable for any supersonic Mach number level and specific-heat ratio are presented. Supersonic starting and flow separation calculations are also performed by the program. An example is included to indicate the use of the program and the results obtainable. The report is organized so that those persons desiring to use the program need only read the sections METHOD OF ANALYSIS, DESCRIPTION OF INPUT, and DESCRIPTION OF OUTPUT. All necessary information pertaining to the program itself is contained in the sections DESCRIPTION OF INPUT, DESCRIPTION OF OUTPUT, and PROGRAM DESCRIPTION.

SYMBOLS

A area, ft² (m²)
a speed of sound, ft/sec (m/sec)
C reduction in maximum weight flow due to two-dimensional flow (eq. (34b))
C* dimensionless blade chord, chord/r*
f(R*) function defined by eq. (10b)
G* dimensionless blade spacing, spacing/r*
g_c dimensional conversion constant, 32.17 ft·lb/(lb)(sec²) (1 kg·m/(N)(sec²))
h blade height, ft (m)
j index for upper surface of blade
K* dimensionless vortex constant defined by eq. (23)
$K^*_\text{max}$ value of $K^*$ for which weight flow is maximum

$k$ index for lower surface of blade

$M$ Mach number, $V/a$

$M^*$ dimensionless velocity or critical velocity ratio, $V/V_{\text{cr}}$

$(M^*)_{\text{max}}^i$ maximum inlet velocity ratio for supersonic starting

$(M^*)_{\text{min}}^i$ minimum lower-surface velocity ratio from separation criterion

$(M^*)_{\text{max}}^u$ maximum upper-surface velocity ratio from separation criterion

$m$ slope of Mach line

$\overline{m}$ slope of wall segment

$p$ static pressure, lb/ft$^2$ (N/m$^2$)

$(p_t^i)_{\text{max}}$ maximum lower-surface static pressure from separation criterion

$Q$ vortex flow parameter (eq. 34a)

$R$ radius in vortex field, ft (m)

$R^*$ dimensionless radius in vortex field, $R/r^*$

$r^*$ radius of sonic velocity streamline in vortex field, ft (m)

$V$ velocity, ft/sec (m/sec)

$V_{\text{cr}}$ critical velocity, ft/sec (m/sec)

$w$ weight flow, lb/sec (kg/sec)

$w_{\text{max}}$ maximum weight flow, lb/sec (kg/sec)

$X^*$ dimensionless $X$-coordinate of blade (fig. I), $X/r^*$

$x^*$ dimensionless $x$-coordinate of transition arc (fig. I), $x/r^*$

$Y^*$ dimensionless $Y$-coordinate of blade (fig. I), $Y/r^*$

$y^*$ dimensionless $y$-coordinate of transition arc (fig. I), $y/r^*$

$\alpha$ circular arc turning angle, rad

$\beta$ flow angle outside blade passage, rad

$\gamma$ specific-heat ratio

$\theta$ total flow turning angle, rad

$\mu$ Mach angle, rad
METHOD OF ANALYSIS

Blade Description

The design of supersonic blade sections described herein is based on establishing vortex flow within the blade passage by a procedure analogous to that given in reference 2. The blade so designed consists essentially of three major parts: (1) inlet transition arcs, (2) circular arcs, and (3) outlet transition arcs. A typical blade, with pertinent nomenclature noted, is shown schematically in figure 1. The inlet transition arcs
Figure 1. - Typical supersonic blade section. (All coordinates are made dimensionless by dividing by $r^*$.)

(lower and upper) are required to convert the assumed uniform parallel flow at the blade inlet into vortex flow. The concentric circular arcs turn and maintain the vortex flow. Finally, the outlet transition arcs reconvert the vortex flow into uniform parallel flow at the blade exit. Straight-line segments parallel to the inlet and outlet flow directions complete the blade profile. Methods for creating a finite thickness at the leading and trailing edges are given in reference 2.

In general, the inlet lower transition arc reduces the Mach number from its value at the blade inlet $M_1$ to a preselected value of the lower-surface Mach number $M_{l1}$, whereas the inlet upper transition arc increases the Mach number to a preselected value...
of the upper-surface Mach number $M_u$. The surface Mach numbers remain constant, at these preselected values, on the lower and upper circular arcs. At the outlet region the procedure is reversed. The surface Mach number variation is shown in figure 2 for a typical blade.

The amount of flow turning produced by either the lower or upper surface of the blade consists, in general, of two parts (fig. 1): (1) the turning produced by the transition arcs and (2) the turning produced by the circular arcs. When isentropic flow turning at supersonic speeds is considered, it is convenient to introduce the Prandtl-Meyer angle $\nu$, which is defined as the angle through which the flow must turn from Mach 1 to the required Mach number. The flow turning produced by a transition arc is then equal to differences in Prandtl-Meyer angles and is $\nu_l - \nu_i$ and $\nu_o - \nu_l$ for the inlet and outlet lower transition arcs and $\nu_u - \nu_i$ and $\nu_u - \nu_o$ for the inlet and outlet upper transition arcs, respectively. The turning produced by the inlet or outlet transition arcs cannot exceed the inlet or outlet flow angle $\beta_i$ or $\beta_o$, respectively. The relation between Prandtl-Meyer angle $\nu$ and Mach number $M$ is given by the following equations (ref. 4):
\[ \nu = \frac{\pi}{4} \left( \sqrt{\frac{\gamma + 1}{\gamma - 1}} - 1 \right) + \frac{1}{2} \left\{ \sqrt{\frac{\gamma + 1}{\gamma - 1}} \arc\sin \left( \frac{(\gamma - 1)M^*^2}{\sqrt{(\gamma + 1)} - \gamma} \right) + \arc\sin (\gamma + 1) \right\} \]

(1)

where the dimensionless velocity or critical velocity ratio \( M^* \) is

\[ M^* = \left( \frac{\gamma + 1}{2} \frac{M^2}{1 + \frac{\gamma - 1}{2} M^2} \right)^{1/2} \]

(2)

This relation is shown in figure 3 for specific-heat ratios of 1.3, 1.4, and 1.66. As the Mach number approaches infinity, \( \nu \) approaches an upper limit of \( \frac{\pi}{2} \left[ \sqrt{(\gamma + 1)/(\gamma - 1)} - 1 \right] \).

Figure 3. - Variation of Prandtl-Meyer angle with Mach number for different specific-heat ratios.
Blade Design

The method of characteristics as applied to the two-dimensional isentropic flow of a perfect gas is utilized in the design of the supersonic blade sections. A description of the method of characteristics is given in references 4 and 5, and its application to the design of supersonic blade sections is given in references 2 and 6. For purposes of calculation, the flow field is considered to be divided into small regions, in each of which the flow properties are assumed to be constant. If adjacent regions are to differ slightly in properties, then the boundary between the regions must be characteristic lines and can also be shown (ref. 5) to be Mach lines. Therefore, each region is, in general, bounded either by a Mach line or a physical boundary. In figure 4 the flow field for a typical blade passage is divided by characteristic lines into a finite number of regions. The vortex-flow region is bounded by the circular arcs and the outermost vortex characteristics AE and CE. The transition arcs are composed of straight-line segments, and within each region bounded by these segments the flow is constrained to follow the wall direction. The mathematical equations necessary to define the blade are developed in the following sections.

Circular arcs. - Within the concentric circular arcs, vortex flow exists; therefore,
\[ VR = \text{Constant} \quad (3) \]

where \( V \) is the velocity and \( R \) is the radius in the vortex field. In this report, dimensionless parameters are used whenever possible; if this procedure is followed, equation (3) can be rewritten as

\[
\left( \frac{V}{V_{cr}} \right) \left( \frac{R}{r^*} \right) = \frac{\text{Constant}}{V_{cr} r^*}
\]

where \( V_{cr} \) is the critical velocity and \( r^* \) is the radius of the sonic velocity streamline in the vortex field. At \( R = r^* \), \( V = V_{cr} \); therefore, the constant is \( V_{cr} r^* \). Equation (4) then becomes

\[
M^* R^* = 1 \quad (5)
\]

where \( M^* = V/V_{cr} \) is the dimensionless velocity and \( R^* = R/r^* \) is the dimensionless radius in the vortex field. The Prandtl-Meyer angle \( \nu \) is related to \( M^* \) through equation (1). Therefore, once \( \nu_i \) and \( \nu_u \) are specified, \( M^*_l \) and \( M^*_u \) are fixed, and the circular arc radii \( R^*_l \) and \( R^*_u \) are determined from equation (5). The amount of lower circular arc turning for the inlet and outlet portions of the blade \( \alpha_{l,i} \) and \( \alpha_{l,o} \) respectively, are

\[
\alpha_{l,i} = \beta_i - (\nu_i - \nu_l) \quad (6a)
\]

and

\[
\alpha_{l,o} = \beta_o + (\nu_o - \nu_l) \quad (6b)
\]

Angles measured in the counterclockwise direction are considered positive. With this convention, the inlet flow angle \( \beta_i \) is positive, and the outlet flow angle \( \beta_o \) is negative. Similarly, for the upper circular arc

\[
\alpha_{u,i} = \beta_i - (\nu_i - \nu_l) \quad (7a)
\]

and

\[
\alpha_{u,o} = \beta_o + (\nu_u - \nu_o) \quad (7b)
\]

The circular arcs are completely described by specification of \( \nu_i \), \( \nu_o \), \( \nu_l \), \( \nu_u \), and \( \beta_i \).
The outlet flow angle $\beta_o$ does not have to be specified because it can be related to $M_1$, $M_o$, and $\beta_i$ from the following consideration.

Because the inlet and outlet blade spacing is the same, the inlet and outlet blade passage areas $A_i$ and $A_o$, respectively, are related by geometry (see fig. 1) according to the equation

$$\frac{A_i}{A_o} = \frac{\cos \beta_i}{\cos \beta_o} \quad \text{(8)}$$

The area ratio $A_i/A_o$ can be obtained from the continuity equation (ref. 5) with the result that equation (8) becomes

$$\beta_o = -\arccos \left\{ \left[ \frac{M_1}{M_o} \left( \frac{1 + \frac{y-1}{2} \frac{M_o^2}{M_1^2}}{2} \right)^{(y+1)/(2(y-1))} \right] \cos \beta_i \right\} \quad \text{(9)}$$

**Lower transition arcs.** - The lower surface is composed of an inlet and outlet transition arc and a circular arc (fig. 1). For symmetric blades (i.e., $\nu_i = \nu_o$), the two transition arcs are identical, and, therefore, only one needs to be calculated. For asymmetric blades, the two arcs are not identical, one being smaller (less turning) than the other. However, the smaller transition arc corresponds to a portion of the larger arc, and, consequently, only the larger arc need be calculated. If $\nu_i$ is greater than $\nu_o$, the inlet transition arc is the larger of the two arcs. For simplicity, the inlet transition arc is assumed to be the larger arc in the following discussion.

In figure 5, the lower transition arc is shown, and the nomenclature used in the computer program is indicated. The calculations are performed with respect to the nondimensional axes $x^*$ and $y^*$ (where the $x$- and $y$-coordinates are made dimensionless by dividing by $r^*$). The transition arc coordinates are generated in a sequential manner (starting at $x^* = 0$, $y^* = R_i^*$) by obtaining the intersection of the straight-line wall segments and straight Mach lines for a specified small change in flow turning. The Mach lines are determined from the outermost or major vortex-expansion characteristic, and the wall segments are determined from the flow direction. After the transition arc coordinates are calculated, they are rotated through an angle of $\alpha_{l,i}$ to obtain the coordinates of interest in the blade design (see fig. 1).

For the vortex region, it can be shown (ref. 2) that the velocity direction $\phi$ and the dimensionless radius $R^*$ are related along the characteristic line by the equation
Figure 5. - Nomenclature used for calculation of inlet lower transition arc.
\[ \varphi = \pm \frac{1}{2} f(R^*) + \text{Constant} \quad (10a) \]

where

\[ f(R^*) = \sqrt{\frac{\gamma + 1}{\gamma - 1}} \arcsin\left(\frac{\gamma - 1}{R^{*2}} - \gamma\right) + \arcsin\left[(\gamma + 1)R^{*2} - \gamma\right] \quad (10b) \]

Two families of characteristics exist: the positive sign in equation (10a) gives the expansion lines, and the negative sign gives the compression lines. The major vortex-expansion-characteristic equation is

\[ \varphi = \frac{1}{2} \left[ f(R^*) - f(R^*_l) \right] \quad (11) \]

since the boundary condition at \( x^* = 0 \) is that \( \varphi = 0 \) and \( R^* = R^*_l \). If the flow field is divided into small regions, the flow direction at any point along the characteristic line (as given by eq. (11)) may be considered to be equal to the flow direction within the adjacent flow region, as indicated in figure 5. For \( k \) transition arc segments, each of which produces \( \Delta \nu \) degrees of turning, the flow direction within any flow segment \( \varphi_{k,i} \) is given by

\[ \varphi_{k,i} = \nu_i - \nu_l - (k - 1)\Delta \nu = \varphi_{k+1,1} + \Delta \nu \quad (12) \]

where \( k \) is an integer that varies from 1 to \( \left[ (\nu_i - \nu_l)/\Delta \nu \right] + 1 \). At \( k = \left[ (\nu_i - \nu_l)/\Delta \nu \right] + 1 \), the flow direction is 0, and at \( k = 1 \), it is \( \nu_i - \nu_l \). Equating equations (11) and (12) and eliminating \( f(R^*_l) \) through equations (1), (5), and (10b) result in

\[ f(R^*_{k,i}) = 2\nu_i - \frac{\pi}{2} \left( \sqrt{\frac{\gamma + 1}{\gamma - 1}} - 1 \right) - 2(k - 1)\Delta \nu \quad (13) \]

which relates the dimensionless radius \( R^* \) to the incremental flow turning \( \Delta \nu \) along the major vortex-expansion characteristic. At \( k = \left[ (\nu_i - \nu_l)/\Delta \nu \right] + 1 \), \( R^*_{k,i} = R^*_l \); as \( k \) is decreased, \( R^*_{k,i} \) decreases and is obtained by the simultaneous solution of equations (10b) and (13) by an iterative procedure. Since \( R^*_{k,i} \) can be determined for any value of \( k \), the \( x^*, y^* \) coordinates along the major expansion characteristic can be obtained from...
\begin{align}
x_{k, i}^* &= -R_{k, i}^* \sin \phi_{k, i} \tag{14a} \\
y_{k, i}^* &= R_{k, i}^* \cos \phi_{k, i} \tag{14b}
\end{align}

