NASA TECHNICAL NOTE

NASA TN D-2546

NASA TN D-2546

GPO PRICE $ ______

OTS PRICE(S) $ 2.00

Hard copy (HC) ______

Microfiche (MF) 0.50

N65-12306 (ACCESSION NUMBER)

(THRU)

48 (PAGES)

1 (CODE)

(NASA CR OR TMX OR AD NUMBER)

01 (CATEGORY)

FACILITY FORM 602

USE OF ARBITRARY QUASI-ORTHOGONALS FOR CALCULATING FLOW DISTRIBUTION IN THE MERIDIONAL PLANE OF A TURBOMACHINE

by Theodore Katsanis

Lewis Research Center

Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • DECEMBER 1964

USE OF ARBITRARY QUASI-ORTHOGONALS FOR CALCULATING FLOW DISTRIBUTION IN THE MERIDIONAL PLANE OF A TURBOMACHINE

By Theodore Katsanis

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Office of Technical Services, Department of Commerce,
Washington, D.C. 20230 -- Price $2.00

USE OF ARBITRARY QUASI-ORTHOGONALS FOR CALCULATING FLOW DISTRIBUTION IN THE MERIDIONAL PLANE OF A TURBOMACHINE

by Theodore Katsanis

Lewis Research Center

SUMMARY

12306

A method of analyzing flow through a turbomachine is presented that is suitable for computer programing. It is assumed that a mean stream surface from hub to shroud between blades is known. On this stream surface a two-dimensional solution for the velocity and pressure distributions is obtained, and then an approximate calculation of the blade surface velocities is made. This method is based on an equation for the velocity gradient along an arbitrary quasi-orthogonal rather than the normal to the streamline as used in previous methods. With this new method a solution can be obtained in a single computer run, even for cases where the distance between hub and shroud is great and there is a change in direction from radial to axial within the rotor. The method was successfully applied to a turbine with this type of geometry. These results are given as a numerical example, and the Fortran computer program is included.

Author

INTRODUCTION

Quasi-three-dimensional methods have been developed for analyzing flow through mixed-flow turbomachines. One such method (ref. 1) is based on the assumption of axial symmetry and on an equation for the velocity gradient along the normal to the projection of the streamlines on a plane containing the axis of rotation. This basic method was used (ref. 2) to redesign the hub-shroud profile of a compressor rotor. The results of reference 3 showed improved performance for impellers redesigned by this method. A computer program using this method for the design of pump impellers is described in reference 4. In reference 4, the same velocity gradient equation given in reference 1 is developed without the assumption of axial symmetry but with the assumption of a known stream surface that extends from hub to shroud. The examples used in the aforementioned references were all compressors or pumps, but the method is equally applicable to turbines.

These methods use streamlines and their normals to establish a grid for the solution. In cases where the distance between hub and shroud is great and there is a large change in flow direction within the rotor, however, the normals vary considerably in length and direction during the course of the calculations. Therefore, it becomes difficult to obtain a direct solution on the computer without resorting to intermediate graphical steps. The use of normals, however, is not essential to the method, and it appeared possible to overcome this difficulty by the use of a set of arbitrary curves from hub to shroud instead of streamline normals. These arbitrary curves will be hereinafter termed quasi-orthogonals. The quasi-orthogonals are not actually orthogonal to each streamline, but merely intersect every streamline across the width of the passage. The quasi-orthogonals remain fixed regardless of any change of streamlines. By using this technique, it appeared possible to develop a computer program that would calculate the velocity and pressure distributions without any intermediate graphical procedures even for turbomachines with wide passages and a change of direction from radial to axial within the rotor blade.

In view of these considerations, a method of analysis utilizing quasi-orthogonals in lieu of streamline normals was developed. This report presents the analysis method and contains a discussion of the numerical techniques required for obtaining solutions with a digital computer. The computer program developed during this study is included. As a numerical example of the application of the analysis method, a radial-inlet mixed-flow gas turbine of high specific speed is analyzed. Such a turbine, which may have application in gas turbine cycle space power systems, has a rotor-channel geometry for which this method, as compared to previous methods, can yield a quick and direct solution.

METHOD OF ANALYSIS

The analysis to be presented herein is basically the same as those presented in references 1, 2, and 4. As pointed out in the INTRODUCTION, the major difference is the use of fixed arbitrary quasi-orthogonals rather than streamline normals to establish a grid for the solution. Another difference is that the reference analyses were based completely on the assumption of isentropic flow, while in this analysis a correction for a loss in relative total pressure is included in the continuity equation to account for blade losses.

This analysis, as that of reference 4, is based on the assumption of a mean flow surface between blades. In general, this surface is assumed to be parallel to the mean blade surface, with arbitrary or empirical corrections made to take care of the difference between the flow angle and the blade angle at the inlet and at the outlet. One factor that is not accounted for by this assumption is that the actual mean stream surface twists considerably in a mixed-flow turbine. Despite this assumption, however, reference 3 shows the value of the analysis method by the improved performance of compressors redesigned in accordance with this assumption. Reference 5 shows that a two-dimensional solution for a particular compressor, when compared to a three-dimensional solution, gives values of the through-flow component of velocity that are of sufficient accuracy for engineering analysis. For convenience, the mean stream surface is projected on a plane containing the axis of rotation.

This plane is called the meridional plane. The projections of the streamlines on the meridional plane are called meridional streamlines.

Analytical Equations

Equations (1) and (2) give the velocity gradient along an arbitrary quasi-orthogonal in the meridional plane

$$\frac{dW}{ds} = \left(A \frac{dr}{ds} + B \frac{dz}{ds}\right)W + C \frac{dr}{ds} + D \frac{dz}{ds} + \left(\frac{dh_i'}{ds} - \omega \frac{d\lambda}{ds}\right)\frac{1}{W} \tag{1}$$

where

$$\left.\begin{aligned}
A &= \frac{\cos\alpha \cos^2\beta}{r_c} - \frac{\sin^2\beta}{r} + \sin\alpha \sin\beta \cos\beta \frac{\partial\theta}{\partial r} \\
B &= -\frac{\sin\alpha \cos^2\beta}{r_c} + \sin\alpha \sin\beta \cos\beta \frac{\partial\theta}{\partial z} \\
C &= \sin\alpha \cos\beta \frac{dW_m}{dm} - 2\omega \sin\beta + r \cos\beta\left(\frac{dW_\theta}{dm} + 2\omega \sin\alpha\right)\frac{\partial\theta}{\partial r} \\
D &= \cos\alpha \cos\beta \frac{dW_m}{dm} + r \cos\beta\left(\frac{dW_\theta}{dm} + 2\omega \sin\alpha\right)\frac{\partial\theta}{\partial z}
\end{aligned}\right\} \tag{2}$$

The coordinate system and the notation are shown in figures 1 and 2. (All symbols are listed in appendix A.) Equation (1) is derived in appendix B.

The value of the parameters h_i' and λ associated with a point inside the rotor is the value of that parameter at the inlet for the streamline which passes through that point. Then dh_i'/ds refers to the total enthalpy at the inlet as a function of the distance along the arbitrary meridional quasi-orthogonal near the point in question.

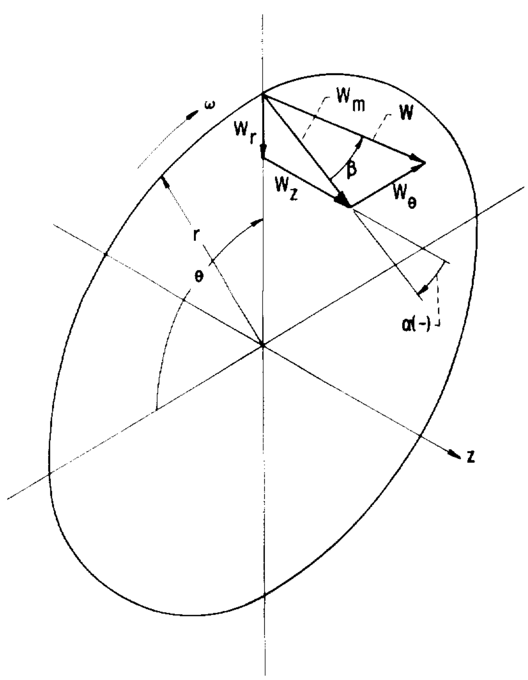

Figure 1. - Coordinate system and velocity components.

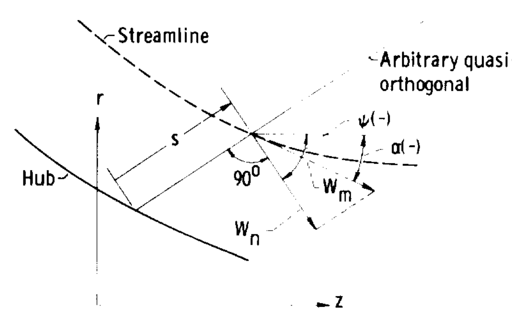

Figure 2. - Component of relative velocity W_n normal to arbitrary quasi-orthogonal.

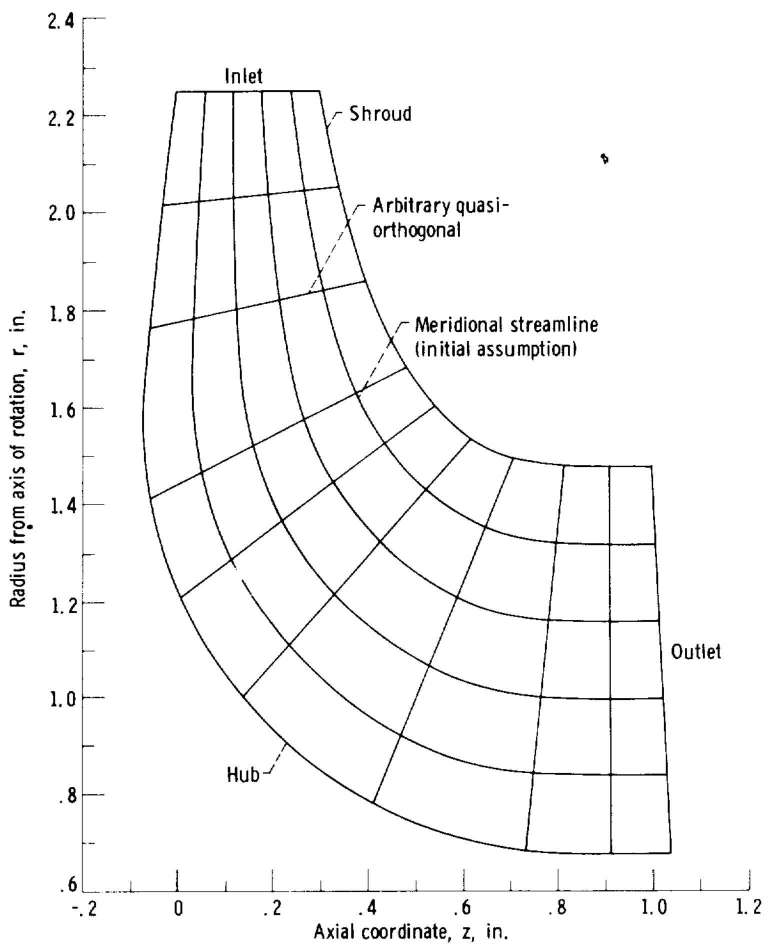

Figure 3. - Profile of rotor in numerical example.

In this analysis, the arbitrary quasi-orthogonals were chosen to be straight lines from hub to shroud (see fig. 3). At the inlet and outlet, the lines were chosen as the leading and trailing edges, respectively.

In addition to equation (1), which is a force equilibrium equation, the continuity equation must be satisfied. This is done by requiring that the calculated weight flow across any line from hub to shroud be equal to the specified weight flow through the turbomachine. For this the density must be known. If the velocity is known, the density may be calculated by equations (3) to (5) following. Equation (B9) is

$$h = h_i' - \omega\lambda + \frac{\omega^2 r^2 - W^2}{2} \tag{B9}$$

hence, assuming c_p constant results in

$$\frac{T}{T_i'} = \frac{h_i}{h_i'} = 1 - \frac{W^2 + 2\omega\lambda - \omega^2 r^2}{2c_p T_i'} \tag{3}$$

With $W = 0$,

$$\frac{T''}{T_i'} = 1 - \frac{2\omega\lambda - \omega^2 r^2}{2c_p T_i'} \tag{4}$$

For isentropic conditions,

$$\frac{\rho}{\rho_i'} = \left(\frac{T}{T_i'}\right)^{1/(\gamma-1)}$$

which gives the static density at any point once the velocity is known if inlet total conditions are specified.

To account for losses, it is necessary to make a correction to the above calculated density. One way to do this is to assume a loss in relative total pressure $\Delta p''$, which is a measure of the loss in efficiency. Then

$$\rho = \left(\frac{\rho}{\rho''}\right)\rho'' = \left(\frac{T}{T''}\right)^{1/(\gamma-1)} \frac{p''}{RT''} = \left(\frac{T}{T''}\right)^{1/(\gamma-1)} \frac{p''_{isen} - \Delta p''}{RT''}$$

$$= \left(\frac{T}{T''}\right)^{1/(\gamma-1)} \rho''_{isen} - \left(\frac{T}{T''}\right)^{1/(\gamma-1)} \frac{\Delta p''}{RT''}$$

$$= \left(\frac{T}{T''}\right)^{1/(\gamma-1)} \left(\frac{T''}{T'_i}\right)^{1/(\gamma-1)} \rho'_i - \left(\frac{T}{T''}\right)^{1/(\gamma-1)} \frac{\Delta p''}{RT''}$$

or

$$\rho = \left(\frac{T}{T'_i}\right)^{1/(\gamma-1)} \rho'_i - \left[\left(\frac{T}{T'_i}\right)\left(\frac{T'_i}{T''}\right)\right]^{1/(\gamma-1)} \frac{\Delta p''}{RT'_i}\left(\frac{T'_i}{T''}\right) \tag{5}$$

This gives the static density with a specified loss in relative total pressure. The temperature ratios can be calculated from equations (3) and (4). It is assumed that inlet total conditions are known.

Weight flow across a quasi-orthogonal can now be computed by

$$w = N \int_0^s \rho W_n r \, \Delta\theta \, ds \tag{6}$$

where $\Delta\theta$ is the angular distance between blades and W_n is the component of W normal to the surface of revolution generated by the fixed line. From figure 2, it can be seen that

$$W_n = W_m \cos (\psi - a) \tag{7}$$

To get $\Delta\theta$, use is made of the fact that

$$\Delta\theta = \frac{2\pi}{N} - \frac{t_\theta}{r} \tag{8}$$

where t_θ is the tangential thickness. If the thickness normal to the mean blade surface t_n is specified,

$$t_\theta^2 = t_n^2\left[1 + r^2\left(\frac{\partial\theta}{\partial z}\right)^2 + r^2\left(\frac{\partial\theta}{\partial r}\right)^2\right] \tag{9}$$

Note that here $\partial\theta/\partial r$ and $\partial\theta/\partial z$ refer to the mean blade shape and not the assumed mean stream surface. Equations (3) to (5) and (7) to (9) give the numerical data for equation (6), which can be integrated by use of a spline fit approximation (see appendix C).

With the velocities on the mean stream surface calculated, blade surface velocities can be calculated by any of several approximate methods. One method that gives good results, when compared with a relaxation solution of the potential flow equation for a surface of revolution, is based on absolute irrotational flow and linear velocity distribution between blades. The following equations based on these assumptions are equations (16) and (17) of reference 6.

$$\left.\begin{aligned} W_t &= \frac{\cos\beta_l \cos\beta_t}{\cos\beta_l + \cos\beta_t}\left\{\frac{2W}{\cos\beta_l} + r\omega(\tan\beta_l - \tan\beta_t) + \frac{d}{dm}\left[(r\omega + W\sin\beta)r\,\Delta\theta\right]\right\} \\ W_l &= 2W - W_t \end{aligned}\right\} \tag{10}$$

The derivative can be evaluated by use of a spline fit curve.

Equations (3) to (5) and the equation of state can be used to calculate the static temperature, density, and pressure on the blade surfaces.

Numerical Techniques and Procedure

The first step in the analysis is the numerical evaluation of the parameters α, β, r_c, $\partial\theta/\partial r$, $\partial\theta/\partial z$, dr/ds, dz/ds, dW_m/dm, and dW_θ/dm for use in equations (1) and (2). In order to evaluate the parameters α, β, and r_c a streamline geometry must be established. First fixed straight lines (quasi-orthogonals) are drawn from hub to shroud along which the velocity gradient for an assumed stream surface will be determined. For an initial approximation to the streamlines, each quasi-orthogonal can be divided into a number of equal spaces, as shown in figure 3. The success of the method is based on the fact that, for a reasonable assumed streamline pattern, the geometrical streamline parameters involved are not too different from those of the final solution.

By means of a spline fit approximation (see appendix C), dr/dz and d^2r/dz^2 can be determined at each of the points established. Then

$$\alpha = \tan^{-1}\frac{dr}{dz}$$

and

$$\frac{1}{r_c} = \frac{\dfrac{d^2r}{dz^2}}{\left[1 + \left(\dfrac{dr}{dz}\right)^2\right]^{3/2}} \tag{11}$$

The reciprocal of the radius of curvature (the curvature) is computed to avoid division by zero in case $d^2r/dz^2 = 0$.

For the remaining parameters, the mean stream surface $\theta = \theta(r,z)$ between blades is needed; it must be given in such a manner that $\partial\theta/\partial r$ and $\partial\theta/\partial z$ can be determined at any given point. The spline fit curve can assist in this. When $\partial\theta/\partial r$ and $\partial\theta/\partial z$ are known, β can be calculated from

$$\tan \beta = r \frac{d\theta}{dm} = r\left(\frac{\partial\theta}{\partial r}\frac{dr}{dm} + \frac{\partial\theta}{\partial z}\frac{dz}{dm}\right) = r\left(\frac{\partial\theta}{\partial r}\sin \alpha + \frac{\partial\theta}{\partial z}\cos \alpha\right) \tag{12}$$

For the initial calculation, W may be assumed constant throughout the rotor. From figure 1, it is seen that

$$W_m = W \cos \beta$$

and

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

Since the distance along the meridional streamline m is known, dW_m/dm and dW_θ/dm can then be determined by the spline fit curve. Since dr/ds and dz/ds are determined by the angles of the quasi-orthogonals, all the quantities necessary for the calculation dW/ds from equation (1), except W itself, are now determined.

The next step is the numerical integration of equation (1), which is in the form

$$\frac{dW}{ds} = f(W,s)$$

where f is known only for a finite number of values of s. For a given initial velocity on, say the hub, the velocity distribution along the quasi-orthogonal can be approximated by

$$W_{j+1} = W_j + \left(\frac{dW}{ds}\right)_j \Delta s$$

where the subscripts denote the number of the streamline, and Δs is the distance along the quasi-orthogonal between streamlines. For an improved estimate, a Runge-Kutta method can be used. The following is a particular Runge-Kutta method that is well adapted for this case. Let

$$
\left.
\begin{aligned}
W^{*}_{j+1} &= W_j + \left(\frac{dW}{ds}\right)_j \Delta s \\
W^{**}_{j+1} &= W_j + \left(\frac{dW}{ds}\right)_{j+1} \Delta s \\
\text{then} \qquad W_{j+1} &= \frac{W^{*}_{j+1} + W^{**}_{j+1}}{2}
\end{aligned}
\right\} \tag{13}
$$

This avoids an obvious bias due to using the derivative at the beginning of the interval (see fig. 4) and gives a higher order approximation. For a mathematical analysis and error estimate, see reference 7. For the calculation of the quantity $(dW/ds)_j$, equation (1) is used with the parameters calculated for the j^{th} streamline and W_j. To calculate $(dW/ds)_{j+1}$, the parameters calculated for the $(j + 1)^{st}$ streamline are used and W^{*}_{j+1} is used for the velocity W in equation (1).

It should be noted that this method of integrating equation (1) involves much less computation than solving equation (1) directly and then numerically evaluating the resulting integral (e.g., eq. (9) in ref. 1). This is especially helpful for hand computation and is also helpful in simplifying computer programing. Accuracy is probably comparable; the method used here certainly gives satisfactory accuracy if the streamlines are spaced closely enough so that the velocity does not vary more than about 30 percent between streamlines. In the numerical example, the results using five streamlines did not differ appreciably from those using twenty streamlines.

Completing this computation for a quasi-orthogonal from hub to shroud results in the complete velocity distribution along that line based on the initial estimate of the velocity on the hub. Equations (3) to (5) and (7) to (9) can be used to compute the integrand in equation (6). The numerical integration can be performed by use of a spline fit approximation (see appendix C). The computed total weight flow is then compared with the actual weight flow. If the computed weight flow is too small, the velocity on the hub is increased, and vice versa. Then the velocity distribution and the weight flow are recalculated. The computed weight flow is a function of the assumed hub velocity; therefore, after two values of weight flow are computed, linear interpolation or extrapolation can be used to get an improved estimate for the hub velocity. A few iterations will determine the hub velocity that will give the correct weight flow.

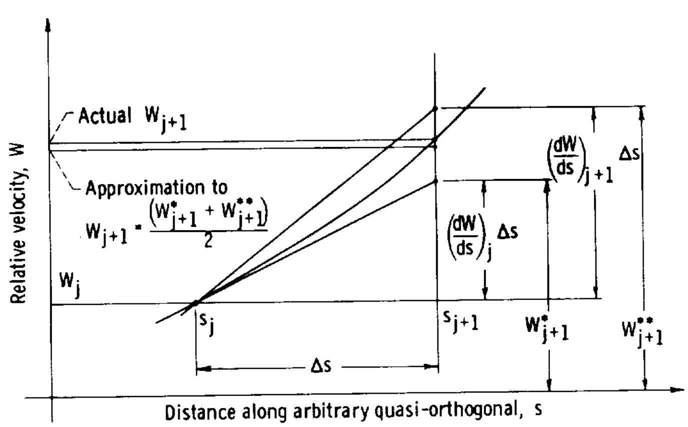

Figure 4. - Approximation to solution of differential equation $dW/ds = f(W, s)$.

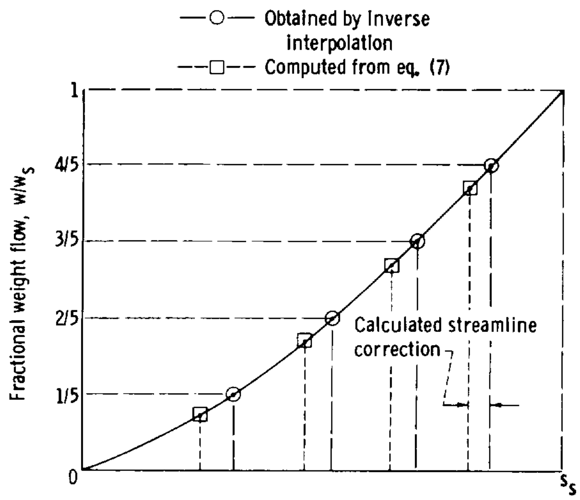

Figure 5. - Weight flow distribution along quasi-orthogonal.

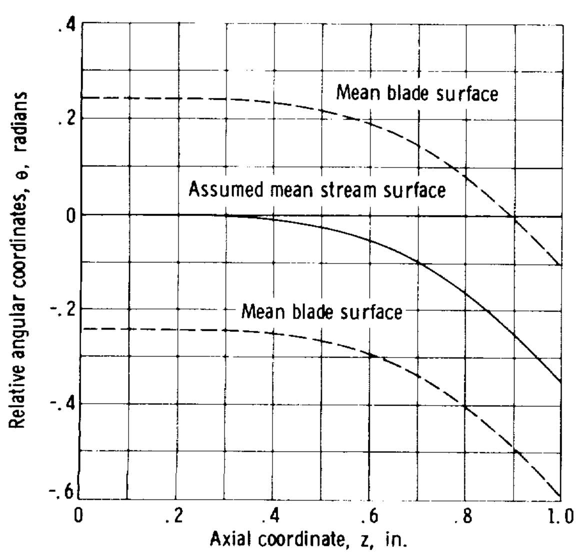

Figure 6. - Mean stream surface for numerical example.

From equation (6) the weight flow distribution along the quasi-orthogonal from hub to shroud can also be obtained. Inverse interpolation (by a spline fit approximation), can be used to determine the spacing of the streamlines on the quasi-orthogonal that will give equal weight flow between any two adjacent streamlines (see fig. 5). When this is done for every quasi-orthogonal from inlet to outlet, a new estimate for the meridional streamline pattern is obtained. This pattern, together with the calculated velocity distribution, can then be used for further iterations; however, using this estimate generally results in overcorrection. Therefore, only a fraction of the calculated correction was made. Another problem is the tendency for the newly computed streamline to be less smooth than the previous streamline. If a computation is based on a set of streamlines that are not extremely smooth, the calculated streamline corrections become erratic. Thus it is important to be sure that the streamline estimate to be used for the following iteration be as smooth as possible. Several methods of accomplishing this have been tried. The method that was successful for the cases tried was to use a streamline correction at each point of one-tenth the calculated correction. With this the streamlines remained smooth, and a solution was reached in a single computer run, requiring about 50 iterations. Computer execution time was 2 minutes (on the IBM 7094). The computer program used for this together with the listing of computed results for the numerical example are given in appendix D.

NUMERICAL EXAMPLE

The procedure outlined herein has been programed for solution on a digital computer. The following results were obtained for a particular turbine. The hub-shroud profile and quasi-orthogonals are shown in figure 3, together with the equally spaced streamlines used for the initial assumption. The blade has radial blade elements with the blade shape indicated in figure 6. There are 13 blades, with no splitter blades. The rotational speed was 51,500 rpm, and the fluid was air. The weight flow was 0.984 pound per second, inlet total temperature was 592° R, V_θ at the inlet was 1010 feet per second, and the total inlet pressure was 42.5 pounds per square inch. The normal blade thickness was given by means of tabulated values on a grid. Blade thickness at any given point was obtained by linear interpolation. It was assumed that h_i' and λ are both constant from hub to shroud.

At the inlet, the flow surface was assumed to deviate from the blade surface in order to agree with the flow direction coming into the rotor. This angle at the inlet was -35°. The meridional streamlines are approximately radial at the inlet, so that the stream surface was assumed to be independent of z where it deviates from the blade surface. The θ coordinate was assumed to vary as the cube of r (and independent of z) for a given distance from the inlet. Let r_b denote the radius where the mean stream surface is assumed to deviate from the mean blade shape. Equation (13) of reference 6 gives an approximate equation for determining r_b, which may be written as follows:

$$r_b = r_i \, e^{-0.71 \, \Delta\theta}$$

The equation of the stream surface for $r \geq r_b$ is

$$\theta = - \frac{(r - r_b)^3 \tan \beta_i}{3r_i(r_i - r_b)^2}$$

which, when differentiated, becomes

$$\frac{\partial\theta}{\partial r} = - \frac{(r - r_b)^2 \tan \beta_i}{r_i(r_i - r_b)^2} \tag{14}$$

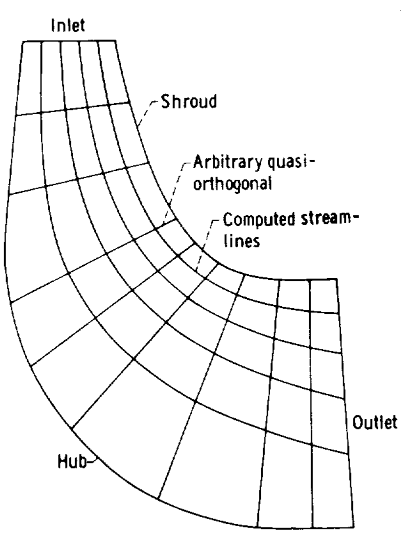

Figure 7. - Meridional projection of mean stream surface for numerical example.

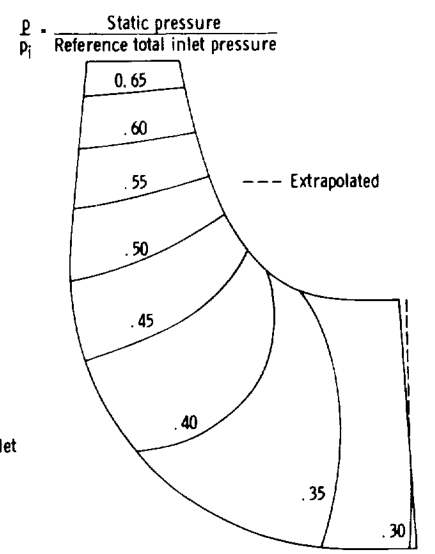

Figure 8. - Static pressure contours on mean stream surface for numerical example.

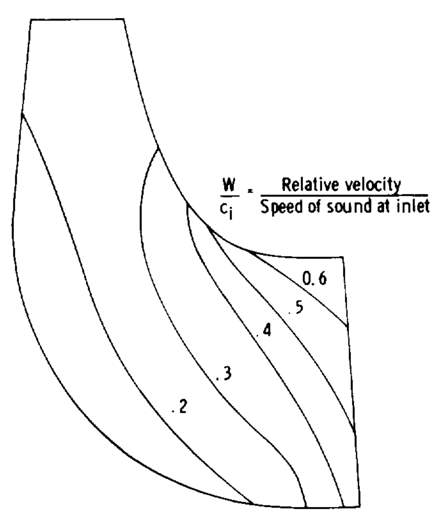

Figure 9. - Relative velocity contours of mean stream surface for numerical example.