These coordinates are also points on the straight Mach lines. The equation specifying the Mach line is therefore determined once the slope is obtained, which is easily accomplished because the Mach line is inclined at the Mach angle $\mu$ to the velocity direction (see fig. 5). The Mach line, therefore, makes an angle of $\phi + \mu$ with respect to the $x^*$ axis. It is possible to define the Mach line at the mean Mach angle to the mean flow direction (ref. 5) so that the slope of the Mach line $m_{k, i}$ is given by

\begin{equation}
m_{k, i} = \tan \left( \frac{\phi_{k, i} + \phi_{k+1, i} + \mu_{k, i} + \mu_{k+1, i}}{2} \right) \tag{15a}
\end{equation}

where

\begin{equation}
\mu_{k, i} = -\arcsin \left( \frac{1}{M_{k, i}} \right) = -\arcsin \left[ \sqrt{\left(\frac{\gamma + 1}{2}\right) \frac{M_{k, i}^2}{\rho_{k, i}^* - \left(\frac{\gamma - 1}{2}\right)}} \right] \tag{15b}
\end{equation}

The equation of the Mach line is therefore

\begin{equation}
y^* = m_{k, i} (x^* - x_{k, i}^*) + y_{k, i}^* \tag{16}
\end{equation}

where $k$ varies from 1 to $k_{\text{max}} = (\nu_1 - \nu_\ell)/\Delta \nu$. The equation of the transition arc segment is more easily determined since each segment is a straight line parallel to the velocity direction $\phi$. The slope of the wall segment $\overline{m}_{k, i}$ is then

\begin{equation}
\overline{m}_{k, i} = \tan \phi_{k+1, i} \tag{17}
\end{equation}

and the equation for the wall segment is

\begin{equation}
y^* = \overline{m}_{k, i} \left[ x^* - (x_{k, i}^*)_{k+1, i} \right] + (y_{\ell}^*)_{k+1, i} \tag{18}
\end{equation}
where $k$ varies from 1 to $k_{\text{max}}$ and $x^*_l$ and $y^*_l$ are the lower transition arc coordinates. The values of $x^*_l$ and $y^*_l$ are known at $k = (v_1 - \nu_l)/\Delta \nu + 1 = k_{\text{max}} + 1$, where $x^*_l = 0$ and $y^*_l = R^*_l$. The remaining transition coordinates are generated by finding the intersection of the Mach lines with the wall segments starting at $k = k_{\text{max}}$ and sequentially decreasing $k$ until $k = 1$. The intersection of the two straight lines is given by

$$
(x^*_l)_{k, i} = \left[ (y^*_l)_{k+1, i} - \frac{\bar{m}_{ki}}{m_{ki}} \left( (x^*_l)_{k+1, i} - (y^*_l)_{k+1, i} \right) \right] - \frac{(y^*_l)_{k, i} - m_{ki} (x^*_l)_{k, i}}{m_{ki} - \bar{m}_{ki}}
$$

(19a)

and

$$
(y^*_l)_{k, i} = \frac{m_{ki} \left[ (y^*_l)_{k+1, i} - \bar{m}_{ki} \left( (x^*_l)_{k+1, i} \right) \right] - \bar{m}_{ki} \left( (y^*_l)_{k, i} - m_{ki} (x^*_l)_{k, i} \right)}{m_{ki} - \bar{m}_{ki}}
$$

(19b)

The transition arc coordinates obtained from equation (19) are rotated through an angle $\alpha^*_l, i$ resulting in the $X^*, Y^*$ coordinates of interest in the blade design. The rotated coordinates $X^*_l$ and $Y^*_l$ are obtained from

$$
(X^*_l)_{k, i} = (x^*_l)_{k, i} \cos \alpha^*_l, i - (y^*_l)_{k, i} \sin \alpha^*_l, i
$$

(20a)

and

$$
(Y^*_l)_{k, i} = (x^*_l)_{k, i} \sin \alpha^*_l, i + (y^*_l)_{k, i} \cos \alpha^*_l, i
$$

(20b)

Upper transition arcs. - The upper surface (like the lower surface) is composed, in part, of an inlet and outlet transition arc, only one of which must be calculated. For simplicity, it is again assumed that the inlet transition arc is the larger of the two arcs. (For the upper arc, this requires that $\nu_0$ be greater than $\nu_1$.) The upper transition arc is shown schematically in figure 6, and the pertinent nomenclature is noted. The procedure employed to calculate the upper transition arc is analogous to that used for the lower transition arc; the resulting equations which are of similar form are not repeated herein. The subscript $j$ is used to represent the upper transition arc coordinates where $j$ varies from 1 to $j_{\text{max}} = (\nu_u - \nu_1)/\Delta \nu$.

Geometric parameters. - After the blade calculations have been performed, a number of blade parameters of interest, including blade solidity, spacing, chord, and total
Figure 6. - Nomenclature used for calculation of inlet upper transition arc.
flow turning angle, are calculated. The blade spacing $G^*$ and chord $C^*$ are obtained from the blade coordinates, and the solidity is obtained from the ratio $C^*/G^*$. The total flow turning angle $\theta$ is obtained from the inlet and outlet flow angles. The blade coordinates are also translated by $G^*$ so that coordinates for two complete blades are obtained.

Design Limitations

The design limitations (i.e., the constraints on the choice of $v_l$ and $v_u$ for specified $v_1^*$, $v_0^*$, $\beta_1^*$, and $\gamma$) imposed by consideration of supersonic starting and flow separation problems have been discussed in reference 3. These limitations are calculated by the procedure described in the following paragraphs and are given as output from the computer program presented herein.

Supersonic starting. - The problem of establishing supersonic flow on startup is discussed in reference 7 for supersonic compressors. For supersonic turbines, the resulting design limitations due to starting are presented in reference 2, where it is assumed that a normal shock wave spans the blade inlet at the instant of startup. Under this condition it is necessary to ensure that the weight flow can pass through the turbine. The maximum value of the inlet Prandtl-Meyer angle $(\nu_1^*)_{\text{max}}$ is determined by first finding the maximum weight flow through the blade passage, while taking into account the normal shock losses. This maximum weight flow is then equated to the flow rate after the shock has passed through the passage.

The weight flow through the passage is obtained by integrating the continuity equation in the vortex region

$$w = \rho \int_{R_u}^{R_l} \rho V \, dR$$

(21)

where $\rho$ is the density, $V$ is the velocity, and $h$ is the blade height. The density can be written as (ref. 2)

$$\rho = \rho_1^* \left[ 1 - \left( \frac{\gamma - 1}{2} \right) \left( \frac{V}{a_1^*, d} \right)^2 \right]^{1/(\gamma - 1)}$$

(22)

where $\rho_1^*, d$ and $a_1^*, d$ are the density and the speed of sound, respectively, just downstream of the shock, and are evaluated at total conditions. For perfect gases, the total
temperature is constant through a normal shock so that \( a'_{i,d} = a'_i \). Utilizing equation (3) and the definition

\[
K^* = \sqrt{\frac{\gamma - 1}{2}} \left( \frac{VR}{a'_i R_l} \right)
\]  

results in equation (21) in the following form:

\[
w = \frac{a'_1 \rho'_1}{d} R_l \sqrt{\frac{2}{\gamma - 1}} \int_{R_u}^{R_l} \left( 1 - \frac{K^* R^2}{R_l^2} \right)^{1/(\gamma - 1)} \frac{K^*}{R} dR
\]  

(24)

Differentiating equation (24) with respect to \( K^* \) and setting the result equal to 0 gives the value of \( K^* \) (denoted as \( K^*_{\text{max}} \)) for which the weight flow is a maximum. This procedure gives

\[
\frac{d}{dK^*} \left( \frac{2}{\gamma - 1} \int_{R_u}^{R_l} \left( 1 - \frac{K^* R^2}{R_l^2} \right)^{1/(\gamma - 1)} \frac{K^*}{R} dR \right) = 0
\]

(25)

Changing the variable from \( R \) to \( M^* \) gives, for equation (25),
Integrating the right side of equation (26) results in

\[ \int_{M_l^*}^{M_u^*} \left[ 1 - \left( \frac{K_{\text{max}}}{M_l^*} \right)^2 M^{**2} \right]^{1/(\gamma-1)} \frac{dM^*}{M^*} = \frac{2}{\gamma - 1} \int_{M_l^*}^{M_u^*} \left( \frac{K_{\text{max}}}{M_l^*} \right)^2 \left[ 1 - \left( \frac{K_{\text{max}}}{M_l^*} \right)^2 M^{**2} \right]^{(2-\gamma)/(\gamma-1)} \frac{M^* \, dM^*}{M^*} \] (27)

Similarly, equation (24) in terms of \( M^* \) becomes

\[ w_{\text{max}} = r^* h a_i^* \rho_i^* d \sqrt{\frac{2}{\gamma - 1}} \int_{M_l^*}^{M_u^*} \frac{K_{\text{max}}}{M_l^*} \left[ 1 - \left( \frac{K_{\text{max}}}{M_l^*} \right)^2 M^{**2} \right]^{1/(\gamma-1)} \frac{dM^*}{M^*} \] (28)

The value of \( K_{\text{max}}^* \) is determined from equation (27) by an iterative procedure (for given values of \( M_l^* \) and \( M_u^* \)), and then the maximum weight flow is obtained from equation (28). After the shock has passed through the turbine, the weight flow can again be obtained from equation (21), which is rewritten by substituting \( VR = V_{\text{cr}} r^* \) and changing the variable from \( R \) to \( M^* \) to give

\[ w = r^* h \int_{M_l^*}^{M_u^*} \rho V_{\text{cr}} \frac{dM^*}{M^*} \] (29)
Substituting the following relations (ref. 5) into equation (29)

\[ \rho = \rho_1 \left( \frac{2}{\gamma + 1} \right)^{1/(\gamma - 1)} \left( \frac{\gamma + 1}{2} - \frac{\gamma - 1}{2} M^* \right)^{1/(\gamma - 1)} \]  

(30)

and

\[ V_{cr} = a_1' \sqrt{\frac{2}{\gamma + 1}} \]  

(31)

gives

\[ w = r^* h a_1' \rho_1' \sqrt{\frac{2}{\gamma + 1}} \left( \frac{2}{\gamma + 1} \right)^{1/(\gamma - 1)} \int_{M'_l}^{M_u} \left( \frac{\gamma + 1}{2} - \frac{\gamma - 1}{2} M^* \right)^{1/(\gamma - 1)} \frac{dM^*}{M^*} \]  

(32)

where the term multiplying \( r^* h a_1' \rho_1' \) is defined as the weight-flow parameter. Equating the two weight flows, equations (28) and (32), results in

\[ \frac{Q}{1 - C} = \frac{\rho_1' d}{\rho_1' d} = \frac{p_1', d}{p_1'} \]  

(33)

where

\[ Q = \frac{M_u^* M_u}{M_u^* - M_l^*} \int_{M_l^*}^{M_u^*} \left( \frac{\gamma + 1}{2} - \frac{\gamma - 1}{2} M^* \right)^{1/(\gamma - 1)} \frac{dM^*}{M^*} \]  

(34a)

and

\[ C = 1 - \sqrt{\frac{\gamma + 1}{\gamma - 1} \left( \frac{\gamma + 1}{2} \right)^{1/(\gamma - 1)} \left( \frac{M_u^*}{M_u^* - M_l^*} \right) \int_{M_l^*}^{M_u^*} K_{max}^* \left[ 1 - \left( \frac{K_{max}^*}{M^*} \right)^2 \right]^{1/(\gamma - 1)} \frac{dM^*}{M^*}} \]  

(34b)
The parameter \( C \) has been shown in reference 2 to be the reduction in maximum flow rate caused by two-dimensional flow. The quantity \( p'_1, d/p'_1 \) is the total pressure recovery for a normal shock and is given by (ref. 5)

\[
\frac{p'_1, d}{p'_1} = \left( M'^*_1 \right)_{\text{max}}^{2}\frac{1 - \left( \frac{\gamma - 1}{\gamma + 1} \right) \left( M'^*_1 \right)_{\text{max}}^{2}}{\left( M'^*_1 \right)_{\text{max}}^{2} - \frac{\gamma - 1}{\gamma + 1}} 1/(\gamma - 1)
\]

(35)

The maximum inlet velocity ratio \((M^*_l)_{\text{max}}\) is obtained (for given values of \( M^*_u \) and \( M^*_l \)) by simultaneous solution of equations (33) and (35), by using the definition in equations (34a) and (34b), and by obtaining \( K^*_{\text{max}} \) from equation (27). The maximum inlet Prandtl-Meyer angle for supersonic starting \((\psi^*_1)_{\text{max}}\) is then obtained from equation (1).

Flow separation. - Analysis of supersonic blade sections (ref. 3) has shown the desirability of maintaining high surface Mach numbers to alleviate the problem of supersonic starting. Under these conditions, however, adverse pressure gradients created on the blade surfaces would be expected to cause flow separation and, consequently, poor performance. Experimental investigation of simple shapes with incompressible flow at fairly high pressure gradients (ref. 8) has indicated that if the coefficient of pressure recovery (defined as the ratio of the pressure rise to the dynamic pressure at the initial point) is less than about 1/2, flow separation may be avoided. This criterion was used in the analysis presented in reference 3 to give some indication of the design restrictions due to flow separation. For supersonic velocities the separation value of the coefficient of pressure recovery may be less than 1/2. The calculational procedure is as follows.