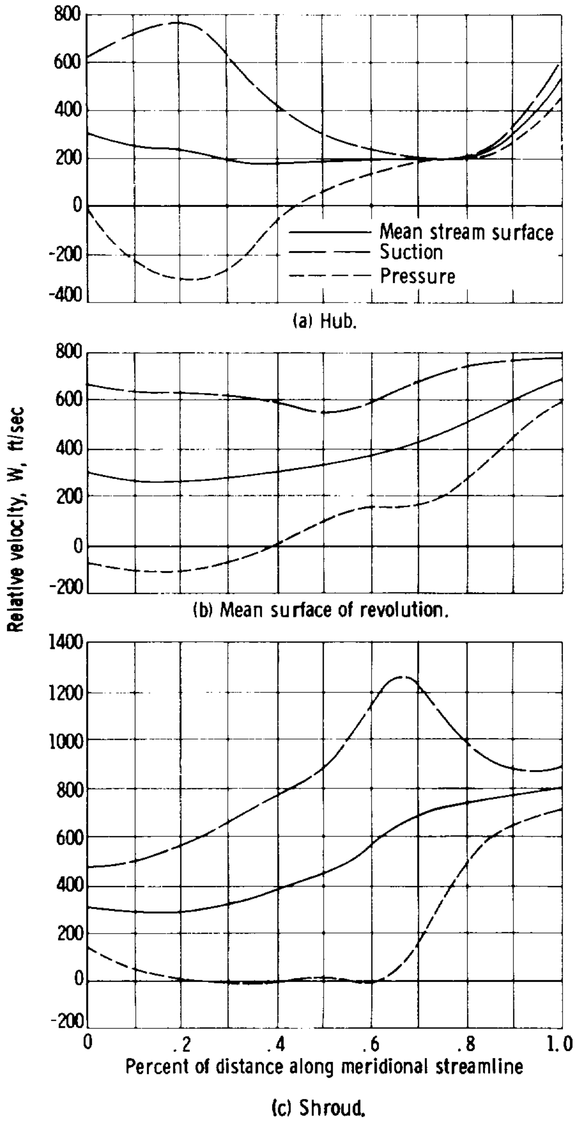

Figure 10. - Blade loading diagram for numerical example.

This is used in equations (2) and (12) when $r > r_b$ but not in equation (9), since equation (9) refers to the blade shape. For the numerical example, r_b is about 1.60 inches. At the outlet it is assumed that the mean stream surface would follow the blade. There was also assumed to be a 2.5-pound-per-square-inch loss of total relative pressure, varying linearly from inlet to outlet.

The calculated streamlines are shown in figure 7. Since the solution is restricted to the rotor blade, the streamlines near the outlet do not show the effect of downstream geometry. Some of the other calculated information is shown in the figures following. Figures 8 and 9 show lines of constant pressure and constant relative velocity, while figure 10 shows blade loading diagrams at the hub, the mean surface of revolution, and the shroud.

Figure 8 shows that the pressure level is always decreasing in the direction of flow. This is, of course, normal for a radial turbine. In figure 9, it is seen that velocities are generally increasing, except along the hub near the inlet where they decrease slightly. Though not desirable, this can be tolerated because of the favorable pressure gradient. More serious are the negative velocities in the blade loading diagrams (fig. 10). This indicates an eddy on the trailing surface of the blade near the inlet, which would result in turbulence and mixing losses. Also, a severe decreasing velocity gradient is indicated on the suction surface near the hub and near the shroud. This could lead to flow separation with accompanying high losses.

CONCLUDING REMARKS

A method of analysis of turbomachines is presented that is suitable for computer programing. The method and the results are similar to that obtained by other streamline analysis methods (e.g., refs. 1, 2, and 4). The difference here is that velocity gradients are given along arbitrary quasi-orthogonals, rather than the normal to the streamlines as has been done in previous methods. The value of the method lies in the fact that a solution can be obtained in a single computer run even for cases where the distance between hub and shroud is great and there is a change of direction from radial to axial within the rotor. The method was successfully applied to a turbine with this type of geometry. These results are given as a numerical example, and the Fortran computer program is included in appendix D.

A more accurate hub-to-shroud analysis could be made by using information from a blade-to-blade streamline analysis. A blade-to-blade analysis would give a better approximation to the mean stream surface and also would give the blade-to-blade streamline spacing. Continuity would then be checked between the two hub-to-shroud stream surfaces instead of between blades.

Lewis Research Center
National Aeronautics and Space Administration
Cleveland, Ohio, September 15, 1964

APPENDIX A

SYMBOLS

A	parameter, eq. (2)
a	parameter, eq. (B14)
B	parameter, eq. (2)
b	parameter, eq. (B14)
C	parameter, eq. (2)
c	parameter, eq. (B14)
c_i	stagnation speed of sound at inlet, ft/sec
c_p	specific heat at constant pressure, (ft)(lb)/(slug)($^{\circ}$R)
D	parameter, eq. (2)
f	any function
g	acceleration due to gravity, ft/sec^2
h	static enthalpy, (ft)(lb)/slug
m	distance along meridional streamline, ft
N	number of blades
n	distance along normal to meridional streamline, ft
p	absolute static pressure, lb/sq ft
$\Delta p''$	loss in relative total pressure between inlet and any point
q	distance along arbitrary three dimensional curve, ft
R	gas constant, (ft)(lb)/(slug)($^{\circ}$R)
r	radius from axis of rotation, ft
r_b	radius at which assumed stream surface is tangent to mean blade shape
r_c	radius of curvature of meridional streamline, ft
s	distance along arbitrary quasi-orthogonal in meridional plane, ft

T temperature, $^{\circ}R$

t time, sec

t_n blade thickness normal to blade mean surface, ft

t_θ blade thickness in circumferential direction, ft

$\bar{u}$ unit vector

V absolute fluid velocity, ft/sec

W relative fluid velocity, ft/sec

w weight flow crossing surface of revolution generated by quasi-orthogonal between hub and given point on quasi-orthogonal

x x-coordinate

y y-coordinate

z axial coordinate

α angle between meridional streamline and z-axis, radians

β angle between relative velocity vector and meridional plane, radians

γ ratio of specific heat

θ relative angular coordinate, radians

$\Delta\theta$ angle between blade surfaces at given point, radians

λ prerotation $r_i V_{\theta_i}$, sq ft/sec

ρ mass density, slugs/cu ft

φ absolute angular coordinate, radians

ψ angle between quasi-orthogonal and radial direction, radians

ω rotational speed, radians/sec

Subscripts:

i inlet

isen isentropic

j number of streamline

l leading surface

m component in direction of meridional streamline

n normal component

r radial component

s shroud

t trailing surface

x x-component

y y-component

z axial component

θ tangential component

Superscripts:

$\bar{}$ vector quantity

$'$ absolute stagnation condition

$''$ relative stagnation condition

APPENDIX B

DERIVATION OF THE VELOCITY GRADIENT EQUATION

Euler's force equation for a nonviscous fluid is

$$\frac{d\overline{V}}{dt} = -\frac{1}{\rho}\nabla p \tag{B1}$$

This is simply expressed as three scalar equations in fixed rectangular coordinates x, y, and z. To reduce the problem to a steady-state condition, equation (B1) should be expressed in terms of the relative velocity $\overline{W}$ and the pressure gradient relative to a rotating cylindrical coordinate system r, θ, and z. The notation $\overline{u}_x$ is used to denote a unit vector in the x direction; similar notation is used for the other coordinates. It should be noted that the directions of the vectors $\overline{u}_r$ and $\overline{u}_\theta$ are functions of t as well as of φ. It is seen from figure 11 that

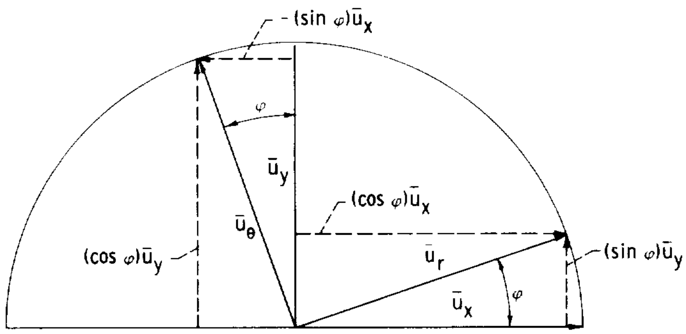

Figure 11. - Relations between unit basis vectors in absolute rectangular coordinates and relative cylindrical coordinates.

$$\left.\begin{aligned} \overline{u}_r &= (\cos\varphi)\overline{u}_x + (\sin\varphi)\overline{u}_y \\ \overline{u}_\theta &= -(\sin\varphi)\overline{u}_x + (\cos\varphi)\overline{u}_y \end{aligned}\right\} \tag{B2}$$

Differentiating equation (B2) results in

$$\left.\begin{aligned} \frac{d\overline{u}_r}{dt} &= -\sin\varphi\frac{d\varphi}{dt}\overline{u}_x + \cos\varphi\frac{d\varphi}{dt}\overline{u}_y = \frac{d\varphi}{dt}\overline{u}_\theta = \frac{V_\theta}{r}\overline{u}_\theta \\ \frac{d\overline{u}_\theta}{dt} &= -\cos\varphi\frac{d\varphi}{dt}\overline{u}_x - \sin\varphi\frac{d\varphi}{dt}\overline{u}_y = -\frac{V_\theta}{r}\overline{u}_r \end{aligned}\right\} \tag{B3}$$

Since $\overline{V} = V_r\overline{u}_r + V_\theta\overline{u}_\theta + V_z\overline{u}_z$, equation (B3) can be used to get

$$\begin{aligned} \frac{d\overline{V}}{dt} &= \frac{d(V_r\overline{u}_r)}{dt} + \frac{d(V_\theta\overline{u}_\theta)}{dt} + \frac{d(V_z\overline{u}_z)}{dt} \\ &= \frac{dV_r}{dt}\overline{u}_r + \frac{V_rV_\theta}{r}\overline{u}_\theta + \frac{dV_\theta}{dt}\overline{u}_\theta - \frac{V_\theta^2}{r}\overline{u}_r + \frac{dV_z}{dt}\overline{u}_z \\ &= \left(\frac{dV_r}{dt} - \frac{V_\theta^2}{r}\right)\overline{u}_r + \frac{1}{r}\frac{d(rV_\theta)}{dt}\overline{u}_\theta + \frac{dV_z}{dt}\overline{u}_z \end{aligned} \tag{B4}$$

The pressure gradient will now be expressed in the relative coordinates. For the fixed cylindrical coordinates r, φ, and z,

$$\nabla p = \frac{\partial p}{\partial r} \bar{u}_r + \frac{1}{r} \frac{\partial p}{\partial \varphi} \bar{u}_\varphi + \frac{\partial p}{\partial z} \bar{u}_z$$

Note that actually $\bar{u}_\varphi = \bar{u}_\theta$, when φ and θ refer to the same point ($\bar{u}_\varphi$ varies with time, since φ varies with time for constant θ). Also $\partial p/\partial \varphi = \partial p/\partial \theta$, since $\partial \varphi/\partial \theta = 1$. This gives

$$\nabla p = \frac{\partial p}{\partial r} \bar{u}_r + \frac{1}{r} \frac{\partial p}{\partial \theta} \bar{u}_\theta + \frac{\partial p}{\partial z} \bar{u}_z \tag{B5}$$

Noting that $W_r = V_r$, $W_z = V_z$ and $V_\theta = W_\theta + \omega r$, substituting equations (B4) and (B5) in equation (B1), and equating coefficients of $\bar{u}_r$, $\bar{u}_\theta$, and $\bar{u}_z$ result in

$$\frac{dW_r}{dt} - \frac{(W_\theta + \omega r)^2}{r} = -\frac{1}{\rho} \frac{\partial p}{\partial r} \tag{B6a}$$

$$\frac{1}{r} \frac{d(rW_\theta + \omega r^2)}{dt} = -\frac{1}{\rho r} \frac{\partial p}{\partial \theta} \tag{B6b}$$

$$\frac{dW_z}{dt} = -\frac{1}{\rho} \frac{\partial p}{\partial z} \tag{B6c}$$

Now an expression for the directional derivative of the relative velocity in any direction will be derived. The parameters in this expression require the knowledge of the streamline passing through a given point; however, once the streamline is known, the velocity gradient in any direction can be computed.

If q denotes the distance along an arbitrary curve, the directional derivative of the pressure p along this curve is

$$\frac{dp}{dq} = \frac{\partial p}{\partial r} \frac{dr}{dq} + \frac{\partial p}{\partial \theta} \frac{d\theta}{dq} + \frac{\partial p}{\partial z} \frac{dz}{dq}$$

Using equations (B6) gives

$$-\frac{1}{\rho} \frac{dp}{dq} = \left[\frac{dW_r}{dt} - \frac{(W_\theta + \omega r)^2}{r}\right] \frac{dr}{dq} + \frac{d(rW_\theta + \omega r^2)}{dt} \frac{d\theta}{dq} + \frac{dW_z}{dt} \frac{dz}{dq} \tag{B7}$$

Equation (B7) is an expression for the pressure gradient in the q direction. It is necessary to find a relation between the velocity gradient and the pressure gradient. This is easily done under the assumption that the flow is isentropic, so that

$$\frac{dp}{\rho} = dh$$

Now multiplying equation (B6a) by $W_r = dr/dt$, equation (B6b) by $W_\theta = r\, d\theta/dt$, and equation (B6c) by $W_z = dz/dt$, then adding and combining terms yield

$$\frac{1}{2}\frac{dW^2}{dt} = \omega^2 r \frac{dr}{dt} - \frac{1}{\rho}\frac{dp}{dt} = \frac{\omega^2}{2}\frac{d(r^2)}{dt} - \frac{dh}{dt}$$

which is the energy equation for isentropic flow. Integrating from the inlet along a streamline results in

$$W^2 - W_i^2 = \omega^2(r^2 - r_i^2) - 2(h - h_i) \tag{B8}$$

Since $V_m = W_m$ and $V_\theta = W_\theta + \omega r$,

$$V^2 - V_\theta^2 = W^2 - W_\theta^2 = W^2 - V_\theta^2 + 2V_\theta\omega r - \omega^2 r^2$$

or

$$V^2 = W^2 + 2V_\theta\omega r - \omega^2 r^2$$

hence, at the inlet,

$$h_i' = h_i + \frac{V_i^2}{2} = h_i + \frac{W_i^2 + 2\omega\lambda - \omega^2 r_i^2}{2}$$

Substituting this for h_i in equation (B8) gives

$$h = h_i' - \omega\lambda + \frac{\omega^2 r^2 - W^2}{2} \tag{B9}$$

Since the flow is assumed isentropic, differentiating results in

$$\frac{1}{\rho}\frac{dp}{dq} = \frac{dh}{dq} = \frac{dh_i'}{dq} - \omega\frac{d\lambda}{dq} + \omega^2 r\frac{dr}{dq} - W\frac{dW}{dq}$$