Flow separation can occur on both the lower and upper surfaces of the blade, but since the calculational procedure is similar for both cases only the derivation for the lower surface is presented. The flow separation criterion can be written as

\[
\frac{(p_l)}{\text{max}} = \frac{p_i}{\rho_i V_i^2 / 2 g_c} = \frac{1}{2 (p'_l)_{\text{max}}}
\]

(36)

where \((p_l)_{\text{max}}\) is the maximum lower-surface pressure possible (for given inlet conditions) without causing separation. Equation (36) can be rewritten in the form
\[
\frac{(p_i)^{\text{max}}}{p_i} = \frac{\rho_i V_i^2}{2g_c p_i} \left[ 1 + \frac{1}{2} \left( \frac{1}{\gamma - 1} \right) \right]
\]  

(37)

where the equation has been divided by the inlet total pressure \( p_i \). Substituting the following relations (ref. 9) into equation (37)

\[
\frac{p}{p'} = \left( 1 - \frac{\gamma - 1}{\gamma + 1} M^{*2} \right)^{\gamma/(\gamma - 1)}
\]

(38a)

and

\[
\frac{1}{2g_c} \rho V^2 = \frac{\gamma}{\gamma + 1} M^{*2} \left( 1 - \frac{\gamma - 1}{\gamma + 1} M^{*2} \right)
\]

(38b)

and simplifying result in

\[
(M^{*})_{\text{min}} = \left( \frac{\gamma + 1}{\gamma - 1} \right)^{1/2} \left[ 1 - \left( \frac{1 - \frac{\gamma - 1}{\gamma + 1} M^{*2}}{1 - \frac{\gamma - 1}{\gamma + 1} M^{*2}} \right)^{\gamma/(\gamma - 1)/\gamma - 1} \right]^{1/2}
\]

(39)

Similarly, applying the same criterion to the upper surfaces gives

\[
M^{*} = \left( \frac{\gamma + 1}{\gamma - 1} \right)^{1/2} \left[ 1 - \left( \frac{1 - \frac{\gamma - 1}{\gamma + 1} (M^{*})^2_{\text{max}}}{1 - \frac{\gamma - 1}{\gamma + 1} (M^{*})^2_{\text{max}}} \right)^{\gamma/(\gamma - 1)/\gamma - 1} \right]^{1/2}
\]

(40)

The corresponding Prandtl-Meyer angles \( (\nu)_{\text{min}} \) and \( (\nu)_{\text{max}} \) are obtained from equation (1).
## TABLE I - INPUT FORMAT FOR FORTRAN COMPUTER PROGRAM

<table>
<thead>
<tr>
<th>STATEMENT NUMBER</th>
<th>FORTRAN STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
</tr>
<tr>
<td>2</td>
<td>V</td>
</tr>
<tr>
<td>3</td>
<td>U</td>
</tr>
<tr>
<td>4</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
</tr>
<tr>
<td>8</td>
<td>V</td>
</tr>
<tr>
<td>9</td>
<td>M</td>
</tr>
<tr>
<td>10</td>
<td>T</td>
</tr>
</tbody>
</table>

### FORTRAN STATEMENT

<table>
<thead>
<tr>
<th>STATEMENT NUMBER</th>
<th>FORTRAN STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
</tr>
<tr>
<td>2</td>
<td>V</td>
</tr>
<tr>
<td>3</td>
<td>U</td>
</tr>
<tr>
<td>4</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
</tr>
<tr>
<td>8</td>
<td>V</td>
</tr>
<tr>
<td>9</td>
<td>M</td>
</tr>
<tr>
<td>10</td>
<td>T</td>
</tr>
</tbody>
</table>

### FORTRAN STATEMENT

<table>
<thead>
<tr>
<th>STATEMENT NUMBER</th>
<th>FORTRAN STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
</tr>
<tr>
<td>2</td>
<td>V</td>
</tr>
<tr>
<td>3</td>
<td>U</td>
</tr>
<tr>
<td>4</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
</tr>
<tr>
<td>8</td>
<td>V</td>
</tr>
<tr>
<td>9</td>
<td>M</td>
</tr>
<tr>
<td>10</td>
<td>T</td>
</tr>
</tbody>
</table>

### FORTRAN STATEMENT

<table>
<thead>
<tr>
<th>STATEMENT NUMBER</th>
<th>FORTRAN STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
</tr>
<tr>
<td>2</td>
<td>V</td>
</tr>
<tr>
<td>3</td>
<td>U</td>
</tr>
<tr>
<td>4</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
</tr>
<tr>
<td>8</td>
<td>V</td>
</tr>
<tr>
<td>9</td>
<td>M</td>
</tr>
<tr>
<td>10</td>
<td>T</td>
</tr>
</tbody>
</table>

### FORTRAN STATEMENT

<table>
<thead>
<tr>
<th>STATEMENT NUMBER</th>
<th>FORTRAN STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
</tr>
<tr>
<td>2</td>
<td>V</td>
</tr>
<tr>
<td>3</td>
<td>U</td>
</tr>
<tr>
<td>4</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
</tr>
<tr>
<td>8</td>
<td>V</td>
</tr>
<tr>
<td>9</td>
<td>M</td>
</tr>
<tr>
<td>10</td>
<td>T</td>
</tr>
</tbody>
</table>

### FORTRAN STATEMENT

<table>
<thead>
<tr>
<th>STATEMENT NUMBER</th>
<th>FORTRAN STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
</tr>
<tr>
<td>2</td>
<td>V</td>
</tr>
<tr>
<td>3</td>
<td>U</td>
</tr>
<tr>
<td>4</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
</tr>
<tr>
<td>8</td>
<td>V</td>
</tr>
<tr>
<td>9</td>
<td>M</td>
</tr>
<tr>
<td>10</td>
<td>T</td>
</tr>
</tbody>
</table>

### FORTRAN STATEMENT

<table>
<thead>
<tr>
<th>STATEMENT NUMBER</th>
<th>FORTRAN STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
</tr>
<tr>
<td>2</td>
<td>V</td>
</tr>
<tr>
<td>3</td>
<td>U</td>
</tr>
<tr>
<td>4</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
</tr>
<tr>
<td>8</td>
<td>V</td>
</tr>
<tr>
<td>9</td>
<td>M</td>
</tr>
<tr>
<td>10</td>
<td>T</td>
</tr>
</tbody>
</table>

### FORTRAN STATEMENT

<table>
<thead>
<tr>
<th>STATEMENT NUMBER</th>
<th>FORTRAN STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
</tr>
<tr>
<td>2</td>
<td>V</td>
</tr>
<tr>
<td>3</td>
<td>U</td>
</tr>
<tr>
<td>4</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>O</td>
</tr>
<tr>
<td>8</td>
<td>V</td>
</tr>
<tr>
<td>9</td>
<td>M</td>
</tr>
<tr>
<td>10</td>
<td>T</td>
</tr>
</tbody>
</table>
DESCRIPTION OF INPUT

The input for the computer program consists of an inlet flow angle, several Prandtl-Meyer angles, the specific-heat ratio $\gamma$, and an angular increment. In addition, three optional switches must be set which regulate the content and form of the output. All the input parameters, except $\gamma$ and the switches, must be specified in degrees. The input variables are as follows:

- **BETAN**: inlet flow angle, $\beta_i$
- **DELV**: flow turning increment (recommended value, 0.1), $\Delta \nu$
- **GAM**: specific-heat ratio, $\gamma$
- **IPRINT**: a value of 0 will result in printing of rotated blade coordinates $X^*$ and $Y^*$; a value of 1 will cause both rotated and unrotated coordinates $X^*, Y^*$ and $x^*, y^*$ (see fig. 1) to be printed
- **ISTART**: a value of 0 will cause both starting and blade design calculations to be printed out; a value of 1 will cause only starting calculations to be performed and printed out
- **NPLOT**: a value of 0 will cause blade profile and flow passage to be plotted; a value of 1 will suppress the plot
- **VIN**: inlet Prandtl-Meyer angle, $\nu_i$
- **VLOW**: lower-surface Prandtl-Meyer angle, $\nu_l$
- **VOUT**: outlet Prandtl-Meyer angle, $\nu_o$
- **VUP**: upper-surface Prandtl-Meyer angle, $\nu_u$

The flow turning increment $\Delta \nu$ must be specified so that $(\nu_i - \nu_l)/\Delta \nu$, $(\nu_o - \nu_l)/\Delta \nu$, $(\nu_u - \nu_i)/\Delta \nu$, and $(\nu_u - \nu_o)/\Delta \nu$ are all integers. Table I shows a sample input card.

DESCRIPTION OF OUTPUT

An example of the output obtained from the program is shown in table II. The output corresponds to the input data shown in table I and consists of tables of coordinates for the description of two blade sections, supersonic starting and flow separation parameters, and a plot of the blade profile and flow passage. This example required approximately 0.2 minute of computer running time. Each section of the output has been numbered to correspond to the following description:

1. The first output of the program is a listing of the supersonic starting parameters.
If ISTART=0, the program will continue with the blade design calculations. If ISTART=1, no further output is obtained.

(2) The next output is a listing of all the input data plus the value of the calculated outlet flow angle.

(3) If IPRINT=1, the following tables are printed:
   (a) Unrotated coordinates of the lower- and upper-surface transition arcs
   (b) Coordinates of the lower- and upper-surface circular arcs
   (c) Coordinates of the upper-surface straight-line segments

This output is, in general, not of interest except for debugging purposes and may be omitted by setting IPRINT=0.

(4) The next output is tables of the rotated coordinates of the lower and upper transition arcs. In addition, these coordinates are translated by the value of the blade spacing so that coordinates for two blade sections are obtained. Every tenth calculation point is printed if $\Delta \nu < 0.2^\circ$; otherwise every calculation point is printed.

(5) The next output is a listing of miscellaneous parameters including
   (a) Inlet, outlet, and surface dimensionless velocities or critical velocity ratios and Mach numbers
   (b) Dimensionless blade spacing, chord and solidity
   (c) Separation limitation for the lower- and upper-surface Prandtl-Meyer angles

(6) If NPLOT=0, the final output is a printer plot of the blade profile and the flow passage. If NPLOT=1, this output is omitted.
### TABLE II - SAMPLE OUTPUT

#### DESIGN OF SUPERSONIC PLACES

**C A L C U L A T I O N S F O R S U P E R S O N I C S T A R T I N G**

<table>
<thead>
<tr>
<th>x(max) = 4.3</th>
<th>C = 0.0439</th>
<th>Q = 0.4701</th>
<th>( \text{MAXIMA} ) = 3.8449</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{THE MAXIMUM DESIGN VALUE FOR VELOCITY IS} )</td>
<td>40,7900 DEG</td>
<td>WHEN VELOCITY IS</td>
<td>19.0000 DEG</td>
</tr>
</tbody>
</table>

**DESIGN PARAMETERS**

1. \( \text{DELTA} = 35.6000 \text{ DEG} \)
2. \( \text{VELOCITY} = 59.0000 \text{ DEG} \)
3. \( \text{DELTA} = 35.6000 \text{ DEG} \)
4. \( \text{VELOCITY} = 28.0000 \text{ DEG} \)

**LUMER SURFACE**

**UNIQUAL TRANSITION ARC**

<table>
<thead>
<tr>
<th>( \text{INPUT} )</th>
<th>( \text{X(NUM)} )</th>
<th>( \text{Y(NUML)} )</th>
<th>( \text{Y(NUML)} )</th>
<th>( \text{Y(NUML)} )</th>
<th>( \text{OUTLET X} )</th>
<th>( \text{Y(NUML)} )</th>
<th>( \text{Y(NUML)} )</th>
<th>( \text{Y(NUML)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>0.7</td>
<td>0.0796</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201</td>
<td>0.8796</td>
<td>0.0796</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>251</td>
<td>0.8796</td>
<td>0.0796</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>301</td>
<td>0.8796</td>
<td>0.0796</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>351</td>
<td>0.8796</td>
<td>0.0796</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>401</td>
<td>0.8796</td>
<td>0.0796</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>451</td>
<td>0.8796</td>
<td>0.0796</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>501</td>
<td>0.8796</td>
<td>0.0796</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>551</td>
<td>0.8796</td>
<td>0.0796</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>601</td>
<td>0.8796</td>
<td>0.0796</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**KURTATEL AND TRANSLATED TRANSITION ARC**

| \( \text{INPUT} \) | \( \text{X(NUM)} \) | \( \text{Y(NUML)} \) | \( \text{Y(NUML)} \) | \( \text{Y(NUML)} \) | \( \text{Y(NUML)} \) | \( \text{Y(NUML)} \) | \( \text{Y(NUML)} \) | \( \text{Y(NUML)} \) | \( \text{OUTLET X} \) | \( \text{Y(NUML)} \) | \( \text{Y(NUML)} \) | \( \text{Y(NUML)} \) | \( \text{Y(NUML)} \) | \( \text{OUTLET X} \) | \( \text{Y(NUML)} \) | \( \text{Y(NUML)} \) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 701 | -0.0726 | 0.0404 | -0.0726 | 0.0404 |
| 751 | -0.0726 | 0.0404 | -0.0726 | 0.0404 |
| 801 | -0.0726 | 0.0404 | -0.0726 | 0.0404 |
| 851 | -0.0726 | 0.0404 | -0.0726 | 0.0404 |
| 901 | -0.0726 | 0.0404 | -0.0726 | 0.0404 |
| 951 | -0.0726 | 0.0404 | -0.0726 | 0.0404 |
| 1001 | -0.0726 | 0.0404 | -0.0726 | 0.0404 |
| 1051 | -0.0726 | 0.0404 | -0.0726 | 0.0404 |
| 1101 | -0.0726 | 0.0404 | -0.0726 | 0.0404 |
| 1151 | -0.0726 | 0.0404 | -0.0726 | 0.0404 |

---

25
<table>
<thead>
<tr>
<th>$\text{Y}^{(1)}(L_n)$</th>
<th>$\text{X}^{(1)}(L_n)$</th>
<th>$\text{Y}^{(3)}(L_n)$</th>
<th>$\text{X}^{(3)}(L_n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.0015</td>
<td>-0.0015</td>
<td>-0.0015</td>
<td>-0.0015</td>
</tr>
<tr>
<td>-0.0020</td>
<td>-0.0020</td>
<td>-0.0020</td>
<td>-0.0020</td>
</tr>
<tr>
<td>-0.0025</td>
<td>-0.0025</td>
<td>-0.0025</td>
<td>-0.0025</td>
</tr>
<tr>
<td>-0.0030</td>
<td>-0.0030</td>
<td>-0.0030</td>
<td>-0.0030</td>
</tr>
</tbody>
</table>

*Note: Additional data not shown.*
### TABLE II. - Continued. SAMPLE OUTPUT

#### UPPER SURFACE

**UNIT 1**

<table>
<thead>
<tr>
<th>INLET</th>
<th>X*UP</th>
<th>Y*UP</th>
<th>OUTLET</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>0.200</td>
<td>0.200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>0.400</td>
<td>0.400</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**UNIT 2**

<table>
<thead>
<tr>
<th>INLET</th>
<th>X*UP</th>
<th>Y*UP</th>
<th>OUTLET</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>170</td>
<td>0.500</td>
<td>0.500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>190</td>
<td>0.700</td>
<td>0.700</td>
<td></td>
<td></td>
</tr>
<tr>
<td>210</td>
<td>0.900</td>
<td>0.900</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**UNIT 3**

<table>
<thead>
<tr>
<th>INLET</th>
<th>X*UP</th>
<th>Y*UP</th>
<th>OUTLET</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>240</td>
<td>1.000</td>
<td>1.000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### UNIT AND TRANSLATED TRANSITION ARCS

<table>
<thead>
<tr>
<th>INLET</th>
<th>X*UP</th>
<th>Y*UP</th>
<th>OUTLET</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>0.001</td>
<td>0.001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>0.010</td>
<td>0.010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>0.020</td>
<td>0.020</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INLET</th>
<th>X*UP</th>
<th>Y*UP</th>
<th>OUTLET</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>0.001</td>
<td>0.001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>0.010</td>
<td>0.010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>0.020</td>
<td>0.020</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OUTLET</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td></td>
</tr>
</tbody>
</table>

27
TABLE II - Continued
SAMPLE OUTPUT

<table>
<thead>
<tr>
<th>A(x)</th>
<th>A(y)</th>
<th>A(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
</tr>
<tr>
<td>0.4</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td>0.7</td>
<td>0.8</td>
<td>0.9</td>
</tr>
</tbody>
</table>

28
### TABLE II - Concluded. SAMPLE OUTPUT

#### STRAIGHT LINES

<table>
<thead>
<tr>
<th>$x_i$</th>
<th>$x_{i+1}$</th>
<th>$y_{i+1}$</th>
<th>$y_{i+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.0426</td>
<td>0.0067</td>
<td>0.1236</td>
<td>0.2473</td>
</tr>
<tr>
<td>0.0067</td>
<td>0.1236</td>
<td>0.2473</td>
<td>0.3764</td>
</tr>
<tr>
<td>0.1236</td>
<td>0.2473</td>
<td>0.3764</td>
<td>0.5090</td>
</tr>
<tr>
<td>0.2473</td>
<td>0.3764</td>
<td>0.5090</td>
<td>0.6422</td>
</tr>
</tbody>
</table>

#### MISCELLANEOUS PARAMETERS

- The minimum lower surface prankl-meyer angle predicted by separation conditions is 1.43009°.
- The maximum upper surface prankl-meyer angle predicted by separation conditions is 30.60838°.

<table>
<thead>
<tr>
<th>$x_{MIN}$</th>
<th>$x_{MAX}$</th>
<th>$y_{MIN}$</th>
<th>$y_{MAX}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.43009°</td>
<td>30.60838°</td>
<td>1.43009°</td>
<td>30.60838°</td>
</tr>
</tbody>
</table>

- Table data

#### Miscellaneous

- The minimum lower surface prankl-meyer angle predicted by separation conditions is 1.43009°.
- The maximum upper surface prankl-meyer angle predicted by separation conditions is 30.60838°.

<table>
<thead>
<tr>
<th>$x_{MIN}$</th>
<th>$x_{MAX}$</th>
<th>$y_{MIN}$</th>
<th>$y_{MAX}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.43009°</td>
<td>30.60838°</td>
<td>1.43009°</td>
<td>30.60838°</td>
</tr>
</tbody>
</table>

#### Other

- Table data
PROGRAM DESCRIPTION

Main Program

The main program generates a table of the inlet and outlet transition arcs of the upper, lower, and translated curves. It also computes several parameters which are pertinent to the blade description, such as Mach numbers and radii, and, as an option, plots a blade profile and flow channel. The plotting is done by subroutine PLOTMY, (ref. 10). The program variables are

ALPH fixed angle
ALPHLN rotation angle for lower-curve inlet transition arc, $\alpha_{l,i}$
ALPHLO rotation angle for lower-curve outlet transition arc, $\alpha_{l,o}$
ALPHUI rotation angle for upper-curve inlet transition arc, $\alpha_{u,i}$
ALPHUO rotation angle for upper-curve outlet transition arc, $\alpha_{u,o}$
ALPHUP temporary storage
ALPLOW temporary storage
ANGLE logical switch
BETAN see INPUT
BETAT outlet flow angle, $\beta_o$
CONVER conversion factor for degrees to radians
COSALN cosine of ALPHLN
COSALO cosine of ALPHLO
COSAUI cosine of ALPHUI
COSAUO cosine of ALPHUO
CSTAR blade chord, $C^*$
DALPH angle increment
DELF see subroutine ROOT
DELV see INPUT
DELXI 1/15 length of straight-line portion of upper-curve inlet arc
DELXO 1/15 length of straight-line portion of upper-curve outlet arc
EMJ slope of Mach lines, $m_j$
EMK: slope of Mach lines (eq. (15)), $m_k$
EMWJ: slope of wall segments, $m_j$
EMWK: slope of wall segments (eq. (17)), $m_k$
F(V, FN): internally defined function (eq. (13)), $f(R^*)$
FLO: see subroutine START
FN: floating point index
FOFX: see subroutine ROOT
FUP: see subroutine START
GAM: ratio of specific heats, $\gamma$
GAMEXP: $1/(\text{GAM} - 1)$
GAMM1: $(\text{GAM} - 1)/2$
GAMP1: $(\text{GAM} + 1)/2$
GRTY: dummy name
GSTAR: blade spacing, $G^*$
I: counter
IPRINT: see INPUT
ISTART: see INPUT
J: variable index for upper curve
JJ: variable index
JDEX: variable index
JMAXN: number of points on upper-curve inlet transition arc
JMAXO: number of points on upper-curve outlet transition arc
JMN: maximum of JMAXO and JMAXN
JN: variable index
JNDEX: number of upper-curve transition arc points to be printed
JNN: variable index
JO: variable index
JOO: variable index
K: variable index for lower curve
KK     variable index
KKK    array required by PLOTMY (see ref. 10)
KDEX   variable index
KMAXN  number of points on lower-curve inlet transition arc
KMAXO  number of points on lower-curve outlet transition arc
KMN    maximum of KMAXN and KMAXO
KN     variable index
KNDEX  number of lower-curve transition arc points to be printed
KNN    variable index
KO     variable index
KOO    variable index
KOUNT  counter
L      counter
LLL    NP1 + NP2
LSTORE number of points saved per 5° of turning
LSTR   temporary storage
M      counter
MAXN   integer constant
MAXO   integer constant
N      variable index
NPER   variable governing selective storage
NPLOT  see INPUT
NP1    number of points on lower curve which have been saved for plotter
NP2    number of points on translated lower curve which have been saved for plotter
NP3    number of points on upper curve which have been saved for plotter
NSUM   total number of points which have been stored for plotter
NUM    counter
P      array required by PLOTMY (see ref. 10)
PERM   \[\left(\frac{(GAM + 1)}{(GAM - 1)}\right)^{1/2}\]
PHIJ  flow direction, $\varphi_j$
PHIJP1 previous value of PHIJ, $\varphi_{j+1}$
PHIK  flow direction (eq. (12)), $\varphi_k$
PHIKP1 previous value of PHIK, $\varphi_{k+1}$
R    array for storing radii of major vortex-compression-characteristic points, $R_j^*$
RA   array for storing radii of major vortex-expansion-characteristic points, $R_k^*$
RECONV conversion factor for radians to degrees
RIN  1/SSMIN
RLOW radius of circular arc of lower curve as calculated in JOKOS, $R_l^*$
ROUT 1/SSMOUT
RUP  radius of upper-curve circular arc, $R_u^*$
SAME see subroutine START
SIGMA blade solidity, $\sigma$
SINALN sine of ALPHLN
SINALO sine of ALPHLO
SINAUI sine of ALPHUI
SINAUO sine of ALPHUO
SM  temporary storage for Mach numbers, $M$
SMIN inlet Mach number, $M_i$
SMLOW lower-surface Mach number, $M_l$
SMOUT outlet Mach number, $M_o$
SMS  temporary storage for velocity ratio, $M^*$
SMUP upper-surface Mach number, $M_u$
SSMIN inlet velocity ratio, $M_i^*$
SSMLOW lower-surface velocity ratio, $M_l^*$
SSMOUT outlet velocity ratio, $M_o^*$
SSMUP upper-surface velocity ratio, $M_u^*$
TANBI tangent of BETAN
TANBO tangent of BETAT
TEMP temporary storage
TEMPP temporary storage
TEMPPP temporary storage
THETA total flow turning angle, $\theta$
TR temporary storage for radii
TX temporary storage for unrotated x*-coordinates
TXLO temporary storage for values of XLOW(I)
TXUP temporary storage for values of XUP(I)
TY temporary storage for unrotated y*-coordinates
TYLO temporary storage for values of YLOW(I)
TYUP temporary storage for values of YUP(I)
UMJ Mach angle, $\mu_j$
UMJP1 previous value of UMJ, $\mu_{j+1}$
UMK Mach angle (eq. (15)), $\mu_k$
UMKP1 previous value of UMK, $\mu_{k+1}$
V temporary storage for Prandtl-Meyer angles, $\nu$
VIMAX see subroutine START
VIN see INPUT
VLOW see INPUT
VLSPMN minimum lower-surface Prandtl-Meyer angle from separation criterion, $(\nu_l)_{\text{min}}$
VNL VIN-VLOW
VOL VOUT-VLOW
VOUT see INPUT
VUI VUP-VIN
VUMAX $\frac{\pi}{2} \left( \sqrt{\frac{\gamma + 1}{\gamma - 1}} - 1 \right)$
VUP see INPUT
VUSPMX  maximum upper-surface Prandtl-Meyer angle from separation criterion,
        \((\nu_u)_{\text{max}}\)
VUT      VUP-VOUT
X0       see subroutine ROOT
X1       see subroutine ROOT
X2       see subroutine ROOT
XCG      \(X^\ast\)-coordinate of a translated-curve circular arc point
XCLOW    \(X^\ast\)-coordinate of a lower-curve circular arc point
XCUP     \(X^\ast\)-coordinate of an upper-curve circular arc point
XDOWN    array for storing \(X^\ast\)-coordinate of points to be plotted
XINTL    see subroutine ROOT
XLOW     array for storing \(X^\ast\)-coordinate of unrotated lower transition arc points,
         \((x^*_f)^k\)
XLOWN    array for storing \(X^\ast\)-coordinate of lower-curve rotated inlet transition arc
         points, \((X^*_i)^k,i\)
XLOWO    array for storing \(X^\ast\)-coordinate of lower-curve rotated outlet transition arc
         points, \((X^*_o)^k,o\)
XMLOW    temporary storage for values of \(-XLOW(I)\)
XMUP     temporary storage for values of \(-XUP(I)\)
XSIN     \(X^\ast\)-coordinate of a point on inlet straight-line portion of upper curve
XSOUT    \(X^\ast\)-coordinate of a point on outlet straight-line portion of upper curve
XUP      array for storing \(X^\ast\)-coordinate of unrotated upper transition arc points,
         \((x^*_u)^j\)
XUPN     array for storing \(X^\ast\)-coordinate of upper-curve rotated inlet transition arc
         points, \((X^*_u)^j,i\)
XUPO     array for storing \(X^\ast\)-coordinate of upper-curve rotated outlet transition arc
         points, \((X^*_u)^j,o\)
YACRS1   array for storing \(Y^\ast\)-coordinates of points for plotter
YACRS2   array for temporary storage of \(Y^\ast\)-coordinate of translated lower-curve
         points for plotter
YCG  $y^*$-coordinate of a translated-curve circular arc point
YCLOW  $y^*$-coordinate of a lower-curve circular arc point
YCUP  $y^*$-coordinate of an upper-curve circular arc point
YLASTI  $y^*$-coordinate of first point on inlet side of upper curve
YLASTO  $y^*$-coordinate of first point on outlet side of upper curve
YLOW  array for storing $y^*$-coordinate of unrotated lower transition arc points, $y^*$
YLOWN  array for storing $y^*$-coordinate of lower-curve rotated inlet transition arc points, $y^*_k$, $i$
YLOWO  array for storing $y^*$-coordinate of lower-curve rotated outlet transition arc points, $y^*_k$, $o$
YNG  temporary storage for $y^*$-coordinate of a point on translated-curve inlet transition arc
YSIN  $y^*$-coordinate of a point on inlet straight-line portion of upper curve
YSNG  $y^*$-coordinate of a point on inlet straight-line portion of translated upper curve
YSOUT  $y^*$-coordinate of a point on outlet straight-line portion of upper curve
YSTG  $y^*$-coordinate of a point on outlet straight-line portion of translated upper curve
YTG  temporary storage for $y^*$-coordinate of a point on translated-curve outlet transition arc
YUP  array for storing $y^*$-coordinate of unrotated upper transition arc points, $y^*_u$
YUPN  array for storing $y^*$-coordinate of upper-curve rotated inlet transition arc points, $y^*_u$, $i$
YUPO  array for storing $y^*$-coordinate of upper-curve rotated outlet transition arc points, $y^*_u$, $o$

Subroutine ROOT

Subroutine ROOT is a general routine for finding the roots of equations and is derived from the "half-interval search" method described in reference 11. This method
depends on successively halving an interval which is known to contain the desired root. Subroutine ROOT is used to calculate $R^*$ from equation (10b).