Substituting this equation in equation (B7) yields

$$\frac{dW}{dq} = \frac{1}{W}\frac{dh_i'}{dq} - \frac{\omega}{W}\frac{d\lambda}{dq} + \left[\frac{\omega^2 r}{W} + \frac{1}{W}\frac{dW_r}{dt} - \frac{(W_\theta + \omega r)^2}{rW}\right]\frac{dr}{dq} + \frac{1}{W}\frac{d(rW_\theta + \omega r^2)}{dt}\frac{d\theta}{dq} + \frac{1}{W}\frac{dW_z}{dt}\frac{dz}{dq} \tag{B10}$$

Note that $W_m = W\cos\beta$ and

$$\frac{d\alpha}{dt} = \frac{d\alpha}{dm}\frac{dm}{dt} = \frac{W_m}{r_c}$$

Using this and differentiating $W_r = W_m \sin \alpha$ and $W_z = W_m \cos \alpha$ result in

$$\left.\begin{aligned}
\frac{dW_r}{dt} &= \frac{W^2 \cos^2\beta \cos \alpha}{r_c} + W \sin \alpha \cos \beta \frac{dW_m}{dm} \\
\frac{dW_z}{dt} &= -\frac{W^2 \cos^2\beta \sin \alpha}{r_c} + W \cos \alpha \cos \beta \frac{dW_m}{dm}
\end{aligned}\right\} \tag{B11}$$

Also,

$$\frac{1}{W}\frac{d(rV_\theta)}{dt} = \frac{1}{W}\frac{d(rW_\theta + r^2\omega)}{dt} = r \cos \beta \frac{dW_\theta}{dm} + W \sin \alpha \cos \beta \sin \beta + 2r\omega \sin \alpha \cos \beta \tag{B12}$$

Using equations (B11) and (B12) and the fact that $V_\theta = W_\theta + \omega r$, and $W_\theta = W \sin \beta$ in equation (B10) gives

$$\frac{dW}{dq} = a \frac{dr}{dq} + b \frac{dz}{dq} + c \frac{d\theta}{dq} + \frac{1}{W}\left(\frac{dh_i'}{dq} - \omega \frac{d\lambda}{dq}\right) \tag{B13}$$

where

$$\left.\begin{aligned}
a &= \frac{W \cos^2\beta \cos \alpha}{r_c} - \frac{W \sin^2\beta}{r} + \sin \alpha \cos \beta \frac{dW_m}{dm} - 2\omega \sin \beta \\
b &= -\frac{W \cos^2\beta \sin \alpha}{r_c} + \cos \alpha \cos \beta \frac{dW_m}{dm} \\
c &= W \sin \alpha \cos \beta \sin \beta + r \cos \beta\left(\frac{dW_\theta}{dm} + 2\omega \sin \alpha\right)
\end{aligned}\right\} \tag{B14}$$

The meridional plane analysis is concerned with the projection of the curve q onto the meridional plane. This projected curve will be the quasi-orthogonal. Letting s denote the distance along this meridional projection, then

$$\frac{dW}{ds} = \frac{dW}{dq}\cdot\frac{dq}{ds} = a \frac{dr}{ds} + b \frac{dz}{ds} + c \frac{d\theta}{ds} + \frac{1}{W}\left(\frac{dh_i'}{ds} - \omega \frac{d\lambda}{ds}\right) \tag{B15}$$

If the line s is a normal to the meridional streamline, then $s = n$, $dr/ds = dr/dn = \cos \alpha$, and $dz/ds = dz/dn = -\sin \alpha$, and equation (B15) reduces to equation (B24) of reference 4.

The quantities dr/ds and dz/ds in equation (B15) are determined by the parametric equations for the arbitrary curve in the meridional plane, $r = r(s)$, and $z = z(s)$. The quantity $d\theta/ds$ refers to the change in θ in the actual curve q. If the curve q lies on a hub-to-shroud surface, which can be defined by $\theta = \theta(r,z)$, then

$$\frac{d\theta}{ds} = \frac{\partial\theta}{\partial r}\frac{dr}{ds} + \frac{\partial\theta}{\partial z}\frac{dz}{ds} \tag{B16}$$

By substituting equations (B14) and (B16), equation (B15) can be rewritten in the following form:

$$\frac{dW}{ds} = \left(A\frac{dr}{ds} + B\frac{dz}{ds}\right)W + C\frac{dr}{ds} + D\frac{dz}{ds} + \left(\frac{dh_i'}{ds} - \omega\frac{d\lambda}{ds}\right)\frac{1}{W} \tag{1}$$

where

$$\left.\begin{aligned}
A &= \frac{\cos\alpha\cos^2\beta}{r_c} - \frac{\sin^2\beta}{r} + \sin\alpha\sin\beta\cos\beta\frac{\partial\theta}{\partial r} \\
B &= -\frac{\sin\alpha\cos^2\beta}{r_c} + \sin\alpha\sin\beta\cos\beta\frac{\partial\theta}{\partial z} \\
C &= \sin\alpha\cos\beta\frac{dW_m}{dm} - 2\omega\sin\beta + r\cos\beta\left(\frac{dW_\theta}{dm} + 2\omega\sin\alpha\right)\frac{\partial\theta}{\partial r} \\
D &= \cos\alpha\cos\beta\frac{dW_m}{dm} + r\cos\beta\left(\frac{dW_\theta}{dm} + 2\omega\sin\alpha\right)\frac{\partial\theta}{\partial z}
\end{aligned}\right\} \tag{2}$$

Equation (1) is written in a form that is convenient for numerical solution.

APPENDIX C

USE OF SPLINE FIT CURVES

If a set of function values corresponding to a set of arguments is given, there are several ways a curve can be fitted through these values so as to approximate the original function with these values. The classical way is by an n^{th}-degree polynomial for $n + 1$ points. This may not be satisfactory, however, for a large number of points, especially for computing derivatives or curvature at end points. Another technique is to use fewer points to determine some sort of piecewise polynomial, but this does not lead to a smooth curve. A method that has received much attention recently is the piecewise cubic, with matching first and second derivatives, commonly referred to as a spline fit curve. Since for small slopes, the second derivative approximates the curvature of a function, the strain energy of a spline can be approximately minimized by minimizing $\int [f''(x)]^2 \, dx$, where $f(x)$ denotes the curve described by the spline. The spline fit curve has this property. This is proven in reference 8. Thus the spline fit curve is a mathematical expression for the shape taken by an idealized spline passing through the given points. In reference 8, a simple procedure is outlined for determining the spline fit curve when the coordinates of the points are given together with two arbitrary end conditions. The end condition actually used in the computer program was that the second derivative at an end point is one-half the second derivative at the next point. This is equivalent to bending the spline beyond the last point slightly, instead of just letting it be straight. The spline fit curve provided a simple analytical method of determining many of the parameters in the equations. The spline fit curve was used to determine first and second derivatives, curvature, interpolated function values, interpolated derivatives, and for integration.

One further point concerning the spline fit should be mentioned; that is, the approximation to an actual spline curve is dependent on the slope not being too large. Experimentally, good results are obtained if the absolute value of the slope is not greater than one. In applying this method to streamlines on a radial turbine, there is a problem since the angle may be around -90° at the inlet. This is easily overcome by rotating the coordinate axes 45° so that the maximum slope is about one.

APPENDIX D

FORTRAN PROGRAM USED FOR NUMERICAL EXAMPLE

Description of Main Program

The FORTRAN program listed herein is the one used in the numerical example. It is written in FORTRAN IV and was run on an IBM 7094 digital computer. The program closely follows the steps given in the section on numerical procedure. The list of program variables preceding the program indicates the equation that is used to calculate a variable or the equation in which it is used. In the program, the number of the streamline is denoted by K and the number of the quasi-orthogonal by I. The inlet or the hub is denoted by 1. The program is written so that all linear measurements are in inches, angles are in degrees, and pressure is in pounds per square inch for both input and output. Units are changed to feet and radians for computation in the program. All other quantities are in the units specified in appendix A.

It will be noted that a complete listing of input data cards is printed out. In the sample program, for example, the listing gives all the data used as input for the program. All input statements precede the comment card END OF INPUT STATEMENTS.

Program Variables and Definitions

A	temporary storage
AB(J)	temporary storage
AC(J)	temporary storage
AD(J)	temporary storage
AL(I,K)	α
ALM	λ (input variable)
AR	R (input variable)
B	temporary storage
BA(K)	total weight flow between hub and K^{th} streamline
BCDP	integer (input variable); 1 will give DN, WA, Z, and R as output on cards in binary form after final iteration, for use as input for alternate conditions; 0 will cause this to be omitted
BETA(I,K)	β

BETAD	β_l, eq. (10)
BETAT	β_t, eq. (10)
BETIN	β_i (input variable), eq. (14)
C	temporary storage
CAL(I,K)	$\cos \alpha$
CBETA(I,K)	$\cos \beta$
CEF	$\tan \beta / r_i(r_i - r_b)^2$, eq. (14)
CI	c_i
CORFAC	percentage of calculated streamline correction to be used for next iteration (input variable)
COSBD	$\cos \beta_l$, eq. (10)
COSBT	$\cos \beta_t$, eq. (10)
CP	c_p
CURV(I,K)	$1/r_c$
DELBTA(I)	$\beta_t - \beta_l$, eq. (10)
DELTA	calculated streamline correction (fig. 5)
DENSTY	ρg
DN(I,K)	distance along quasi-orthogonal from hub
DRDM(I)	$\frac{d}{dm}[(r\omega + W \sin \beta) r \, \Delta\theta]$, eq. (10)
DTDR(I)	$\partial\theta/\partial r$, eq. (2) and (12)
DTDZ(I)	$\partial\theta/\partial z$, eq. (2), (9), and (12)
DWMDM(I)	dW_m/dm, eq. (2)
DWTDM(I)	dW_θ/dm, eq. (2)
E	temporary storage
ERROR	maximum calculated streamline correction for present iteration (fig. 5)
ERROR1	ERROR from previous iteration

EXPON	$1/(\gamma-1)$, eq. (5)
G	temporary storage
GAM	γ (input variable)
HR	increment along quasi-orthogonal in r-direction
HZ	increment along quasi-orthogonal in z-direction
I	subscript to indicate number of quasi-orthogonal, 1 at inlet and MX at outlet
IND	code number for use by subroutine CONTIN
ITER	number of iterations to be performed after ERROR is less than TOLER or after ERROR has started to increase (input variable); if ITER = 0, data will be printed for every iteration; if ITER > 0, data will be printed only for final iteration
J	subscript
K	subscript to indicate number of streamline, 1 at hub and KMX at shroud
KMX	number of streamlines (input variable)
KMXM1	KMX - 1
MR	number of r values of TN in thickness table (input variable)
MTHTA	number of values of THTA in table of θ against z (input variable)
MX	number of fixed lines (input variable)
MZ	number of z values of TN in thickness table (input variable)
NPRT	data is listed for every $(\mathrm{NPRT})^{th}$ streamline (input variable)
NULL	dummy variable, not used
OMC	1. - CORFAC
PLOSS	$\Delta p''$ at outlet (input variable), eq. (5)
PRS(I,K)	p
PSI	ψ, eq. (7)
R(I,K)	r

RB	r_b (input variable), eq. (14)
RC	$1/r_c$
RH(I)	r-coordinate of hub (input variable)
RHO	$\rho_i' g$ (input variable)
ROOT	$\sqrt{2}$
RS(I)	r-coordinate of shroud (input variable)
RUNO	integer, run number
SA(I,K)	A, eq. (2)
SAL(I,K)	sin α
SB(I,K)	C, eq. (2)
SBETA(I,K)	sin β
SC(I,K)	B, eq. (2)
SD(I,K)	D, eq. (2)
SFACT	blade multiplier to allow for splitter blades (input variable)
SM(I,K)	distance from inlet along meridional streamline
SRW	integer (input variable) that will cause subroutines to write out data for certain values, used in debugging; SRW = 13 causes SPLINE to write
T	t_n (interpolated)
TEMP	T_i' (input variable)
THTA(J)	θ (as function of z) (input variable), blade shape (fig. 6)
TN(J,K)	t_n (input variable), first subscript refers to z-coordinate, second subscript refers to r-coordinate
TOLER	if maximum calculated streamline correction is less than TOLER, iterations are considered to have converged and desired output is printed (input variable)
TP	$r\,\partial\theta/\partial z$
TPP1P	T''/T_i', eq. (4)

TQ $r\ \partial\theta/\partial r$

TT(I,K) t_θ, eq. (9)

TYPE integer (input variable), used as code to indicate how arrays DN, WA, Z, and R are given initially
- 0 - These quantities will be calculated by program
- 1 - These quantities are given as input on binary cards
- 2 - Quantities just computed for previous case will be used for next case (Used only when more than one case is calculated on single computer run)

T1P T/T_i', eq. (3)

W ω (input variable)

WA(I,K) W, eqs. (1) and (13)

WAS W^*, eq. (13)

WASS W^{**}, eq. (13)

WT total weight flow

WTFL(K) calculated total weight flow between hub and K^{th} streamline, eq. (6)

WTHRU W_n, eq. (7)

WTOLER If $|WTFL(KMX) - WT| < WTOLER$ (input variable), then velocity distributions used for computing eq. (6) is accepted as solution to eq. (1)

WTR(I,K) W_t, eq. (10)

XN N (input variable)

XR(J) r-coordinate of TN in thickness table (input variable)

XT(J) z-coordinate of THTA for blade shape (input variable)

XZ(J) z-coordinate of TN in thickness table (input variable)

Z(I,K) z

ZH(I) z-coordinate of hub (input variable)

ZS(I) z-coordinate of shroud (input variable)