A call to ROOT has the form CALL ROOT (X0, X2, XINTL, FOFX, FUNC, X1), where the elements of the call vector are

X0    lower bound of initial root interval  
X2    upper bound of initial root interval  
XINTL   initial estimate of value of root  
FOFX   given value of dependent variable  
FUNC   externally defined function  
X1    value of root

Other program variables are

A       FOFX - F2
DELF   convergence criterion
F0     function FUNC evaluated at XX0
F2     function FUNC evaluated at XX2
FX     function FUNC evaluated at X
KOUNT  count of number of iterations performed
X       temporary storage for present estimate of root
XX0    present value of lower bound of root interval
XX2    present value of upper bound of root interval

Subroutine START

Subroutine START is used to compute the maximum value of the inlet Prandtl-Meyer angle ($\nu_l$) for supersonic starting for various lower- and upper-surface Prandtl-Meyer angles. Several other parameters of interest are computed by START and are printed as output. Among these parameters are the vortex constant for maximum weight flow $K^*_{\text{max}}$, the reduction in weight flow due to two-dimensional flow $C$, and the maximum value of the inlet velocity ratio for supersonic starting ($M^*_1$).

A call to START has the form CALL START (VLOW, FLO, VUP, FUP, VIMAX), where the elements of the call vector are

VLOW    lower-surface Prandtl-Meyer angle, $\nu_l$
FLO \quad \text{eq. (10b) evaluated at } R_v^*

VUP \quad \text{upper-surface Prandtl-Meyer angle, } \nu_u

FUP \quad \text{eq. (10b) evaluated at } R_u^*

VIMAX \quad \text{maximum inlet Prandtl-Meyer angle for supersonic starting, } (\nu_i^*)_{\text{max}}

Other program variables are

BINTGR \quad \text{value of integral (eq. (27b))}

C \quad \text{reduction in maximum weight flow due to two-dimensional flow (eq. (34b))}

CINTGR \quad \text{value of integral (eq. (32))}

FINTL \quad \text{eq. (27) evaluated at } K_{\text{max}}^* = XINTL

F0 \quad \text{eq. (27) evaluated at } K_{\text{max}}^* = X0

F2 \quad \text{eq. (27) evaluated at } K_{\text{max}}^* = X2

FOFX \quad \text{eq. (27) evaluated at } K_{\text{max}}^* = XAMK

Q \quad \text{eq. (34a)}

RATIO \quad \text{ratio of } Q \text{ to } 1 - C

RLOW \quad \text{radius of circular portion of lower curve, } R_l^*

RUP \quad \text{radius of circular portion of upper curve, } R_u^*

SAME \quad \text{square of ratio of } XAMK \text{ to } SSMLOW

SLOPE \quad \text{slope of line}

SSMIAAX \quad \text{maximum value of entering velocity ratio for starting, } (M_{i\text{max}}^*)

SSMLOW \quad \text{lower-surface velocity ratio, } M_l^*

SSMUP \quad \text{upper-surface velocity ratio, } M_u^*

WSTAR \quad \text{weight-flow parameter (eq. (32))}

XAMK \quad \text{vortex constant for maximum weight flow, } K_{\text{max}}^*

XINTL \quad \text{initial estimate of a parameter}

X0 \quad \text{lower bound of a parameter}

X2 \quad \text{upper bound of a parameter}

YINCPT \quad \text{y-intercept of a line}
Subroutine MSSTAR

Subroutine MSSTAR is used to determine the minimum lower-surface Prandtl-Meyer angle and the maximum upper-surface Prandtl-Meyer angle from separation considerations. This subroutine uses ROOT and ADSTR in the calculation of these angles.

A call to MSSTAR has the form CALL MSSTAR (M, N, VSSTAR), where the elements of the call vector are

- **M**: inlet or outlet dimensionless velocity, $M_i^*$ or $M_o^*$, respectively
- **N**: variable switch
- **VSSTAR**: Prandtl-Meyer angle from separation criterion, $(\nu_\ell)_{\min}^*$ or $(\nu_u)_{\max}^*$

The other variables used by MSSTAR are

- **A**: \[ \frac{\pi}{4} \left( \sqrt{\frac{\gamma + 1}{\gamma - 1}} - 1 \right) \]
- **B**: \[ \frac{1}{2} \sqrt{\frac{\gamma + 1}{\gamma - 1}} \]
- **C**: $\gamma - 1$
- **D**: $\gamma + 1$
- **MS**: velocity ratio from separation criterion, $(M_i^*)_{\min}^*$ or $(M_o^*)_{\max}^*$
- **X0**: see subroutine ROOT
- **X2**: see subroutine ROOT
- **XINTL**: see subroutine ROOT
- **FOFX**: see subroutine ROOT
- **SQRDMS**: $(\text{MS})^2$

Subroutine SIMPS1

This function subprogram is used to perform numerical integration of explicit functions of one variable. The integration is performed by a modification of Simpson's rule, in which a sufficient number of intervals is used to assure six or more significant figures in the result.

A call to SIMPS1 has the form \( \text{ANSWER} = \text{SIMPS1} (\text{XMIN}, \text{XMAX}, \text{FUNC1}, \text{KER}) \), where the elements of the call vector are
XMIN  lower limit of integration
XMAX  upper limit of integration
FUNC1 externally defined function of a single variable
KER    storage for flagging result if necessary

Other program variables are
A      array for storing functional values at certain partition points
ANS    sum of subapproximations
B      array for storing functional values at certain partition points
C      array for storing functional values at certain partition points
E      array for storing difference terms
FRAC   variable tolerance used for subapproximations
H      distance between successive points of partition
K      variable index
N      counter
NE     equivalent to E
NTEST  equivalent to TEST
P      array for storing successive subapproximations
Q      sum of difference terms
SIMPS1 value of desired integral
TEST   tester for subapproximations
T      tolerance for difference terms
V      array for storing partition points of interval

Function Subprograms

The following function subprograms are used intermittently throughout the main
program and subroutines:

FUNCTION ALFUNC (A, B, Y), defined by ALFUNC where

\[
ALFUNC = \frac{1}{Y} \left( A - BY^2 \right)^{1/(\gamma-1)}
\]
FUNCTION CFACT (Y), defined by CFACT where

\[ CFACT = \frac{1}{Y} \left[ 1 - \left( \frac{K^*_{\text{max}}}{M^*_{\ell}} \right)^2 Y^2 \right]^{1/(\gamma - 1)} \]

FUNCTION QFACT (Y), defined by QFACT where

\[ QFACT = \frac{1}{Y} \left( \frac{\gamma + 1}{2} - \frac{\gamma - 1}{2} Y^2 \right)^{1/(\gamma - 1)} \]

FUNCTION FRAT (Y), defined by FRAT where

\[ FRAT = \frac{2\gamma}{Y\gamma - 1} \left( \frac{\gamma + 1 - \gamma - 1 Y^2}{2} \right)^{1/(\gamma - 1)} \]

FUNCTION FOFRS (X), defined by FOFRS where

\[ FOFRS = \sqrt{\frac{\gamma + 1}{\gamma - 1}} \arcsin \left( \frac{\gamma - 1}{X^2} - \gamma \right) + \arcsin \left[ (\gamma + 1)X^2 - \gamma \right] \]

FUNCTION FKMAX (Y, L), defined by FKMAX where

\[ FKMAX = \int_{M^*_\ell}^{M^*_u} \left[ 1 - \left( \frac{X^2}{M^*_\ell} \right) Z^2 \right]^{1/(\gamma - 1)} \frac{dZ}{Z} + \left[ 1 - Y^2 \left( \frac{M^*_u}{M^*_\ell} \right)^2 \right]^{1/(\gamma - 1)} - (1 - Y^2)^{1/(\gamma - 1)} \]

and \( L \) is an optional switch.

FUNCTION ADSTR (X), defined by ADSTR where

\[ ADSTR = \sqrt{\frac{\gamma + 1}{\gamma - 1}} \left\{ 1 - \left[ 1 - \left( \frac{\gamma - 1}{\gamma + 1} \right) X^2 \right] \left[ 1 + \frac{1}{2} \left( \frac{\gamma}{\gamma + 1} \times X^2 \right) \left( \frac{\gamma - 1}{\gamma + 1} \right) \right]^{(\gamma - 1)/\gamma} \right\}^{1/2} \]
COMMON/EXPALF/GAMEXP
COMMON/ROOTS/DELFL
COMMON/FACTOR/PERM,SAME,GAM,GAMM1,GAMPI,SSMLow,SSMUP,RECONV,GRTY

DIMENSION R(800),RA(800),XLOW(800),YLOW(800),XLOWN(800),YLOWN(800),
1,XUP(800),YUP(800),XLOWO(800),YLOWO(800),XUPN(800),YUPN(800),
2XUP0(800),YUP0(800)
DIMENSION XDOWN(400),YACRS1(400),YACRS2(200),KKK(14),P(20)

LOGICAL ANGLE

EXTERNAL FOFRS

F(V,FN) = (2.*V) - ((3.14159265/2.*((PERMIT.)) - (2._(FN-I._)*DELV)

INPUT AND TITLE

ISTART=0 FOR STARTING AND DESIGN ISTART=1 FOR STARTING ONLY
NPLOT=0 IF PLOT IS DESIRED NPLOT=1 IF PLOT IS NOT DESIRED
IPRINT=0 PRINT ROTATED COORDINATES ONLY IPRINT=1 PRINT UNROTATED
AND ROTATED COORDINATES

READ (5,11) VIN,VOUT,BETAN,VLOW,VUP,DELV,GAM,ISTART,NPLOT_,IPRINT
11 FORMAT ( 7(F6.2,2X),3(I1,2X) )
WRITE (6,99)
99 FORMAT (IHI,3FJX,53HO E S I G N O F S U P E R S O N I C B L A

CONVERSION FACTORS AND CONSTANTS
CONVER = .174532925E-01
RECONV = 57.2957796

ONE POINT WILL BE PRINTED FOR EVERY NPER POINTS CALCULATED
IF (DELV .GE. 0.2) GO TO 12
NPER = 10
GO TO 13
12 NPER = 1
13 GAMPI = (GAM + I.)/2.
GAMMI = (GAM-I.)/2.
GAMEXP = 1./(GAM-1.)
PERM = SQRT(GAMPI/GAMMI)
DELFL = 0.000001
X0 = 1./PERM
X2 = 0.999999999
XINTL = (X0 + X2)/2.
LSTORE = (5.*DELV)/FLOAT(NPER)
DAPHL = 1.0*CONVER

ANGLE = .TRUE.
IF (VLOW .LE. AMINI(VUP,VIN,VOUT)) GO TO 120
WRITE (6,119)
114 FORMAT (/31X,7OHV(LOW) MUST BE LESS THAN OR EQUAL TO THE MINIMUM
10F V(UP), V(IN), V(OUT))
ANGLE = .FALSE.

120 IF (VUP GE AMAX1(VIN, VOUT)) GO TO 118
WRITE (6,117)
117 FORMAT (/33X,66HV(UP) MUST BE GREATER THAN OR EQUAL TO THE MAXIMUM
1M OF V(IN), V(OUT))
ANGLE = .FALSE.

118 VUMAX = (3.14159265/2.)*(PERM-1.)*RECONV
IF (VUP LE VUMAX) GO TO 116
WRITE (6,115) VUMAX
115 FORMAT (/41X,37HV(UP) MUST BE LESS THAN V(UP)(MAX) = ,F9.4,4H DEG
1)
ANGLE = .FALSE.

116 IF (.NOT. ANGLE) GO TO 1

CC PARAMETERS FOR STARTING

VLOW = VLOW*CONVER
FLO = F(VLOW,1.0)
VUP = VUP*CONVER
FUP = F(VUP,1.0)
CALL START (VLOW,FLO,VUP,FUP,VIMAX)

IF (ISTART NE 0) GO TO 1

WRITE (6,97)
97 FORMAT (/SRX,17HDESIGN PARAMETERS)

CC MISCELLANEOUS CALCULATIONS
DELV = DELV*CONVER
FN = 1.
V = VIN*CONVER
DO 4 I=1,2
FOFX = F(V,FN)
CALL ROOT (X0,X2,XINTL,FOFX,FOFRS,XI)
IF (I EQ 2) GO TO 4
RIN = XI
V = VOUT*CONVER
4 CONTINUE
ROUT = XI
SSMIN = 1./RIN
CALL MSSTAR (SSMIN,0,VLSPMN)
SSMOUT = 1./ROUT
CALL MSSTAR (SSMOUT,1,VUSPMX)
SMS = SSMIN
I = 1
16 SM = SORT(((1./GAMPI)*SMS*SMS)/(1.-(GAMM1/GAMPI)*SMS*SMS))
GO TO (17,18,19,20), I
17 S MIN = SM
SMS = SSMOUT
I = 2
GO TO 16
18 SMSM = SM
   TEMP = (((GAMMI*SOUT*SMOUT)+1.)/((GAMMI*SMIN*SMIN)+1.))**(GAMPI
1/(2.*GAMMI))
   BETAN = BETAN*CONVER
   BETAT = -ARCOS(COS(BETAN)*(SMIN/SMOUT)*TEMP)
   BETAN = BETAN*RECONV
   BETAT = BETAT*RECONV
   DELV = DELV*RECONV
CC PRINT ALL DESIGN PARAMETERS
   WRITE (6,95) BETAN,VIN,VUP,VOUT,BETAT
95 FORMAT (/2X,11HBETA(IN) = _F7.4_,4X,4HV(IN) = _F7.4,_4X,8HV(UP) : _F7.4,_4X,8HV(OUT) = _F7.4,_12HBETA(OUT)
   BETA(N) = _F7.4)
   WRITE (6,94) DELV, VLOW, GAM
94 FORMAT (/20X,16HDDELTA V = _F7.4,6X,11X,9HV(LOW) = _F7.4)
CC CONVERT FROM DEGREES TO RADIANS
   VIN = VIN*CONVER
   VOUT = VOUT*CONVER
   VUP = VUP*CONVER
   VLOW = VLOW*CONVER
   BETAN = BETAN*CONVER
   BETAT = BETAT*CONVER
   DELV = DELV*CONVER
CC CHOOSE LONGEST TRANSITION ARC OF LOWER SURFACE
   VNL = VIN - VLOW
   KMAXN = (VNL/DELV) + 0.5
   VOL = VOUT - VLOW
   KMAXO = (VOL/DELV) + 0.5
   KMN = MAXO(KMAXN,KMAXO)
   V = AMAXI(VIN,VOUT)
CC CALCULATE R*(LOW)=RLOW, M*(LOW)=SSMLOW, M(LOW)=SSMLOW
   IF (VLOW .LE. 0.0) GO TO 2
   FN = KMN + 1
   FOFX = F(V,FN)
   CALL ROOT (XO,X2,XINTL,FOFX,FOFRS,RLOW)
   GO TO 3
2 RLOW = 1.0
3 SSMLOW = 1./RLOW
   SMS = SSMLOW
   I = 3
   GO TO 16
19 SSMLOW = SM
CC SET INITIAL POINTS FOR LOWER ARC CALCULATIONS
   KNDEX = KMN/NPER
   KDEX = KNDEX
   RA(KDDEX+1) = RLOW
   XLOW(KDDEX+1) = 0.0
   YLOW(KDDEX+1) = RLOW
PHIKP1 = -(V-VLOW) + FLOAT(KMN)*DELV
UMKP1 = ARSIN(SQRT(GAMPI*RLOW*RLOW - GAMM1))
TXLO = XLOW(KDEX+1)
TYLO = YLOW(KDEX+1)
ALPHLN = VNL - BETAN
ALPHLO = -(VOL+BETAT)

IF (ALPHLN .LE. 0.0 .AND. ALPHLO .GE. 0.0) GO TO 110
ANGLE = .FALSE.
WRITE (6,111)
111 FORMAT (/27X,79HV(LOW) MUST BE GREATER THAN OR EQUAL TO V(IN) - BETA(IN) ANO V(OUT) + BETA(OUT))

CC CHOOSE LONGEST TRANSITION ARC OF UPPER SURFACE
110 VUT = VUP - VOUT
JMAXO = (VUT/DELV)+0.5
VUI = VUP - VIN
JMAXN = (VIN/DELV)+0.5
JMN = MAXO(JMAXO,JMAXN)
V = AMINI(VUIT,VIN)

CC CALCULATE R*(UP)=RUP, M*(UP)=SSMUP, M(UP)=MUP
FN = -((JMN+1) + 2
FOFX = F(V,FN)
CALL ROOT (X0,X2,XINTL,FOFX,FOFRS,RUP)
SSMUP = *RUP
SMS = SSMUP
I = 4
GO TO 16
20 SMUP = SM

CC SET INITIAL POINTS FOR UPPER ARC CALCULATIONS
JNDEX = JMN/NPER
JDEX = JNDEX
R(JDEX+1) = RUP
XUP(JDEX+1) = 0.0
YUP(JDEX+1) = RUP
PHIJPI = -(VHP-V) + FLOAT(JMN)*DELV
UMJPI = ARSIN(SQRT(GAMPI*RUP*RUP - GAMM1))
TXUP = XUP(JDEX+1)
TYUP = YUP(JDEX+1)
ALPHU1 = VUI - BETAN
ALPHUO = -(VUT+BETAT)

IF (ALPHU1 .LE. 0.0 .AND. ALPHUO .GE. 0.0) GO TO 112
ANGLE = .FALSE.
WRITE (6,113)
113 FORMAT (/28X,75HV(UP) MUST BE LESS THAN OR EQUAL TO V(IN) + BETA(IN) ANO V(OUT) - BETA(OUT))
112 IF (.NOT. ANGLE) GO TO 1

IF (VIN .EQ. VLOW .AND. VLOW .EQ. VOUT) GO TO 100

C****CALCULATE COORDINATES FOR LOWER TRANSITION ARC - UNROTATED
KDEX = KNDEX + 1
NUM = 0
\[ V = \text{AMAX1}(\text{V}_\text{IN}, \text{V}_\text{OUT}) \]

DO 30 KK = 1, KMN
K = (KMN + 1) - KK
NUM = NUM + 1
PHIK = PHIKP1 - DELV
FN = K
FOFX = F(V, FN)
CALL ROOT (XO, X2, XINTL, FOFX, FOFRS, TR)
TX = TR \times \sin(\text{PHIK})
TY = TR \times \cos(\text{PHIK})
EMWK = \tan(-\text{PHIKP1})
UMK = \arcsin(\sqrt{\text{GAMMA}^2 \cdot \text{TR} \cdot \text{TR} - \text{GAMMA}})
EMK = -\tan((\text{PHIK} + \text{UMK} + \text{PHIKP1} + \text{UMKP1})/2)
TEMP = \text{TYLO} - \text{EMWK} \times \text{TXLO}
TEMPP = TY - \text{EMK} \times \text{TX}
TEMPPP = \text{EMK} - \text{EMWK}
TXLO = ((\text{EMK} \times \text{TEMP} - \text{EMWK} \times \text{TEMPP})/\text{TEMPPP})
TYLO = (((\text{EMK} \times \text{TEMP}) - (\text{EMWK} \times \text{TEMPP})) / \text{TEMPPP})
PHIKP1 = \text{PHIK}
UMKP1 = \text{UMK}

SAVE EVERY \(=\text{NPER-TH}=\) POINT
N = NUM - \((\text{NUM}/\text{NPER}) \times \text{NPER})
IF (N GT 0) GO TO 30
KDEX = KDEX - 1
RA(KDEX) = TR
XLOW(KDEX) = TXLO
YLOW(KDEX) = TYLO

30 CONTINUE

C***CALCULATE COORDINATES FOR LOWER TRANSITION ARC - ROTATED

100 KDEX = KNDEX + 1
KMN = KMN + 1
SINALN = \sin(\text{ALPHLN})
COSALN = \cos(\text{ALPHLN})
SINALO = \sin(\text{ALPHLO})
COSALO = \cos(\text{ALPHLO})
KN = (KMAXN/NPER) + 2
KO = (KMAXO/NPER) + 2

DO 40 KK = 1, KDEX
K = (KDEX + 1) - KK
KN = KN - 1
KO = KO - 1
IF (KN LE 0) GO TO 401
XLOWN(KN) = YLOW(K) \times \text{SINALN} + XLOW(K) \times \text{COSALN}
YLOWN(KN) = YLOW(K) \times \text{COSALN} - XLOW(K) \times \text{SINALN}
401 IF (KO LE 0) GO TO 40
XLOWO(KO) = YLOW(K) \times \text{SINALO} - XLOW(K) \times \text{COSALO}
YLOWO(KO) = YLOW(K) \times \text{COSALO} + XLOW(K) \times \text{SINALO}
40 CONTINUE

IF (\text{VIN} \text{EQ} \text{VUP} \text{AND} \text{VUP} \text{EQ} \text{VOUT}) GO TO 200

C***CALCULATE COORDINATES FOR UPPER TRANSITION ARC - UNROTATED
JDEX = JNDEX + 1
NUM = 0
V = AMIN1(VOUT,VIN)
DO 41 JJ=1,JMN
J = (JMN+1) - JJ
NUM = NUM + 1
PHIJ = PHIJP1 - DELV
FN = -J + 2
FOFX = F(V,FN)
CALL ROOT (XO,X2,XINTL,FOFX,FOFRS,TR)
TX = TR*SIN(PHIJ)
TY = TR*COS(PHIJ)
EMWJ = TAN(-PHIJP1)
UMJ = ARSIN(SQRT(GAMPI*TR*TR - GAMMI))
EMJ = TAN((-PHIJ+UMJ-PHIJP1+UMJP1)/2.)
TEMP = TYUP - EMWJ*TXUP
TEMPP = TY - EMJ*TX
TEMPPP = EMJ - EMWJ
TXUP = (TEMP - TEMPP)/TEMPPP
TYUP = ((EMJ*TEMP) - (EMWJ*TEMPP))/TEMPPP
PHIJPI = PHIJ
UMJPI = UMJ
CC SAVE EVERY =NPER-TH= POINT
N = NUM - (NUM/NPER)*NPER
IF (N .GT. 0) GO TO 41
JDEX = JDEX - 1
R(JDEX) = TR
XUP(JDEX) = TXUP
YUP(JDEX) = TYUP
41 CONTINUE

C****CALCULATE COORDINATES FOR UPPER TRANSITION ARC - ROTATED

200 JDEX = JNDEX + 1
JMN = JMN + 1
SINAUI = SIN(ALPHU1)
COSAUI = COS(ALPHU1)
SINAUO = SIN(ALPHU0)
COSAUO = COS(ALPHU0)
JN = (JMAXN/NPER) + 2
JO = (JMAXO/NPER) + 2

DO 47 JJ=1,JDEX
J = (JDEX+1) - JJ
JO = JO - 1
JN = JN - 1
IF (JO .LE. 0) GO TO 471
XUPO(JO) = YUP(J)*SINAUO - XUP(J)*COSAUO
YUPO(JO) = YUP(J)*COSAUO + XUP(J)*SINAUO
471 IF (JN .LE. 0) GO TO 47
XUPN(JN) = YUP(J)*SINAUI + XUP(J)*COSAUI
YUPN(JN) = YUP(J)*COSAUI - XUP(J)*SINAUI
47 CONTINUE

CC CALCULATE G* - THE DIMENSIONLESS BLADE SPACING
TANBI = TAN(BETAN)
YLASTI = YUPN(1) + TANBI*(XLOWN(1) - XUPN(1))
GSTAR = YLOWN(1) - YLASTI

CC TITLES
WRITE (6,93)
93 FORMAT (///54X,25HL O W E R S U R F A C E)
     IF (IPRINT .EQ. 0) GO TO 844
     WRITE (6,88)
88 FORMAT (///54X,25HUNROTATED TRANSITION ARCS)
     WRITE (6,87)
87 FORMAT (///2X,8H INLET K,4X,8HX*(LOW),3X,8HY*(LOW),68X,8HY*(LOW)
1,3X,8HX*(LOW),2X,8HOUTLET K)

C*****PRINT COORDINATES FOR LOWER TRANSITION ARC - UNROTATED
KDEX = KINDEX + 2
DO 51 KK=1,KMIN,NPER
   K = (KMIN+1) - KK
   KN = (KMAXN+2) - KK
   KO = (KMAXO+2) - KK
   KDEX = KDEX - 1
   XMLOW = -XLOW(KDEX)
   IF (KN .GT. 0 .AND. KO .GT. 0) GO TO 510
   IF (KN .LE. 0) GO TO 511
   WRITE (6,861) KN,XLOW(KOEX),YLOW(KI)EX)
51 CONTINUE

CC TITLES
844 WRITE (6,92)
92 FORMAT (///HL,46X,38HRotated and Translated Transition Arcs)
     WRITE (6,84)
84 FORMAT (///2X,8H INLET K,6X,7HX*(LOW),7X,7HY*(LOW),6X,10HY*(LOW)-G*
1,28X,10HY*(LOW)-G*,5X,7HY*(LOW),7X,7HX*(LOW),5X,8HOUTLET K)

C*****PRINT COORDINATES FOR LOWER TRANSITION ARC - ROTATED
M = 1
   XDOW(M) = XLOWN(1)
   YACRS1(M) = YLOWN(1)
   YACRS2(M) = YLOWN(1) - GSTAR
   M = M + 1
   XDOW(M) = XLOWO(1)
   YACRS1(M) = YLOWO(1)
   YACRS2(M) = YLOWO(1) - GSTAR