Z SPLIT z-coordinate where splitter ends (input variable)

```
      COMMON SRW
      DIMENSION AL(21,21),BETA(21,21),CAL(21,21),CBETA(21,21),
     1CURV(21,21),DN(21,21),PPS(21,21),R(21,21),Z(21,21),SM(21,21),
     2SA(21,21),SB(21,21),SC(21,21),SD(21,21),SAL(21,21),SBETA(21,21),
     3TN(21,21),TT(21,21),WA(21,21),WTR(21,21)
      DIMENSION AB(21),AC(21),AD(21),BA(21),DELBTA(21),DRDM(21),
     1DTDR(21),DTDZ(21),DWMDM(21),DWTDM(21),RH(21),RS(21),ZH(21),ZS(21),
     2THTA(21),WTFL(21),XR(21),XT(21),XZ(21)
      INTEGER RUNO,TYPE,BCDP,SRW
      RUNO=0
   10 READ  (5,1010)MX,KMX,MR,MZ,W,WT,XN,GAM,AR
      ITNO = 1
      RUNO=RUNO+1
      WRITE (6,1020) RUNO
      WRITE (6,1010)MX,KMX,MR,MZ,W,WT,XN,GAM,AR
      READ (5,1010)TYPE,BCDP,SRW,NULL,TEMP,ALM,RHO,TOLER,PLOSS,WTOLER
      WRITE(6,1010)TYPE,BCDP,SRW,NULL,TEMP,ALM,RHO,TOLER,PLOSS,WTOLER
      READ (5,1010)MTHTA,NPRT,ITER,NULL,SFACT,ZSPLIT,BETIN,RB,CORFAC
      WRITE(6,1010)MTHTA,NPRT,ITER,NULL,SFACT,ZSPLIT,BETIN,RB,CORFAC
      READ(5,1030)(ZS(I),I=1,MX)
      WRITE(6,1030)(ZS(I),I=1,MX)
      READ(5,1030)(ZH(I),I=1,MX)
      WRITE(6,1030)(ZH(I),I=1,MX)
      READ(5,1030)(RS(I),I=1,MX)
      WRITE(6,1030)(RS(I),I=1,MX)
      READ(5,1030)(RH(I),I=1,MX)
      WRITE(6,1030)(RH(I),I=1,MX)
      DO 20 I=1,MX
      ZS(I)=ZS(I)/12.
      ZH(I)=ZH(I)/12.
      RS(I)=RS(I)/12.
   20 RH(I)=RH(I)/12.
      IF(TYPE.NE.0) GO TO 40
      WA(1,1) = WT/RHO/(ZS(1)-ZH(1))/3.14/(RS(1)+RH(1))
      DO 30 I=1,MX
      DN(I,KMX)=SQRT((ZS(I)-ZH(I))**2+(RS(I)-RH(I))**2)
      DO 30 K=1,KMX
      DN(I,K)=FLOAT(K-1)/FLOAT(KMX-1)*DN(I,KMX)
      WA(I,K)=WA(1,1)
      Z(I,K)=DN(I,K)/DN(I,KMX)*(ZS(I)-ZH(I))+ZH(I)
   30 R(I,K)=DN(I,K)/DN(I,KMX)*(RS(I)-RH(I))+RH(I)
      GO TO 50
   40 IF(TYPE.NE.1) GO TO 145
      CALL BCREAD(DN(1,1),DN(21,21))
      CALL BCREAD (WA(1,1),WA(21,21))
      CALL BCREAD (Z(1,1),Z(21,21))
      CALL BCREAD (R(1,1),R(21,21))
      WRITE (6,1040)
   50 READ  (5,1030)(THTA(I),I=1,MTHTA)
      WRITE (6,1030)(THTA(I),I=1,MTHTA)
      READ (5,1030)(XT(I),I=1,MTHTA)
```

```
      WRITE(6,1030)(XT(I),I=1,MTHTA)
      DO 60 K=1,MR
      READ  (5,1030)(TN(I,K),I=1,MZ)
   60 WRITE (6,1030)(TN(I,K),I=1,MZ)
      READ  (5,1030)(XZ(I),I=1,MZ)
      WRITE (6,1030)(XZ(I),I=1,MZ)
      READ  (5,1030)(XR(I),I=1,MR)
      WRITE (6,1030)(XR(I),I=1,MR)
C
C  END OF INPUT STATEMENTS
C
C  SCALING-CHANGE INCHES TO FEET AND PSI TO LB/SQ FT, INITIALIZE,
C      CALCULATE CONSTANTS
C
   70 DO 90 K=1,MR
      DO 80 I=1,MZ
   80 TN(I,K) = TN(I,K)/12.
   90 XR(K) = XR(K)/12.
      DO 100 I=1,MZ
  100 XZ(I) = XZ(I)/12.
      DO 110 K=1,KMX
  110 SM(1,K)=0.
      BA(1)=0.
      DO 120 K=2,KMX
  120 BA(K) = FLOAT(K-1)*WT/FLOAT(KMX -1)
      DO 130 I=1,MX
  130 DN(I,1)=0.
      DO 140 I=1,MTHTA
  140 XT(I)=XT(I)/12.
      ROOT = SQRT(2.0)
  145 CONTINUE
      TOLER =TOLER/12.
      RB=RB/12.
      ZSPLIT = ZSPLIT/12.
      PLOSS=PLOSS*144.
      CI = SQRT(GAM*AR*TEMP)
      WRITE (6,1050) CI
      KMXM1 = KMX-1
      CP=AR*GAM/(GAM-1.)
      EXPON = 1./(GAM-1.)
      BETIN = -BETIN/57.29577
      RINLET = (RS(1)+RH(1))/2.
      CEF=SIN(BETIN)/COS(BETIN)/RINLET/(RINLET-RB)**2
      ERROR=100000.
C
C  BEGINNING OF LOOP FOR ITERATIONS
C
  150 IF(ITER.EQ.0) WRITE (6,1060) ITNO
      IF(ITER.EQ.0) WRITE (6,1070)
      ERROR1=ERROR
      ERROR=0.
C
C  START CALCULATION OF PARAMETERS
C
      DO 230 K=1,KMX
```

```
      DO 160 I=1,MX
      AB(I) = (Z(I,K)-R(I,K))/ROOT
  160 AC(I)=(Z(I,K)+R(I,K))/ROOT
      CALL SPLINE (AB,AC,MX,AL(1,K),CURV(1,K))
      DO 170 I=1,MX
      CURV(I,K)=CURV(I,K)/(1.+AL(I,K)**2)**1.5
      AL(I,K) = ATAN(AL(I,K))-.785398
      CAL(I,K) = COS(AL(I,K))
  170 SAL(I,K) = SIN(AL(I,K))
      DO 180 I=2,MX
  180 SM(I,K) = SM(I-1,K)+SQRT((Z(I,K)-Z(I-1,K))**2+(R(I,K)-R(I-1,K))**
     1  2)
  190 CALL SPLDER(XT(1),THTA(1),MTHTA,Z(1,K),MX,DTDZ(1))
      DO 220 I=1,MX
      CALL LININT(Z(I,K),R(I,K),XZ,XR,TN,21,21,T)
      IF(R(I,K).LE.RB)GO TO 200
      DTDR(I)=CEF*(R(I,K)-RB)**2
      GO TO 210
  200 DTDR(I)=0.
  210 TQ=R(I,K)*DTDR(I)
      TP = R(I,K)*DTDZ(I)
      TT(I,K)=T*SQRT(1.+TP*TP)
      BETA(I,K)=ATAN(TP*CAL(I,K)+TQ*SAL(I,K))
      SBETA(I,K) = SIN(BETA(I,K))
      CBETA(I,K) = COS(BETA(I,K))
      SA(I,K)=CBETA(I,K)**2*CAL(I,K)*CURV(I,K)-SBETA(I,K)**2/R(I,K)+
     1SAL(I,K)*CBETA(I,K)*SBETA(I,K)*DTDR(I)
      SC(I,K)=-SAL(I,K)*CBETA(I,K)**2*CURV(I,K)+SAL(I,K)*CBETA(I,K)
     1*SBETA(I,K)*DTDZ(I)
      AB(I)=WA(I,K)*CBETA(I,K)
  220 AC(I)=WA(I,K)*SBETA(I,K)
      CALL SPLINE(SM(1,K),AB,MX,DWMDM,AD)
      CALL SPLINE(SM(1,K),AC,MX,DWTDM,AD)
      IF((ITER.LE.0).AND.(MOD(K-1,NPRT).EQ.0)) WRITE (6,1080) K
      DO 230 I=1,MX
      SB(I,K)=SAL(I,K)*CBETA(I,K)*DWMDM(I)-2.*W*SBETA(I,K)+DTDR(I)*
     1R(I,K)*CBETA(I,K)*(DWTDM(I)+2.*W*SAL(I,K))
      SD(I,K)=CAL(I,K)*CBETA(I,K)*DWMDM(I)+DTDZ(I)*
     1R(I,K)*CBETA(I,K)*(DWTDM(I)+2.*W*SAL(I,K))
      IF((ITER.GT.0).OR.(MOD(K-1,NPRT).NE.0))GO TO 230
      A=  AL(I,K)*57.29577
      B=  SM(I,K)*12.
      E=  TT(I,K)*12.
      G=BETA(I,K)*57.29577
      WRITE (6,1090) A,CURV(I,K),B,G,E,  SA(I,K),SB(I,K),SC(I,K),SD(I,K)
  230 CONTINUE
C
C  END OF LOOP - PARAMETER CALCULATION
C  CALCULATE BLADE SURFACE VELOCITIES (AFTER CONVERGENCE)
C
      IF(ITER.NE.0) GO TO 260
      DO 250 K=1,KMX
      CALL SPLINE (SM(1,K),TT(1,K),MX,DELBTA,AC)
      A=XN
```

```
      DO 240 I=1,MX
  240 AB(I)=(R(I,K)*W+WA(I,K)*SBETA(I,K))*(6.283186*R(I,K)/ A-TT(I,K))
      CALL SPLINE (SM(1,K),AB,MX,DRDM,AC)
      IF (SFACT.LE. 1.0) GO TO 245
      A = SFACT*XN
      DO 244 I=1,MX
  244 AB(I)=(R(I,K)*W+WA(I,K)*SBETA(I,K))*(6.283186*R(I,K)/ A-TT(I,K))
      CALL SPLINE (SM(1,K),AB,MX,AD   ,AC)
  245 DO 250       I=1,MX
      BETAD = BETA(I,K)-DELBTA(I)/2.
      BETAT = BETAD+DELBTA(I)
      COSBD = COS(BETAD)
      COSBT = COS(BETAT)
      IF(Z(I,K).LT.ZSPLIT) DRDM(I) = AD(I)
      WTR(I,K)=COSBD*COSBT/(COSBD+COSBT)*(2.*WA(I,K)/COSBD+R(I,K)*W*
     1(BETAD-BETAT)/CBETA(I,K)**2+DRDM(I))
  250 CONTINUE
C
C      END  OF  BLADE  SURFACE  VELOCITY  CALCULATIONS
C  START CALCULATION OF WEIGHT FLOW VS. DISTANCE FROM HUB
C
  260 DO 370 I=1,MX
      IND=1
      DO 270 K=1,KMX
  270 AC(K)=DN(I,K)
      GO TO 290
  280 WA(I,1)=.5*WA(I,1)
  290 DO 300 K=2,KMX
      J=K-1
      HR=R(I,K)-R(I,J)
      HZ=Z(I,K)-Z(I,J)
      WAS=WA(I,J)*(1.+SA(I,J)*HR+SC(I,J)*HZ)+SB(I,J)*HR+SD(I,J)*HZ
      WASS=WA(I,J)+WAS*(SA(I,K)*HR+SC(I,K)*HZ)+SB(I,K)*HR+SD(I,K)*HZ
  300 WA(I,K)=(WAS+WASS)/2.
  310 DO 340 K=1,KMX
      T1P= 1.-(WA(I,K)**2+2.*W*ALM-(W*R(I,K))**2)/2./CP/TEMP
      IF(T1P.LT..0) GO TO 280
      TPP1P= 1.          (2.*W*ALM-(W*R(I,K))**2)/2./CP/TEMP
      DENSTY=T1P**EXPON*RHO-(T1P/TPP1P)**EXPON*PLOSS/AP/TPP1P/TEMP
     1 *32.17*SM(I,K)/SM(MX,K)
      PRS(I,K)=DENSTY*AR*T1P*TEMP/32.17/144.
      IF(ZS(I).LE.ZH(I)) GO TO 320
      PSI=ATAN((RS(I)-RH(I))/(ZS(I)-ZH(I)))-1.5708
      GO TO 330
  320 PSI=ATAN((ZH(I)-ZS(I))/(RS(I)-RH(I)))
  330 WTHRU=WA(I,K)*CBETA(I,K)*COS(PSI-AL(I,K))
      A=XN
      IF(Z(I,K).LT.ZSPLIT) A=SFACT*XN
      C = 6.283186*R(I,K)-A*TT(I,K)
  340 AD(K)=DENSTY*WTHRU*C
      CALL INTGRL(AC(1),AD(1),KMX,WTFL(1))
      IF (ABS(WT-WTFL(KMX)).LE.WTOLER) GO TO 350
      CALL CONTIN (WA(I,1),WTFL(KMX),IND,I,WT)
      IF (IND.NE.6) GO TO 290
```

```
  350 CALL SPLINT (WTFL,AC,KMX,BA,KMX,AB)
      DO 360 K=1,KMX
      DELTA=ABS(AB(K)-DN(I,K))
      DN(I,K)=(1.-CORFAC)*DN(I,K)+CORFAC*AB(K)
  360 IF(DELTA.GT.ERROR)ERROR=DELTA
  370 CONTINUE
C
C   END OF LOOP -  WEIGHT FLOW CALCULATION
C   CALCULATE STREAMLINE COORDINATES FOR NEXT ITERATION
C
      DO 380 K=2,KMXM1
      DO 380 I=1,MX
      Z(I,K)=DN(I,K)/DN(I,KMX)*(ZS(I)-ZH(I))+ZH(I)
  380 R(I,K)=DN(I,K)/DN(I,KMX)*(RS(I)-RH(I))+RH(I)
      IF((ERROR.GE.ERROR1).OR.(ERROR.LE.TOLER)) ITER=ITER-1
      IF(ITER.GT.0) GO TO 410
      WRITE (6,1100)
      DO 400 K=1,KMX,NPRT
      WRITE (6,1080) K
      DO 390 I=1,MX
      AB(I)=(Z(I,K)-R(I,K))/ROOT
  390 AC(I)=(Z(I,K)+R(I,K))/ROOT
      CALL SPLINE (AB,AC,MX,AL(1,K),CURV(1,K))
      DO 400 I=1,MX
      CURV(I,K)=CURV(I,K)/(1.+AL(I,K)**2)**1.5
      A=DN(I,K)*12.
      B= Z(I,K)*12.
      D= R(I,K)*12.
  400 WRITE (6,1110) A,B,D,WA(I,K),PRS(I,K),WTR(I,K),CURV(I,K)
      WRITE (6,1130)
  410 A=ERROR*12.
      WRITE (6,1120) ITNO,A
      ITNO=ITNO+1
      IF (ITER.GE.0) GO TO 150
      IF(BCDP.NE.1) GO TO 10
      CALL BCDUMP (DN(1,1),DN(21,21))
      CALL BCDUMP (WA(1,1),WA(21,21))
      CALL BCDUMP ( Z(1,1), Z(21,21))
      CALL BCDUMP ( R(1,1), R(21,21))
  420 GO TO 10
 1010 FORMAT (4I5,6F10.4)
 1020 FORMAT (8H1RUN NO.I3,10X,25HINPUT DATA CARD LISTING    )
 1030 FORMAT (7F10.4)
 1040 FORMAT (10X24HBCD CARDS FOR DN,WA,Z,R   )
 1050 FORMAT (36HK     STAG. SPEED OF SOUND AT INLET = ,F9.2)
 1060 FORMAT (///5X13HITERATION NO.I3)
 1070 FORMAT (1H 6X5HAL    9X5HRC    9X5HSM    9X5HBETA 9X5HTT    9X5HSA    9
     1X5HSB    9X5HSC    9X5HSD    )
 1080 FORMAT (2X10HSTREAMLINEI3)
 1090 FORMAT (9F14.6)
 1100 FORMAT (1HL9X5HDN    15X5HZ     15X5HR     15X5HWA    15X5HPRS   14X3HW
     1TR14X3HRC )
 1110 FORMAT (6F19.6,F18.6)
 1120 FORMAT (18H      ITERATION NO. I3,10X,24HMAX. STREAMLINE CHANGE = ,
     1F10.6)
 1130 FORMAT (1HJ)
      END
```

Description of Subroutines

The subroutines SPLINE, SPLINT, SPLDER, and INTGRL are based on the spline fit curve (see appendix C). SPLINE gives the first and second derivatives, SPLINT is used for interpolation, SPLDER is used for interpolated values of the derivative, and INTGRL is used for numerical integration of a function given at unequally spaced points. The calling sequences for these subroutines are as follows:

```
CALL SPLINE (X,Y,N,SLOPE,EM)
```

where

X	input array
Y	input array, function of X
N	input, number of X and Y values given
SLOPE	output array, first derivative, dY/dX
EM	output array, second derivative, d^2Y/dX^2

```
CALL SPLINT (X,Y,N,Z,MAX,YINT)
```

where

X	input array
Y	input array, function of X
N	input, number of X and Y values given
Z	input array, values at which interpolated function values are desired
MAX	input, number of Z values given
YINT	output array, interpolated values

```
CALL SPLDER (X,Y,N,Z,MAX,DYDX)
```

where

X input array

Y input array, function of X

N input, number of X and Y values given

Z input array, values at which the derivative is desired

MAX input, number of Z values given

DYDX output array, derivatives at each Z

CALL INTGRL (X,Y,N,SUM)

where

X input array

Y input array, function of X

N input, number of X and Y values given

SUM(I) output array, $\int_{X(1)}^{X(I)} Y\,DX$

The subroutines SPLINE, SPLINT, SPLDER, and INTGRL are as follows:

```
      SUBROUTINE SPLINE (X,Y,N,SLOPE,EM)
      DIMENSION X(50),Y(50),S(50),A(50),B(50),C(50),F(50),W(50),SB(50),
     1G(50),EM(50),SLOPE(50)
      COMMON Q
      INTEGER Q
      DO 10 I=2,N
   10 S(I)=X(I)-X(I-1)
      NO=N-1
      DO 20 I=2,NO
      A(I)=S(I)/6.
      B(I)=(S(I)+S(I+1))/3.
      C(I)=S(I+1)/6.
   20 F(I)=(Y(I+1)-Y(I))/S(I+1)-(Y(I)-Y(I-1))/S(I)
      A(N)=-.5
      B(1)=1.
      B(N)=1.
      C(1)=-.5
      F(1)=0.
      F(N)=0.
      W(1)=B(1)
      SB(1)=C(1)/W(1)
      G(1)=0.
      DO 30 I=2,N
      W(I)=B(I)-A(I)*SB(I-1)
      SB(I)=C(I)/W(I)
```

```
 30 G(I)=(F(I)-A(I)*G(I-1))/W(I)
    EM(N)=G(N)
    DO 40 I=2,N
    K=N+1-I
 40 EM(K)=G(K)-SB(K)*EM(K+1)
    SLOPE(1)=-S(2)/6.*(2.*EM(1)+EM(2))+(Y(2)-Y(1))/S(2)
    DO50 I=2,N
 50 SLOPE(I)=S(I)/6.*(2.*EM(I)+EM(I-1))+(Y(I)-Y(I-1))/S(I)
    IF (Q.EQ.13) WRITE (6,100) N,(X(I),Y(I),SLOPE(I),EM(I),I=1,21)
100 FORMAT (2X15HNO. OF POINTS =I3/10X5HX     15X5HY     15X5HSLOPE15X5H
   1EM   /(4F20.8))
    RETURN
    END


    SUBROUTINE SPLINT (X,Y,N,Z,MAX,YINT)
    DIMENSION X(50),Y(50),S(50),A(50),B(50),C(50),F(50),W(50),SB(50),
   1G(50),EM(50),Z(50),YINT(50)
    COMMON Q
    INTEGER Q
    DO 10 I=2,N
 10 S(I)=X(I)-X(I-1)
    NO=N-1
    DO 20 I=2,NO
    A(I)=S(I)/6.0
    B(I)=(S(I)+S(I+1))/3.0
    C(I)=S(I+1)/6.0
 20 F(I)=(Y(I+1)-Y(I))/S(I+1)-(Y(I)-Y(I-1))/S(I)
    A(N)=-.5
    B(1)=1.0
    B(N)=1.0
    C(1)=-.5
    F(1)=0.0
    F(N)=0.0
    W(1)=B(1)
    SB(1)=C(1)/W(1)
    G(1)=0.0
    DO 30 I=2,N
    W(I)=B(I)-A(I)*SB(I-1)
    SB(I)=C(I)/W(I)
 30 G(I)=(F(I)-A(I)*G(I-1))/W(I)
    EM(N)=G(N)
    DO 40 I=2,N
    K=N+1-I
 40 EM(K)=G(K)-SB(K)*EM(K+1)
    DO 90 I=1,MAX
    K=2
    IF(Z(I)-X(1)) 60,50,70
 50 YINT(I)=Y(1)
    GO TO 90
 60 IF(Z(I).LT.(1.1*X(1)-.1*X(2)))WRITE (6,1000)Z(I)
    GO TO 85
1000 FORMAT (17H OUT OF RANGE Z =F10.6)
 65 IF(Z(I).GT.(1.1*X(N)-.1*X(N-1))) WRITE (6,1000)Z(I)
    K=N
    GO TO 85
```

```
   70 IF(Z(I)-X(K)) 85,75,80
   75 YINT(I)=Y(K)
      GO TO 90
   80 K=K+1
      IF(K-N) 70,70,65
   85 YINT(I) = EM(K-1)*(X(K)-Z(I))**3/6./S(K)+EM(K)*(Z(I)-X(K-1))**3/6.
     1/S(K)+(Y(K)/S(K)-EM(K)*S(K)/6.)*(Z(I)-X(K-1))+(Y(K-1)/S(K)-EM(K-1)
     2*S(K)/6.)*(X(K)-Z(I))
   90 CONTINUE
      IF(Q.EQ.16) WRITE(6,1010) N,MAX,(X(I),Y(I),Z(I),YINT(I),I=1,N)
 1010 FORMAT (2X21HNO. OF POINTS GIVEN =,I3,30H, NO. OF INTERPOLATED POI
     1NTS =,I3,/10X5HX     15X5HY     12X11HX-INTERPOL.9X11HY-INTERPOL./(4
     2E20.8))
  100 RETURN
      END


      SUBROUTINE SPLDER(X,Y,N,Z,MAX,DYDX)
      DIMENSION X(50),Y(50),S(50),A(50),B(50),C(50),F(50),W(50),SB(50),
     1G(50),EM(50),Z(50),DYDX(50)
      DO 10 I=2,N
   10 S(I)=X(I)-X(I-1)
      NO=N-1
      DO 20 I=2,NO
      A(I)=S(I)/6.0
      B(I)=(S(I)+S(I+1))/3.0
      C(I)=S(I+1)/6.0
   20 F(I)=(Y(I+1)-Y(I))/S(I+1)-(Y(I)-Y(I-1))/S(I)
      A(N)=-.5
      B(1)=1.0
      B(N)=1.0
      C(1)=-.5
      F(1)=0.0
      F(N)=0.0
      W(1)=B(1)
      SB(1)=C(1)/W(1)
      G(1)=0.0
      DO 30 I=2,N
      W(I)=B(I)-A(I)*SB(I-1)
      SB(I)=C(I)/W(I)
   30 G(I)=(F(I)-A(I)*G(I-1))/W(I)
      EM(N)=G(N)
      DO 40 I=2,N
      K=N+1-I
   40 EM(K)=G(K)-SB(K)*EM(K+1)
      DO 90 I=1,MAX
      K=2
      IF(Z(I)-X(1)) 60,70,70
   60 WRITE (6,1000)Z(I)
 1000 FORMAT (17H OUT OF BLADE Z =F10.6)
      GO TO 85
   65 WRITE (6,1000)Z(I)
      K=N
      GO TO 85
   70 IF(Z(I)-X(K)) 85,85,80
   80 K=K+1
```

```
      IF(K-N) 70,70,65
   85 DYDX(I)=-EM(K-1)*(X(K)-Z(I))**2/2.0/S(K)+EM(K)*(X(K-1)-Z(I))**2/2.
     10/S(K)+(Y(K)-Y(K-1))/S(K)-(EM(K)-EM(K-1))*S(K)/6.
   90 CONTINUE
  100 RETURN
      END


      SUBROUTINE INTGRL (X,Y,N,SUM)
      DIMENSION X(50),Y(50),S(50),A(50),B(50),C(50),F(50),W(50),SB(50),
     1G(50),EM(50),SUM(50)
C     DIMENSION X(50),Y(50),S(50),A(50),B(50),C(50),F(50),W(50),SB(50),
C    1G(50),EM(50),SUM(50)
      DO 10 I=2,N
   10 S(I)=X(I)-X(I-1)
      NO=N-1
      DO 20 I=2,NO
      A(I)=S(I)/6.0
      B(I)=(S(I)+S(I+1))/3.0
      C(I)=S(I+1)/6.0
   20 F(I)=(Y(I+1)-Y(I))/S(I+1)-(Y(I)-Y(I-1))/S(I)
      A(N)=-.5
      B(1)=1.0
      B(N)=1.0
      C(1)=-.5
      F(1)=0.0
      F(N)=0.0
      W(1)=B(1)
      SB(1)=C(1)/W(1)
      G(1)=0.0
      DO 30 I=2,N
      W(I)=B(I)-A(I)*SB(I-1)
      SB(I)=C(I)/W(I)
   30 G(I)=(F(I)-A(I)*G(I-1))/W(I)
      EM(N)=G(N)
      DO 40 I=2,N
      K=N+1-I
   40 EM(K)=G(K)-SB(K)*EM(K+1)
      SUM(1) =0.0
      DO 50 K=2,N
   50 SUM(K) = SUM(K-1)+S(K)*(Y(K)+Y(K-1))/2.0-S(K)**3*(EM(K)+EM(K-1))/2
     14.0
      RETURN
      END
```

The subroutine LININT performs linear interpolation of a function of two variables. It is used here to obtain interpolated values of normal blade thickness t_n from a table of thickness values given as input. The calling sequence for LININT is as follows:

CALL LININT (X1,Y1,X,Y,TN,MX,MY,F)

where

X1 input, x-coordinate of point for which interpolated function value is desired

Y1 input, y-coordinate of point for which interpolated function value is desired

X input array, x-coordinates at which function values are specified

Y input array, y-coordinates at which function values are specified

TN input two-dimensional array, function of x and y, first subscript refers to x-coordinate

MX input, number of x values given

MY input, number of y values given

F output, interpolated value

The subroutine LININT is as follows:

```
      SUBROUTINE LININT(X1,Y1,X,Y,TN,MX,MY,F)
      COMMON K
      DIMENSION X(MX),Y(MY),TN(MX,MY)
      DO 10 J3=1,MX
   10 IF(X1.LE.X(J3))GO TO 20
      J3=MX
   20 DO 30 J4=1,MY
   30 IF(Y1.LE.Y(J4))GO TO 40
      J4=MY
   40 J1=J3-1
      J2=J4-1
      EPS1=(X1-X(J1))/(X(J3)-X(J1))
      EPS2=(Y1-Y(J2))/(Y(J4)-Y(J2))
      EPS3=1.-EPS1
      EPS4=1.-EPS2
      F=TN(J1,J2)*EPS3*EPS4+TN(J3,J2)*EPS1*EPS4+TN(J1,J4)*EPS2*EPS3+
     1TN(J3,J4)*EPS1*EPS2
      IF(K.EQ.14) WRITE(6,1)X1,Y1,F,J1,J2,EPS1,EPS2
    1 FORMAT (8H  LININT3F10.5,2I3,2F10.5)
      K=0
      RETURN
      END
```

The subroutine CONTIN is used to predict the hub velocity to be used in the next iteration to satisfy continuity of flow (eq. (6)) between hub and shroud. An initial estimate is furnished by the main program, say W_1 (see fig. 12). CONTIN furnishes the next estimate W_2 by linear interpolation or extrapolation from the origin. Subsequent estimates are obtained by linear interpolation from the two previous estimates.

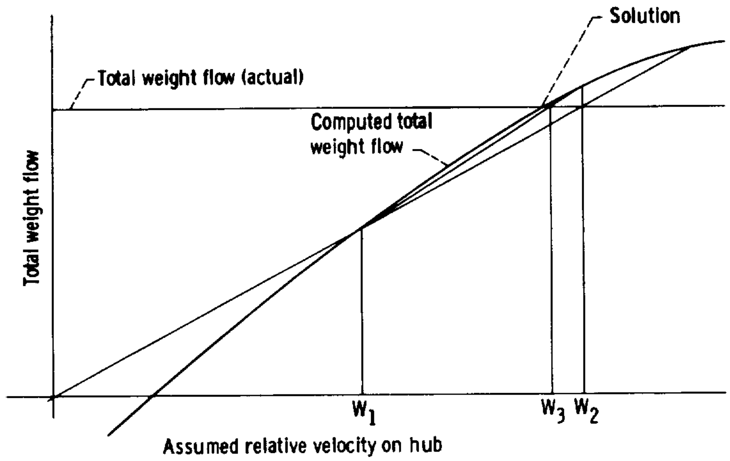

Figure 12. - Method used by subroutine CONTIN to determine relative hub velocity.