CC STORE POINTS FOR PLOTTER - ONE POINT FOR EVERY FIVE DEGREES OF TURNING
MAXN = (KMAXN/NPER) + 1
MAXO = (KMAXO/NPER) + 1
KNN = MAXN + 1
KOO = MAXO + 1
I = 0
DO 55 KK=1,KMN,NPER
   KN = (KMAXN+2) - KK
   KO = (KMAXO+2) - KK
   KNN = KNN - 1
   KO0 = KO0 - I
   I = I + 1
   LSTR = LSTORE(I)
   IF (KN .GT. 0 .AND. KO .GT. 0) GO TO 550
   IF (KN .LE. 0) GO TO 551
   IF (LSTR .GT. MAXN) GO TO 559
   M = M + 1
   XDOWN(M) = XLOWN(LSTR)
   YACRS1(M) = YLOWN(LSTR)
   YACRS2(M) = YLOWN(LSTR) - GSTAR
   559 YNG = YLOWN(KNN) - GSTAR
   WRITE (6,831) KN,XLOWN(KNN),YLLOWN(KNN),YNG
   831 FORMAT (4X,I4,7X,F8.4,6X,F8.4,6X,F8.4)
   GO TO 55
   550 YNG = YLOWN(KNN) - GSTAR
   YTG = YLOWO(KO0) - GSTAR
   IF (LSTR .GT. MAXO) GO TO 557
   M = M + 1
   XDOWN(M) = XLOWO(LSTR)
   YACRS1(M) = YLOWO(LSTR)
   YACRS2(M) = YLOWO(LSTR) - GSTAR
   557 YTG = YLOWO(KO0) - GSTAR
   WRITE (6,833) YTG,YLOWO(KO0),XLOWO(KO0),KO
   833 FORMAT (4X,F8.4,6X,F8.4,6X,F8.4,6X,F8.4,7X,I4)
   GO TO 55
   551 IF (LSTR .GT. MAXO) GO TO 557
   M = M + 1
   XDOWN(M) = XLOWO(LSTR)
   YACRS1(M) = YLOWO(LSTR)
   YACRS2(M) = YLOWO(LSTR) - GSTAR
   557 YTG = YLOWO(KO0) - GSTAR
   WRITE (6,830) KN,XLOWN(KNN),YLOWN(KNN),YNG,YTG,YLOWO(KO0)-XLOWO(KO0),KO
   830 FORMAT (4X,I4,7X,F8.4,6X,F8.4,6X,F8.4,30X,F8.4,6X,F8.4,6X,F8.4,7X,114)
   55 CONTINUE
   M = M + 1
   XDOWN(M) = XLOWO(MAXO)
   YACRS1(M) = YLOWO(MAXO)
   YACRS2(M) = YLOWO(MAXO) - GSTAR

C###CIRCULAR ARC (LOWER)
   IF (IPRINT .EQ. 0) GO TO 810
WRITE (6, 81)
81 FORMAT (/60X,13HCIRCULAR ARCS//40X,8HX*C(LOW),3X,8HY*C(LOW),16X,  
1 8HX*C(LOW),3X,11HY*C(LOW)-G*)

810 M = M + 1
  XDOWN(M) = 0.0
  YACRS1(M) = RLOW
  YACRS2(M) = RLOW - GSTAR
  THETA = (BETAN - BETAT)*RECONV
  ALPH = ALPHLO + DALPH
  ALPLOW = ALPHLO
  KOUNT = 0
60 XCLOW = RLOW*SIN(ALPLOW)
  YCLOW = RLOW*COS(ALPLOW)
  XCLOW = XCLOW
  YCG = YCLOW - GSTAR
  KOUNT = KOUNT + 1
  IF (KOUNT .NE. LSTORE) GO TO 601
  KOUNT = 0
  M = M + 1
  XDOWN(M) = XCLOW
  YACRS1(M) = YCLOW
  YACRS2(M) = YCG
601 IF (IPRINT .EQ. 0) GO TO 800
  WRITE (6, 80) XCLOW, YCLOW, XCG, YCG
80 FORMAT (39X,FB.4,3X,FB.4,16X,FS*4,3X,F8.4)
800 ALPLOW = ALPLOW + DALPH
  IF (ABS(ALPH-ALPLOW) .LT. 0.001) GO TO 56
  IF (ALPHLO .LT. ALPLOW .AND. ALPLOW .LT. ALPH) ALPLOW = ALPHLO
  GO TO 60

CC STORE THE TRANSLATED LOWER ARC FOR PLOTTER
56 NP1 = M
  DO 2000 I=1,NP1
  M = M + 1
  XDOWN(M) = XDOWN(I)
2000 YACRS1(M) = YACRS2(I)
  NP2 = NP1

CC TITLES
WRITE (6, 79)
79 FORMAT (1H1,53X,25HU P P E R S U R F A C E)
  IF (IPRINT .EQ. 0) GO TO 700
  WRITE (6, 74)
74 FORMAT (/54X,25HU R T R A N S I T I O N A R C S)
73 FORMAT (/2X,8H INLET J,3X,8H X*(UP),3X,8H Y*(UP),6HX,8H Y*(UP)
1,3X,8H X*(UP),3X,8HOUTLET J)

C****PRINT CO [DINATES FOR UPPER TRANSITION ARC - UNROTATED
JDEX = JNDEX + 2
  DO 65 JJ=1,JMN,NPER
  J = (JMN+I) - JJ
  JD = (JMAXO+2) - JJ
JN = (JMAXN+2) - JJ
JDEX = JDEX - 1
XMUP = -XUP(JDEX)
IF (JN .GT. 0 .AND. JO .GT. 0) GO TO 650
IF (JN .LE. 0) GO TO 651
WRITE (6,720) JN,XUP(JDEX),YUP(JDEX)
720 FORMAT (4X,14,5X,F8.4,3X,F8.4)
GO TO 65
650 WRITE (6,723) YUP(JDEX),XMUP,JO
723 FORMAT (100X,F8.4,3X,F8.4,5X,14)
GO TO 65
650 WRITE (6,721) JN,XUP(JDEX),YUP(JDEX),YUP(JDEX),XMUP,JO
721 FORMAT (4X,14,5X,F8.4,3X,F8.4,68X,F8.4,3X,F8.4,5X,14)
65 CONTINUE

CC TITLES
700 WRITE (6,78)
78 FORMAT (1HL,46X,38HROTATED AND TRANSLATED TRANSITION ARCS)
WRITE (6,70)
70 FORMAT (/2X,8H INLET J,6X,7H X*(UP),7X,7H Y*(UP),6X,9HY*(UP)+G*,
129X,9HY*(UP)+G*,6X,7H Y*(UP),7X,7H X*(UP),5X,8HOUTLET J)
L = L + NPER
C***PRINT COORDINATES FOR UPPER TRANSITION ARC - ROTATED
CC STORE POINTS FOR PLOTTER - ONE POINT FOR EVERY FIVE DEGREES OF TURNING
L = L + 1
XDOWN(L) = XUPN(1)
YACRS1(L) = YUPN(1)
L = L + 1
XDOWN(L) = XUPO(1)
YACRS1(L) = YUPO(1)
MAX0 = (JMAXO/NPER) + 1
MAXN = (JMAXN/NPER) + 1
J00 = MAXO + 1
JNN = MAXN + 1
I = 0
DO 303 JJ=1,JMN,NPER
303 J0 = (JMAXO+2) - JJ
JN = (JMAXN+2) - JJ
J00 = J0 + 1
JNN = JNN + 1
I = I + 1
LSTR = LSTORE*1
IF (JN .GT. 0 .AND. J0 .GT. 0) GO TO 3030
IF (JN .LE. 0) GO TO 3031
IF (LSTR .GT. MAXN) GO TO 688
L = L + 1
XDOWN(L) = XUPN(LSTR)
YACRS1(L) = YUPN(LSTR)
688 YNG = YUPN(JNN) + GSTAR
WRITE (6,68) JN,XUPN(JNN),YUPN(JNN),YNG
68 FORMAT (4X,14,7X,F8.4,6X,F8.4,6X,F8.4)
GO TO 303
3031 IF (LSTR .GT. MAXO) GO TO 689
L = L + 1
XDOWN(L) = XUPO(LSTR)
YACRSI(L) = YUPO(LSTR)

689 YTG = YUPO(J00) + GSTAR
WRITE (6,683) YTG, YUPO(J00), XUPO(J00), J0
683 FORMAT (81X,F8.4,6X,F8.4,6X,F8.4,7X,14)
GO TO 303

3030 YNG = YUPN(JNN) + GSTAR
YTG = YUPO(J00) + GSTAR
IF (LSTR .GT. MAXN) GO TO 670
L = L + 1
XDOWN(L) = XUPN(LSTR)
YACRSI(L) = YUPN(LSTR)

670 IF (LSTR .GT. MAXO) GO TO 671
L = L + 1
XDOWN(L) = XUPO(LSTR)
YACRSI(L) = YUPO(LSTR)

671 WRITE (6,680) JN, XUPN(JNN), YUPN(JNN), YNG, YTG, YUPO(J00), XUPO(J00), J0
680 FORMAT (4X,14,7X,FS.4,6X,F8.4,6X,FS.4,30X,F8.4,6X,FS.4,6X,FS.4,7X,14)
303 CONTINUE

L = L + 1
XDOWN(L) = XUPO(MAXN)
YACRSI(L) = YUPN(MAXN)
L = L + 1
XDOWN(L) = XUPO(MAXO)
YACRSI(L) = YUPO(MAXO)

C****CIRCULAR ARC (UPPER)
IF (IPRINT .EQ. 0) GO TO 6700
WRITE (6,67)
67 FORMAT (//60X,13HCIRCULAR ARCS//4OX,8HX*C(UP) ,3X,SHY*C(UP) ,16X,
1 8HX*C(UP) ,3X,10HY*C(UP)+G*)

6700 L = L + 1
XDOWN(L) = 0.0
YACRSI(L) = RUP

ALPH = ALPHUO + DALPH
ALPHUP = ALPHUI
KOUNT = 0

305 XCUP = RUP*SIN(ALPHUP)
YCUP = RUP*COS(ALPHUP)
XCG = XCUP
YCG = YCUP + GSTAR
KOUNT = KOUNT + 1

IF (KOUNT .NE. LSTORE) GO TO 672
KOUNT = 0
L = L + 1
XDOWN(L) = XCUP
YACRSI(L) = YCUP

672 IF (IPRINT .EQ. 0) GO TO 660
WRITE (6,66) XCUP,YCUP,XCUP,YCG

66 FORMAT (39X,F8.4,3X,F8.4,16X,F8.4,3X,F8.4)

660 ALPHUP = ALPHUP + DALPH
IF (ALPHUP .LT. ALPH ) GO TO 306
IF (ALPHLUO .LT. ALPHUP .AND. ALPHUO .LT. ALPH) ALPHUP = ALPHUO
GO TO 305

C****CALCULATE COORDINATES FOR STRAIGHT LINE PORTION OF UPPER ARC
CC FIFTEEN POINTS ARE CALCULATED FOR PLOTTING PURPOSES

306 IF (IPRINT .EQ. 0) GO TO 3070
WRITE (6,307)

307 FORMAT (/59X,14HSTRAIGHT LINES//5X,8H X*S(IN),5X,8H Y*S(IN),3X,
110H Y*S(IN)+G*,54X,11HY*S(OUT)+G*,2X,8HY*S(OUT),5X,8HX*S(OUT))

3070 KOUNT = -1
DELXI = (XUPN(1) - XLOWN(1))/15.
DELXO = (XLOWO(1) - XUPO(1))/15.
XSIN = XUPN(1)
YSIN = YUPN(1)
XSOUT = XUPO(1)
YSOUT = YUPO(1)
TANBO = TAN(BETAT)
GO TO 309

310 XSIN = XSIN - DELXI
XSOUT = XSOUT + DELXO
YSIN = YUPN(1) + TANBI*(XSIN - XUPN(1))
YSOUT = YUPO(1) + TANBO*(XSOUT - XUPO(1))

309 YSNG = YSIN + GSTAR
YSTG = YSOUT + GSTAR
IF (XSIN .LE. XLOWN(1)) GO TO 312
KOUNT = KOUNT + 1
N = KOUNT - (KOUNT/3)*3
IF (N .GT. 0) GO TO 673
L = L + 1
XDOWN(L) = XSIN
YACRS(L) = YSIN