If there is choked flow, there is no solution of equation (1) that will also satisfy continuity (eq. (6)). In this case, CONTIN will find the hub velocity that gives the maximum calculated weight flow. It should be noted that CONTIN does not calculate the weight flow; this is calculated by the main program. CONTIN stores information from up to three previous iterations to assist in predicting the next value to be used for the hub velocity. The calling sequence for CONTIN is as follows:

CALL CONTIN (WA,WTFL,IND,I,WT)

where

WA input and output; as input, hub relative velocity used to calculate latest weight flow and as output, velocity used for next iteration

WTFL input, calculated weight flow based on input value of WA

IND input and output; main program sets IND = 1 to indicate start of weight-flow calculation for new quasi-orthogonal and CONTIN changes value of IND for following iterations to indicate procedure followed in calculating new hub velocity

I input, number of quasi-orthogonal used by subroutine CONTIN in WRITE statement if there is choked flow

WT input, total weight flow

The subroutine CONTIN is as follows:

```
    SUBROUTINE CONTIN (WA,WTFL,IND,I,WT)
    DIMENSION SPEED(3),WEIGHT(3)
135 GO TO (140,150,210,270,370),IND
140 SPEED(1) = WA
    WEIGHT(1) = WTFL
    WA = WT/WTFL*WA
    IND = 2
    RETURN
150 IF ((WTFL-WEIGHT(1))/(WA-SPEED(1))) 180,180,160
160 SPEED(2) = WA
    WA = (WT-WTFL)/(WTFL-WEIGHT(1))
   1   *(WA-SPEED(1))+WA
```

```
    IF (ABS(WA-SPEED(2))-100.0) 166,166,161
161 IF(WA-SPEED(2))163,163,162
162 WA = SPEED(2)+100.0
    GO TO 166
163 WA = SPEED(2)-100.0
166 SPEED(1) = SPEED(2)
    WEIGHT(1) = WTFL
    RETURN
170 WRITE (6,1000) I,WTFL
    IND = 6
    RETURN
180 IND = 3
    IF (WTFL.GE.WT) GO TO 14
    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 14
    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)-10.0) 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)-10.0) 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.
      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
 1000 FORMAT (/12H FIXED LINE I2,12H,  MAX WT = F10.6)
      END
```

Sample Output from Program

The output given here is the listing for the case used in the numerical example. It will be noted that there is an exact listing of all input data cards at the beginning of the listing. This is followed by the maximum calculated streamline change for each iteration, which is used as the criterion for convergence. After 47 iterations, there is convergence within the specified limit of 0.001-inch maximum streamline change. At this time, streamline coordinates are printed together with the velocity and pressure at each point. This is followed by another iteration to give additional information of interest, such as α, β, and the parameters A, B, C, and D from equation (2). Since it indicates the smoothness of the streamline at a glance, the streamline curvature is also printed out. The velocities and the pressures are computed again on the final iteration so that the variation of these quantities on the final iteration can be checked.

RUN NO. 1 INPUT DATA CARD LISTING

```
  10    11    17    13 5390.0000    0.9840   13.0000    1.4000 1715.0000
   0     0    -0     0  592.0000  155.3000    0.1941    0.0010    2.5000
  13     2     2     1    1.0000   -1.0000  -35.0000    1.7500    0.1000
   0.3000    0.3400    0.3948    0.4810    0.5412    0.6193    0.7080
   0.8130    0.9100    1.0000
   0.       -0.0270   -0.0530   -0.0540    0.0090    0.1370    0.4100
   0.7300    0.9100    1.0400
   2.2500    2.0520    1.8610    1.6800    1.6000    1.5341    1.4960
   1.4785    1.4751    1.4750
   2.2500    2.0180    1.7630    1.4120    1.2080    1.0010    0.7790
   0.6800    0.6750    0.6750
   0.        0.        0.       -0.0004   -0.0027   -0.0090   -0.0240
  -0.0517   -0.0972   -0.1632   -0.2487   -0.3512   -0.4660
  -0.1000    0.        0.1000    0.2000    0.3000    0.4000    0.5000
   0.6000    0.7000    0.8000    0.9000    1.0000    1.1000
   0.3850    0.3580    0.3220    0.2900    0.2600    0.2300    0.2010
   0.1750    0.1500    0.1280    0.1080    0.0920    0.0790
   0.3750    0.3450    0.3110    0.2800    0.2500    0.2200    0.1910
   0.1650    0.1400    0.1190    0.1000    0.0840    0.0740
   0.3650    0.3330    0.3000    0.2700    0.2400    0.2100    0.1810
   0.1550    0.1310    0.1100    0.0920    0.0780    0.0690
   0.3550    0.3210    0.2900    0.2590    0.2290    0.2000    0.1710
   0.1460    0.1210    0.1010    0.0850    0.0730    0.0640
   0.3450    0.3100    0.2800    0.2490    0.2190    0.1890    0.1610
   0.1360    0.1130    0.0930    0.0790    0.0690    0.0590
   0.3330    0.3000    0.2690    0.2380    0.2080    0.1790    0.1500
   0.1270    0.1040    0.0860    0.0730    0.0630    0.0560
   0.3200    0.2900    0.2580    0.2270    0.1970    0.1680    0.1400
   0.1170    0.0950    0.0780    0.0680    0.0590    0.0520
   0.3100    0.2790    0.2470    0.2150    0.1860    0.1570    0.1300
   0.1070    0.0870    0.0730    0.0620    0.0550    0.0490
   0.3000    0.2680    0.2360    0.2040    0.1750    0.1470    0.1200
   0.0980    0.0790    0.0680    0.0580    0.0510    0.0460
   0.2800    0.2570    0.2240    0.1930    0.1640    0.1360    0.1100
   0.0880    0.0730    0.0620    0.0540    0.0470    0.0430
   0.2510    0.2460    0.2130    0.1820    0.1530    0.1240    0.1000
   0.0790    0.0670    0.0570    0.0500    0.0450    0.0400
   0.2230    0.2230    0.2020    0.1700    0.1410    0.1130    0.0890
   0.0700    0.0610    0.0520    0.0460    0.0430    0.0370
   0.1940    0.1940    0.1900    0.1590    0.1300    0.1020    0.0790
   0.0600   -0.       -0.       -0.       -0.       -0.
   0.1660    0.1660    0.1660    0.1480    0.1180    0.0910    0.0690
   0.0500   -0.       -0.       -0.       -0.       -0.
   0.1370    0.1370    0.1370    0.1370    0.1060    0.0800    0.0590
   0.0400   -0.       -0.       -0.       -0.       -0.
   0.1090    0.1090    0.1090    0.1090    0.0930    0.0700    0.0500
   0.0300   -0.       -0.       -0.       -0.       -0.
   0.0800    0.0800    0.0800    0.0800    0.0800    0.0600    0.0400
   0.0200   -0.       -0.       -0.       -0.       -0.
  -0.1000    0.        0.1000    0.2000    0.3000    0.4000    0.5000
   0.6000    0.7000    0.8000    0.9000    1.0000    1.1000
   0.6500    0.7500    0.8500    0.9500    1.0500    1.1500    1.2500
   1.3500    1.4500    1.5500    1.6500    1.7500    1.8500    1.9500
   2.0500    2.1500    2.2500
```

```
STAG. SPEED OF SOUND AT INLET =   1192.22
ITERATION NO.  1          MAX. STREAMLINE CHANGE =   0.222103
ITERATION NO.  2          MAX. STREAMLINE CHANGE =   0.208163
ITERATION NO.  3          MAX. STREAMLINE CHANGE =   0.184211
ITERATION NO.  4          MAX. STREAMLINE CHANGE =   0.161841
ITERATION NO.  5          MAX. STREAMLINE CHANGE =   0.141914
ITERATION NO.  6          MAX. STREAMLINE CHANGE =   0.124332
ITERATION NO.  7          MAX. STREAMLINE CHANGE =   0.108987
ITERATION NO.  8          MAX. STREAMLINE CHANGE =   0.095636
ITERATION NO.  9          MAX. STREAMLINE CHANGE =   0.084035
ITERATION NO. 10          MAX. STREAMLINE CHANGE =   0.073942
ITERATION NO. 11          MAX. STREAMLINE CHANGE =   0.065150
ITERATION NO. 12          MAX. STREAMLINE CHANGE =   0.057468
ITERATION NO. 13          MAX. STREAMLINE CHANGE =   0.050736
ITERATION NO. 14          MAX. STREAMLINE CHANGE =   0.044821
ITERATION NO. 15          MAX. STREAMLINE CHANGE =   0.039616
ITERATION NO. 16          MAX. STREAMLINE CHANGE =   0.035032
ITERATION NO. 17          MAX. STREAMLINE CHANGE =   0.030988
ITERATION NO. 18          MAX. STREAMLINE CHANGE =   0.027419
ITERATION NO. 19          MAX. STREAMLINE CHANGE =   0.024267
ITERATION NO. 20          MAX. STREAMLINE CHANGE =   0.021483
ITERATION NO. 21          MAX. STREAMLINE CHANGE =   0.019023
ITERATION NO. 22          MAX. STREAMLINE CHANGE =   0.016847
ITERATION NO. 23          MAX. STREAMLINE CHANGE =   0.014923
ITERATION NO. 24          MAX. STREAMLINE CHANGE =   0.013248
ITERATION NO. 25          MAX. STREAMLINE CHANGE =   0.011771
ITERATION NO. 26          MAX. STREAMLINE CHANGE =   0.010461
ITERATION NO. 27          MAX. STREAMLINE CHANGE =   0.009301
ITERATION NO. 28          MAX. STREAMLINE CHANGE =   0.008270
ITERATION NO. 29          MAX. STREAMLINE CHANGE =   0.007356
ITERATION NO. 30          MAX. STREAMLINE CHANGE =   0.006546
ITERATION NO. 31          MAX. STREAMLINE CHANGE =   0.005824
ITERATION NO. 32          MAX. STREAMLINE CHANGE =   0.005219
ITERATION NO. 33          MAX. STREAMLINE CHANGE =   0.004610
ITERATION NO. 34          MAX. STREAMLINE CHANGE =   0.004130
ITERATION NO. 35          MAX. STREAMLINE CHANGE =   0.003672
ITERATION NO. 36          MAX. STREAMLINE CHANGE =   0.003252
ITERATION NO. 37          MAX. STREAMLINE CHANGE =   0.002925
ITERATION NO. 38          MAX. STREAMLINE CHANGE =   0.002615
ITERATION NO. 39          MAX. STREAMLINE CHANGE =   0.002326
ITERATION NO. 40          MAX. STREAMLINE CHANGE =   0.002068
ITERATION NO. 41          MAX. STREAMLINE CHANGE =   0.001845
ITERATION NO. 42          MAX. STREAMLINE CHANGE =   0.001645
ITERATION NO. 43          MAX. STREAMLINE CHANGE =   0.001487
ITERATION NO. 44          MAX. STREAMLINE CHANGE =   0.001315
ITERATION NO. 45          MAX. STREAMLINE CHANGE =   0.001187
ITERATION NO. 46          MAX. STREAMLINE CHANGE =   0.001047
ITERATION NO. 47          MAX. STREAMLINE CHANGE -   0.000947
```

DN	Z	R	WA	PRS	WTR	RC
STREAMLINE 1						
0.	0.	2.250000	295.700397	28.947868	-0.000000	0.204638
0.	-0.027000	2.018000	244.298780	26.055681	-0.000000	0.418743
0.	-0.053000	1.763000	224.834787	23.130037	-0.000000	2.373524
0.	-0.054000	1.412000	176.895653	20.036808	-0.000000	17.658230
0.	0.009000	1.208000	183.239071	18.452905	-0.000000	10.869264
0.	0.137000	1.001000	185.988209	17.094800	-0.000000	14.145438
0.	0.410000	0.779000	193.639532	15.786651	-0.000000	11.166778
0.	0.730000	0.680000	248.210934	14.877375	-0.000000	14.006964
0.	0.910000	0.675000	398.924618	13.777308	-0.000000	-0.110846
0.	1.040000	0.675000	536.335518	12.539055	-0.000000	-0.055573
STREAMLINE 3						
0.060672	0.060684	2.250000	296.588036	28.939592	-0.000000	1.553005
0.080339	0.053007	2.025412	248.140682	26.072129	-0.000000	3.607813
0.118810	0.063075	1.788403	241.029270	23.195781	-0.000000	9.014218
0.209884	0.133647	1.505999	236.847181	20.346146	-0.000000	16.133527
0.258480	0.217108	1.361285	248.612078	19.011727	-0.000000	15.957128
0.305917	0.342227	1.227843	270.067642	17.808045	-0.000000	13.116173
0.340800	0.540788	1.093681	309.053890	16.529749	-0.000000	8.133155
0.315836	0.762652	0.994124	409.698055	15.039541	-0.000000	5.959627
0.271259	0.910000	0.946240	509.067570	13.808785	-0.000000	6.614564
0.242013	1.027915	0.916692	596.051422	12.722304	-0.000000	3.035073
STREAMLINE 5						
0.120930	0.120953	2.250000	298.582912	28.920617	-0.000000	2.357783
0.157646	0.129995	2.032544	256.080162	26.068767	-0.000000	5.717441
0.221072	0.162982	1.810267	262.185879	23.215000	-0.000000	10.279274
0.344840	0.254305	1.566441	285.821499	20.448035	-0.000000	15.918233
0.406109	0.335968	1.448833	310.106243	19.164313	-0.000000	16.069598
0.460545	0.445961	1.342503	342.531384	17.975591	-0.000000	13.803547
0.500214	0.601966	1.240878	399.700100	16.637705	-0.000000	9.957688
0.484882	0.780128	1.162254	506.095100	14.986505	-0.000000	7.680776
0.446389	0.910000	1.121359	593.856674	13.718478	-0.000000	6.951878
0.420479	1.019004	1.094920	654.802147	12.799590	-0.000000	3.193338
STREAMLINE 7						
0.180798	0.180832	2.250000	301.208126	28.895584	-0.000000	2.524349
0.231773	0.203816	2.039383	266.384010	26.044178	-0.000000	6.006097
0.309998	0.249861	1.829280	285.503525	23.183644	-0.000000	10.570901
0.447230	0.345848	1.612298	335.243137	20.390576	-0.000000	15.842568
0.511895	0.421139	1.511567	375.243790	19.052803	-0.000000	17.236639
0.567952	0.518016	1.422148	425.197060	17.756366	-0.000000	16.849025
0.612053	0.644886	1.344144	496.102688	16.334171	-0.000000	14.242857
0.610084	0.793071	1.286777	594.406387	14.713892	-0.000000	11.129697
0.583089	0.910000	1.258050	666.576347	13.561663	-0.000000	6.827234
0.565662	1.011754	1.239910	710.973190	12.805255	-0.000000	3.126906
STREAMLINE 9						
0.240425	0.240470	2.250000	304.193565	28.866869	-0.000000	2.337487
0.302140	0.273893	2.045876	277.541824	26.006223	-0.000000	5.367771
0.388486	0.326542	1.846062	311.534645	23.096389	-0.000000	10.364191
0.529675	0.419558	1.649222	389.888870	20.164071	-0.000000	16.181698
0.594161	0.487373	1.560353	452.635353	18.635005	-0.000000	20.016058
0.650805	0.573599	1.483585	528.748474	17.063663	-0.000000	23.717952
0.700598	0.678866	1.425903	604.926697	15.606427	-0.000000	21.306215
0.712931	0.803704	1.389066	680.438080	14.258159	-0.000000	14.370825
0.698300	0.910000	1.373253	728.139481	13.408373	-0.000000	5.774815
0.690357	1.005528	1.364440	761.253441	12.801967	-0.000000	2.659021
STREAMLINE 11						
0.299943	0.300000	2.250000	307.510876	28.834654	-0.000000	2.001920
0.368522	0.340000	2.052000	289.256516	25.956492	-0.000000	4.448114
0.458352	0.394800	1.861000	340.851379	22.946434	-0.000000	9.891359
0.598398	0.481000	1.680000	453.113907	19.734822	-0.000000	16.634179
0.661016	0.541200	1.600000	551.139595	17.782610	-0.000000	25.384029
0.718929	0.619300	1.534100	670.391991	15.612218	-0.000000	38.460735
0.776513	0.708000	1.496000	727.136063	14.446990	-0.000000	29.737586
0.802852	0.813000	1.478500	759.647667	13.735610	-0.000000	15.735754
0.800154	0.910000	1.475100	777.727600	13.326141	-0.000000	2.622090
0.801064	1.000000	1.475000	804.465561	12.829419	-0.000000	1.255736