673 IF (IPRINT .EQ. 0) GO TO 3133
WRITE (6,313) XSIN,YSIN,YSNG

313 FORMAT (5X,F8.4,4X,F8.4,4X,F8.4)

3133 IF (XSOUT .GE. XLOWO(1)) GO TO 310
IF (N .GT. 0) GO TO 674
L = L + 1
XDOWN(L) = XSOUT
YACRS(L) = YSOUT

674 IF (IPRINT .EQ. 0) GO TO 310
WRITE (6,315) YSTG,YSOUT,XSOUT

315 FORMAT (1H*,93X,F8.4,4X,F8.4,4X,F8.4)

312 IF (XSOUT .GE. XLOWO(1)) GO TO 311
IF (IPRINT .EQ. 0) GO TO 310
WRITE (6,321) YSTG,YSOUT,XSOUT

321 FORMAT (94X,F8.4,4X,F8.4,4X,F8.4)

311 NP3 = L - (NP1 + NP2)
NSUM = NP1 + NP2 + NP3 + 1

53
C****MISCELLANEOUS CALCULATIONS
WRITE (6,622)
622 FORMAT (//54X,24HMISCELLANEOUS PARAMETERS//)
  YLASTO = YUP0(1) + TANBO*(XLOWO(1) - XUPO(1))
  CSTAR = SQRT( ((XLOWO(1) - XLOW(1))**2) + ((YLOWO(1) - YLOW(1))
                I**2) )
  SIGMA = CSTAR /GSTAR
WRITE (6,999) VLSPMN,VUSPMX
999 FORMAT (17X,84HTHE MINIMUM LOWER SURFACE PRANDTL-MEYER ANGLE PREDICTED
         BY SEPARATION CONDITIONS IS ,F9.4,4H IEG//IIX,B4HTHE MAXIMUM
         UPPER SURFACE PRANDTL-MEYER ANGLE PREDICTED BY SEPARATION CONDITIO
         NS IS ,F9.4,4H DEG)
WRITE (6,1000) SSMIN,SMIN,SMOUT,SSMOUT
1000 FORMAT (/25X, 9HM*(IN) = ,F8.4,2X,9HM(OUT) =
           1,F8.4,5X,10HM*(OUT) = ,F8.4)
WRITE (6,1001) RLOW,SSMLOW,SMLOW,SSMUP,SMUP,RUP
1001 FORMAT (/2X,gHR*(LOW) =,F8.4,5X,10H M(LOW) = ,F8.4,2X,9H M(LOW) = ,
           1,F8.4,10X,9H M(UP) = ,F8.4,5X,10H M(UP) = ,F8.4,2X,9H M(UP) = ,
           2,F8.4)
WRITE (6,1002) THETA,GSTAR,CSTAR,SIGMA
1002 FORMAT (/IIX,BHTHETA = ,F8.4,4H DEG,12X,5HG_* = ,F8.4,11X,5H_*
           = ,F8.4,10X,5H_*(OUT) = ,F8.4)
IF (NPLOI .NE. 0) GO TO I
C IF PLOTMY IS NOT AVAILABLE, REMOVE THE FOLLOWING CARDS
C****MULTIPLE PLOT - START
LLL = NP1 + NP2
CALL SORTXY (XDOWN(1),YACRSI(1),NP1)
CALL SORTXY (XDOWN(NP1+1),YACRSI(NP1+1),NP2)
CALL SORTXY (XDOWN(LLL+1),YACRSI(LLL+1),NP3)
  P(1) = 5.0
  P(3) = 12.0
  P(4) = 20.0
  P(11) = (1. - AMINI(YACRSI(1),YACRSI(NP1+1),YACRSI(LLL+1)))/100.*
         1(10.,**4)
  P(6) = 2.0
  P(7) = AMINI( XDOWN(1),XDOWN(NP1+1),XDOWN(LLL+1) ) *(10.,**4)
  P(8) = P(11)*5./3.
  P(9) = 2.0
  P(10) = AMINI(YACRSI(1),YACRSI(NP1+1),YACRSI(LLL+1))*10.,**4
KKK(1) = 55
KKK(2) = 4
KKK(3) = NP1
KKK(5) = NP2
KKK(7) = NP3
KKK(9) = 1
DATA KKK(4),KKK(6),KKK(8),1H#,1H#,#,KJKK(10)/1HO/
CALL PLOTMY (XDOWN,YACRSI,KKK,P)
C****MULTIPLE PLOT - STOP
GO TO 1
END
SUBROUTINE ROOT (XO, X2, XINTL, FOFX, FUNC, XI)
COMMON/ROOTS/DELF
double precision x, xx0, xx2

WE ARE SEEKING AN X SUCH THAT FUNC(X) = FOFX WHERE FOFX IS A KNOWN FUNCTIONAL VALUE

1 LOCATE FOFX IN (FO, FX) OR (FX, F2) WHERE FX IS THE PREVIOUS APPROXIMATION TO FOFX
2 LET X = 0.5*(XX0 + X) OR X = 0.5*(X + XX2)
3 IS FUNC(X) = FOFX = IF NOT, REPEAT PROCEDURE

XX0 = XO
XX2 = X2
FO = FUNC(XX0)
F2 = FUNC(XX2)
IF (FOFX .LT. FO .AND. FOFX .LT. F2)
   GO TO 1005
IF (FOFX .GT. F2)
   GO TO 1007
IF (ABS(FOFX - F2) .LE. DELF) GO TO 1008
X = XINTL
KOUNT = 0
1000 X1 = X
KOUNT = KOUNT + 1
A = FOFX - F2
FX = FUNC(X)
IF (KOUNT .GE. 60) WRITE (6, 1004) KOUNT, X, FX, FOFX
1004 FORMAT (1HL, 9H KOUNT, G16.9, 9H X, G16.9, 9H FX, G16.9, 19H FOFX, G16.9)
IF (ABS(FX - FOFX) .LE. DELF) RETURN
IF (KOUNT .EQ. 75) GO TO 1002
IF (A*(FX - FOFX) .LT. 0.) GO TO 1001
XX0 = X
X = (X + XX2)/2.
GO TO 1000
1001 XX2 = X
X = (XX0 + X)/2.
F2 = FX
GO TO 1000

1002 WRITE (6, 1003)
1003 FORMAT (3/30X, 62H7h iterations have been performed without converging to a root)
RETURN

1005 WRITE (6, 1006) FOFX
1006 FORMAT (3/10X, 7HF(X) = G16.9, 31H is outside of specified limits)
RETURN

1007 X1 = XO
RETURN

1008 X1 = X2
RETURN

END
SUBROUTINE START (VLOW, FLO, VUP, FUP, VIMAX)

COMMON/FACTOR/PERM, SAME, GAM, GAMM1, GAMPI, SSMLOW, SSMUP, RECONV, BINTGR
EXTERNAL CFAC_, OFACT, FRAT, FOFRS, FKMAX

X0 = 1./PERM
X2 = 0.999999999
XINTL = (X0 + X2)/2.

IF (VLOW .LE. 0.0) GO TO 70
CALL ROOT (X0, X2, XINTL, FLU, FOFRS, RLOW)
GO TO 71

70 RLOW = 1.0
71 SSMLOW = 1./RLow
CALL ROOT (X0, X2, XINTL, FUP, FOFRS, RUP)
SSMUP = 1./RUP

IF (SSMLOW .LE. SSMUP) GO TO 40

C C FKMAX(X) IS LINEAR IN A NEIGHBORHOOD OF X WHEN X IS SUCH THAT FKMAX(X) = 0
C USE GOOD INITIAL ESTIMATE PLUS LINEARITY TO FIND X SUCH THAT FKMAX(X) = 0

XINTL = (1./PERM)*SORT( SSMLOW/SSMUP )
X0 = XINTL - 0.005
F0 = FKMAX(X0, 0)
X2 = XINTL + 0.001
F2 = FKMAX(X2, 0)
SLOPE = (F2 - F0)/(X2 - X0)
FINTL = FKMAX(XINTL, 0)
YINCPT = FINTL - SLOPE*XINTL
XAMK = -YINCPT/SLOPE
FOFX = FKMAX(XAMK,

IF (ABS(FOFX) .GT. 0.00009) WRITE (6,60) FOFX, XAMK

60 FORMAT (//29X,35HSEARCH FOR ROOT FAILED F(X) : ,G16,9,7H X
1= ,G16,9)

SAME = (XAMK/SSMLOW)*(XAMK/SSMLOW)
C = 1. - PERM*GAMPI**((1./GAM-1.))*SSMUP/(SSMUP-SSMLOW)*XAMK*
BINTGR
CINTGR = SIMPS1(SSMLOW, SSMUP, OFACT, K)
Q = (SSMLOW*SSMUP/(SSMUP-SSMLOW)*CINTGR
RATIO = Q/(1. - C)
GO TO 50

40 XAMK = 1./PERM
RATIO = SSMUP*SSMUP*OFACI(SSMUP)
C = 0.0
Q = 0.0
GO TO 50

50 XO = 1.0
X2 = PERM
XINTL = (X0 + X2)/2.
CALL ROOT (X0, X2, XINTL, RATIO, FRAT, SSMIAx)
VIMAX = (3.14159265/4.)*(PERM-1.) + (PERM/2.)*ARSin(2.*GAMM1*
1 SSMIAx*SSMIAx - GAM) + 0.5*ARSin(2.*GAMPI/(SSMIAx*SSMIAx) - GAM)
VIMAX = VIMAX*RECONV
VLOW = VLOW*RECONV
VUP = VUP*RECONV
WSTAR = ((I./GAMPI)**(GAMPI/(2.*GAMMI)))*CINTGR

WRITE (6,10)
10 FORMAT (//48X,36HCALCULATIONS FOR SUPERSONIC STARTING)
WRITE (6,90) WSTAR
90 FORMAT (/50X,24HWEIGHT-FLOW PARAMETER = ,F9.4)
WRITE (6,20) XAMK,C,O,SSMIAX
20 FORMAT (/2OX,10HK*(MAX) = ,F9.4,SX,5H C = ,F9.4,SX,5H O = ,F9.4)
WRITE (6,30) VIMAX,VLOW,VUP,GAM
30 FORMAT (/4X,38HTHE MAXIMUM DESIGN VALUE FOR V(IN) IS ,F9.4,21H DEG
1 WHEN V(LOW) IS ,F9.4,16H DEG, V(UP) IS ,F9.4,16H DEG, GAMMA IS
2 ,F7.4)
RETURN
END

$IBFTC MSS LIST
SUBROUTINE MSSTAR (M,N,VSSIAR)
COMMON/FACTOR/PERM,SAME,GAM,GAMMI,GAMPI,SSMLOW,SSMUP,RECONV,D
REAL M,MS
EXTERNAL ADSTR
A = 0.785398162*(PERM -I.)
B = 0.5*PERM
C = GAM -1.
D = GAM + 1.
IF(N .NE. 0) GO TO 1
MS = ADSTR(M)
IF (MS ,LT. 1.) GO TO 3
GO TO 2
1 X0 = I.
X2 = PERM
XINTL = (X0 + X2)/2.
FOFX = M
CALL ROOT (X0,X2,XINTL,FOFX,ADSTR,MS)
2 SORDMS = MS*MS
VSTAR = ( A + B*ARSin(C*SORDMS-GAM) + 0.5*ARSin(D/SORDMS-GAM) )*RECONV
RETURN
3 VSTAR = 0.
RETURN
END
FUNCTION SIMPS1(XMIN, XMAX, FUNCI, KER)
DIMENSION V(200), H(200), A(200), B(200), C(200), P(200), E(200), NE(200)
EQUIVALENCE (E, NE), (TEST, NTEST)

T = 3.0E-5
V(1) = XMIN
H(1) = 0.5*(XMAX-XMIN)
A(1) = FUNCI(XMIN)
B(1) = FUNCI(XMIN+H(1))
C(1) = FUNCI(XMAX)
P(1) = H(1)*(A(1) + 4.0*B(1) + C(1))
E(1) = P(1)
ANS = P(1)

N = 1
FRAC = 2.0*T

1 FRAC = 0.5*FRAC
2 TEST = ABS(FRAC*ANS)
3 K = N
4 DO 7 I = 1, K
5 N = N + 1
6 IF (N = 200) 7, 13, 13
7 CONTINUE
5
6 IF (N - K) 9, 9, 2
8 IF (N - K) 9, 9, 2
9 Q = 0.0

10 DO 11 I = 1, N
11 Q = Q + E(I)
12 IF (ABS(Q) - T*ABS(ANS)) 14, 14, 1
13 KER = KER + 1
14 ANS = 0.0
15 DO 16 I = 1, N
16 ANS = ANS + P(I)
SIMPS1 = (ANS + Q/30.0)/3.0
17 RETURN

END
$1BFTC  ALLFUN
FUNCTION  ALFUNC (A,B,Y)

COMMON/EXPALF/GAMEXP
COMMON/FACTOR/PERM,SAME,GAM,GAMM1,GAMP1,SSMLOW,SSMUP,RECONV,GRTY

ALFUNC = (1./Y)*((A - B*Y*Y)**GAMEXP)

RETURN
END

$1BFTC  BAKE
FUNCTION  CFACT (Y)

COMMON/FACTOR/PERM,SAME,GAM,GAMM1,GAMP1,SSMLOW,SSMUP,RECONV,GRTY
EXTERNAL  ALFUNC

CFACT = ALFUNC(1.,SAME,Y)

RETURN
END

$1BFTC  CHARL
FUNCTION  OFACT(Y)

COMMON/FACTOR/PERM,SAME,GAM,GAMM1,GAMP1,SSMLOW,SSMUP,RECONV,GRTY
EXTERNAL  ALFUNC

OFACT = ALFUNC(GAMPI,GAMMI,Y)

RETURN
END

$1BFTC  DOGG
FUNCTION  FRAT(Y)

COMMON/FACTOR/PERM,SAME,GAM,GAMM1,GAMP1,SSMLOW,SSMUP,RECONV,GRTY
EXTERNAL  OFACT,ALFUNC

FRAT = (Y**(GAM/GAMM1))*OFACT(Y)/ALFUNC(-GAMM1,-GAMP1,Y)

RETURN
END
$IBFTC FELI

FUNCTION FOFRS (X)

COMMON/FACTOR/PERM,SAME,GAM,GAMMI,GAMPI,SSMLow,SSMup,RECONV,GRTY
DOUBLE PRECISION X

ARG1 = 2.*GAMMI/(X*X) - GAM
ARG2 = 2.*GAMPI*X*X - GAM
IF (ABS(ARG1) .GT. 1.0 .OR. ABS(ARG2) .GT. 1.0) WRITE (6,1) ARG1
1 ARG2
1 FORMAT (/14X,61HARGUMENT OF ARCSIN IS OUTSIDE DOMAIN OF DEFINITION
IN ARG1 = ,G16.9,1H ARG2 = ,G16.9)

FOFRS = PERMSIN(ARG1) + SIN(ARG2)

RETURN
END

$IBFTC GERT

FUNCTION FKMAX(Y,L)

COMMON/FACTOR/PERM,SAME,GAM,GAMMI,GAMPI,SSMLow,SSMup,RECONV,BINTGR
EXTERNAL ALFUNC,CFACT

SAME = (Y/SSMLow)*(Y/SSMLow)
K = 0
FKMAX = SIMPS1(SSMLow,SSMup,CFACT,K)
IF (K .EQ. 1) WRITE (6,1)
1 FORMAT (/10X,26HFAILURE TO INTEGRATE CFACT)
IF (L .EQ. 1) BINTGR=FKMAX

FKMAX = FKMAX + SSMup*CFACT(SSMup) - ALFUNC(1.,Y+Y,1.)

RETURN
END
FUNCTION AUSTR(MSTAR)

COMMON/FAC'TUR/PERM, SAME, GAM, GAM1, GAM2, SSMLUM, SSMUP, RECONV, D
REAL MSTAR, M

C = GAM - 1.
F = C/I)
G = C/GAM
H = GAM/I)
M = MSTAR*MSTAR

AUSTR = PERM*SORT( (1 - (1 - F*M)*(1 + 0.5*(H*M)/((1 - F*M)))*G)) )

RETURN
END

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, October 6, 1967,
128-31-02-25-22.

REFERENCES