ITERATION NO. 48   MAX. STREAMLINE CHANGE = 0.000833

ITERATION NO. 49

AL	RC	SM	BETA	TT	SA	SB	SC	SD
STREAMLINE 1								
-96.791329	0.204638	0.	-34.811041	0.080000	-0.016337	4749.847290	0.137180	0.718269
-96.445220	0.418748	0.233566	-10.164403	0.146280	-0.045545	1000.156158	0.403172	28.158364
-94.803148	2.373624	0.489888	-0.021144	0.219230	-0.198750	1496.086212	2.365288	125.762896
-81.547207	17.658230	0.840889	-0.000052	0.289255	2.595658	453.016899	17.466414	-67.254692
-65.289629	10.869264	1.054396	0.001206	0.291358	4.543693	-422.685627	9.873993	193.911631
-50.600565	14.145438	1.297774	-0.060098	0.273430	8.978408	-110.496499	10.930716	114.005714
-28.514803	11.166778	1.649645	-4.243374	0.214962	9.674118	858.094780	5.255855	406.167065
-5.167763	14.006964	1.984609	-22.761247	0.152448	9.220404	3765.058563	0.833892	7251.293640
0.028702	-0.110846	2.164678	-33.029459	0.124524	-5.359719	5879.310730	0.002685	12094.271240
-0.022972	-0.055573	2.294678	-37.589165	0.107395	-6.649906	6573.061096	-0.002666	13172.711060
STREAMLINE 3								
-93.118897	1.553005	0.	-34.960107	0.080000	-0.056754	4696.964172	1.041558	-3.414272
-90.431669	3.607813	0.224719	-10.826351	0.144130	-0.026223	852.774353	3.480459	-0.363412
-83.699650	9.014218	0.461942	-0.187020	0.202735	0.989211	310.238937	8.959679	-33.906509
-66.569610	16.133527	0.753030	-0.047930	0.218702	6.415236	-340.324947	14.803196	172.663132
-53.506522	15.957128	0.920087	-0.667775	0.208837	9.487714	-850.368507	12.824967	897.430389
-40.602605	13.116173	1.103011	-3.000268	0.187637	9.904289	-359.318771	8.489786	1658.862259
-28.451371	8.133155	1.342648	-14.100324	0.152210	6.075231	1795.320847	3.291908	4385.999512
-20.305481	5.959627	1.585826	-32.619006	0.127045	0.457653	5049.995544	0.169341	9174.457651
-15.618416	6.614564	1.740759	-41.274627	0.113728	-1.920442	6438.878540	-0.536863	10459.242920
-12.862891	3.035073	1.862320	-45.229616	0.103789	-5.130140	7033.912720	-1.171463	10601.291138
STREAMLINE 5								
-89.348869	2.357783	0.	-34.998891	0.080000	0.017938	4512.893311	1.578415	5.493444
-85.312701	5.717441	0.217643	-11.392505	0.141120	0.448767	470.546402	5.473328	51.081024
-76.994141	10.279274	0.442355	-0.559493	0.174999	2.313019	-676.827492	10.014124	239.159569
-60.750076	15.918233	0.702723	-1.037799	0.175564	7.772901	-1268.069473	13.879563	1187.986649
-49.682521	16.069598	0.845902	-2.831104	0.165534	10.351817	-1158.549133	12.198901	2181.394012
-38.657282	13.803547	0.998887	-8.607185	0.148087	10.337504	212.493286	8.269257	3839.445374
-28.080564	9.957688	1.185074	-21.637005	0.128805	6.276276	3150.224579	3.348484	7229.672546
-19.897895	7.680776	1.379813	-38.255044	0.115591	0.495365	6314.752258	0.179299	10573.176270
-15.137877	6.951878	1.515971	-46.194237	0.108507	-2.358145	7587.568726	-0.637949	9534.872925
-12.464280	3.193338	1.628136	-50.087487	0.102216	-5.164270	8144.871399	-1.141514	7883.192678
STREAMLINE 7								
-85.565472	2.524349	0.	-34.966721	0.080010	0.127992	4214.860901	1.650407	70.493948
-81.396382	6.006097	0.211867	-12.062828	0.137026	0.855045	31.712341	5.651320	285.175068
-73.103230	10.570901	0.426956	-1.468962	0.146943	3.068384	-1607.140839	10.101295	915.319283
-58.158588	15.842568	0.664222	-2.840673	0.144660	8.319252	-2246.390564	13.395957	2755.937531
-48.148163	17.236639	0.789981	-6.849531	0.136815	11.223905	-1508.543655	12.530424	4621.057495
-37.308163	16.849025	0.921818	-14.538452	0.124925	12.025264	908.280350	9.163506	6894.059631
-26.173255	14.242857	1.070749	-28.154876	0.114749	7.948738	4714.625549	3.906657	9630.716675
-16.549163	11.129697	1.229652	-42.660808	0.107174	1.486904	7437.082825	0.441829	10351.455078
-11.467521	6.827234	1.350057	-49.899707	0.104685	-2.804941	8358.256592	-0.569016	8009.629761
-9.051965	3.126906	1.453416	-53.652507	0.099651	-5.193770	8772.619629	-0.827442	5755.924500
STREAMLINE 9								
-82.305455	2.337487	0.	-34.989594	0.080081	0.194941	3882.623199	1.442850	197.201910
-78.600956	5.367771	0.206843	-12.972187	0.114782	0.992282	-463.984436	4.921598	664.918777
-71.058163	10.364191	0.413476	-2.793755	0.123482	3.347211	-2627.989960	9.753165	1850.621643
-57.335515	16.181688	0.631187	-5.973829	0.121620	8.560147	-3490.297974	13.352002	5396.945251
-47.626312	20.016058	0.742975	-11.351406	0.117094	12.669536	-2414.196045	13.887719	8602.198975
-35.385800	23.717952	0.858423	-20.716563	0.110732	15.904707	2018.682144	11.296958	10969.803955
-22.290471	21.306215	0.978458	-34.042324	0.105239	10.898697	6523.880310	4.467765	10777.033813
-11.311732	14.370825	1.108617	-46.286274	0.103303	2.216258	8267.744263	0.443324	7954.541565
-6.340032	5.774815	1.216083	-52.739881	0.100102	-3.431499	8768.545776	-0.381268	5606.352856
-4.438718	2.659021	1.312016	-56.304519	0.097717	-5.272015	9050.917114	-0.409243	4460.529236
STREAMLINE 11								
-79.907404	2.001920	0.	-35.148235	0.080281	0.197647	3645.781311	1.110417	356.526695
-76.858636	4.448114	0.202000	-14.171409	0.095969	0.915087	-1087.359116	3.919538	1201.083969
-70.267831	9.891359	0.400706	-4.927414	0.103808	3.292170	-3713.461395	9.150561	3307.145905
-57.715281	16.634179	0.601184	-9.344786	0.105957	8.462180	-5658.535156	13.393758	9212.995605
-47.657956	25.384029	0.701304	-15.757485	0.104238	15.283514	-4647.681213	16.771626	15780.298706
-31.560049	38.460735	0.803492	-27.278743	0.101191	24.244970	4232.239014	14.892315	15347.742065
-15.618170	29.737586	0.900029	-39.560645	0.099287	13.768619	8393.373657	3.848975	7430.751709
-4.485098	15.735754	1.006477	-49.197497	0.099745	2.047968	8514.308960	0.160643	3156.978271
-0.525712	2.622090	1.103537	-54.859308	0.097809	-4.571248	8839.857422	-0.041944	2772.932007
0.300710	1.255736	1.193537	-58.251249	0.095023	-5.535313	9161.080933	0.029052	4187.106445

DN	Z	R	WA	PRS	WTR	RC
STREAMLINE 1						
0.	0.	2.250000	295.700397	28.947868	-22.104850	0.204638
0.	-0.027000	2.018000	244.298780	26.055681	-232.784702	0.418748
0.	-0.053000	1.763000	224.834787	23.130037	-306.823132	2.373624
0.	-0.054000	1.412000	176.895653	20.036808	-117.254062	17.658230
0.	0.009000	1.208000	183.239071	18.452905	25.664505	10.869264
0.	0.137000	1.001000	185.988209	17.094800	115.742643	14.145438
0.	0.410000	0.779000	193.639532	15.786651	197.585930	11.166778
0.	0.730000	0.680000	248.210934	14.877375	232.723198	14.006964
0.	0.910000	0.675000	398.967968	13.776997	344.500111	-0.110846
0.	1.040000	0.675000	536.335518	12.539055	457.657112	-0.055573
STREAMLINE 3						
0.060673	0.060685	2.250000	296.588394	28.939498	-23.730222	1.553489
0.080350	0.053018	2.025413	248.142447	26.072117	-221.816166	3.609064
0.118836	0.063101	1.788408	241.034775	23.195787	-217.088028	9.015433
0.209936	0.133693	1.506022	236.857012	20.346262	-3.141071	16.132396
0.258541	0.217157	1.361321	248.633646	19.011854	97.262347	15.957800
0.305987	0.342273	1.227894	270.089710	17.808236	177.749821	13.114767
0.340872	0.540815	1.093746	309.091137	16.529879	200.866070	8.131182
0.315897	0.762658	0.994183	409.738766	15.039520	283.336746	5.956387
0.271305	0.910000	0.946286	509.126015	13.808458	401.696693	6.617541
0.242051	1.027914	0.916728	596.058434	12.722378	499.906059	3.036371
STREAMLINE 5						
0.120931	0.120955	2.250000	298.583912	28.920606	-53.266858	2.358270
0.157662	0.130011	2.032546	256.085121	26.068731	-156.106657	5.718700
0.221107	0.163017	1.810275	262.196823	23.214976	-119.569639	10.280068
0.344901	0.254358	1.566467	285.835472	20.448139	45.190903	15.916787
0.406178	0.336023	1.448874	310.141842	19.164348	127.893261	16.069123
0.460620	0.446010	1.342557	342.563675	17.975713	170.073149	13.802033
0.500290	0.601994	1.240946	399.754654	16.637682	190.375256	9.957924
0.484956	0.780135	1.162325	506.141823	14.986462	318.726822	7.677694
0.446455	0.910000	1.121423	593.917297	13.718166	464.800022	6.950983
0.420536	1.019001	1.094975	654.813416	12.799726	561.800369	3.193009
STREAMLINE 7						
0.180799	0.180834	2.250000	301.209805	28.895567	-83.040575	2.524727
0.231789	0.203832	2.039385	266.391666	26.044117	-74.177796	6.007010
0.310030	0.249894	1.829288	285.517944	23.183576	-33.488862	10.571429
0.447285	0.345894	1.612321	335.257111	20.390633	48.717430	15.840850
0.511954	0.421185	1.511601	375.287552	19.052671	122.137831	17.236588
0.568016	0.518057	1.422193	425.234100	17.756336	143.309679	16.848888
0.612118	0.644910	1.344203	496.161919	16.333957	196.058693	14.243553
0.610151	0.793078	1.286840	594.444382	14.713859	374.366112	11.126811
0.583151	0.910000	1.258109	666.623207	13.561447	532.477699	6.825364
0.565718	1.011752	1.239963	710.979408	12.805460	626.039001	3.126163
STREAMLINE 9						
0.240425	0.240472	2.250000	304.195923	28.866847	31.624759	2.337692
0.302150	0.273903	2.045877	277.550880	26.006145	-19.894819	5.368242
0.388505	0.326563	1.846066	311.548267	23.096297	-21.628932	10.364577
0.529709	0.419586	1.649236	389.894302	20.164118	40.750961	16.180407
0.594196	0.487401	1.560373	452.670174	18.634791	82.249205	20.016625
0.650845	0.573623	1.483612	528.767303	17.063639	107.006126	23.718976
0.700638	0.678881	1.425939	604.958633	15.606282	232.651516	21.306772
0.712973	0.803708	1.389104	680.447067	14.258292	466.458229	14.368608
0.698338	0.910000	1.373289	728.165421	13.408259	607.934540	5.774084
0.690394	1.005526	1.364472	761.247803	12.802229	680.406281	2.658730
STREAMLINE 11						
0.299941	0.300000	2.250000	307.513741	28.834627	138.704298	2.001920
0.368521	0.340000	2.052000	289.264946	25.956421	26.413971	4.448114
0.458350	0.394800	1.861000	340.856804	22.946384	-24.200536	9.891359
0.598401	0.481000	1.680000	453.091351	19.735065	18.565243	16.634179
0.661017	0.541200	1.600000	551.122826	17.782814	-13.943725	25.384029
0.718932	0.619300	1.534100	670.326637	15.613094	78.383554	38.460735
0.776515	0.708000	1.496000	727.097916	14.447514	355.786674	29.737586
0.802856	0.813000	1.478500	759.615166	13.736058	596.755402	15.735754
0.800157	0.910000	1.475100	777.732147	13.326078	681.073318	2.622090
0.801069	1.000000	1.475000	804.445007	12.829704	718.886520	1.255736

ITERATION NO. 49 MAX. STREAMLINE CHANGE = 0.000750

REFERENCES

1. Hamrick, Joseph T., Ginsburg, Ambrose, and Osborn, Walter M.: Method of Analysis for Compressible Flow Through Mixed-Flow Centrifugal Impellers of Arbitrary Design. NACA Rep. 1082, 1952.

2. Smith, Kenneth J., and Hamrick, Joseph T.: A Rapid Approximate Method for the Design of Hub Shroud Profiles of Centrifugal Compressors of Given Blade Shape. NACA TN 3399, 1955.

3. Osborn, Walter M., Smith, Kenneth J., and Hamrick, Joseph T.: Design and Test of Mixed-flow Impellers. VIII - Comparison of Experimental Results for Three Impellers with Shroud Redesigned by Rapid Approximate Method. NACA RM E56L07, 1957.

4. Stockman, Norbert O., and Kramer, John L.: Method for Design of Pump Impellers Using a High-Speed Digital Computer. NASA TN D-1562, 1963.

5. Ellis, Gaylord O., and Stanitz, John D.: Comparison of Two- and Three-Dimensional Potential-Flow Solutions in a Rotating Impeller Passage. NACA TN 2806, 1952.

6. Stanitz, John D., and Prian, Vasily D.: A Rapid Approximate Method for Determining Velocity Distribution on Impeller Blades of Centrifugal Compressors. NACA TN 2421, 1951.

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

8. Walsh, J. L., Ahlberg, J. H., and Nilson, E. N.: Best Approximation Properties of the Spline Fit. Jour. Math. and Mech., vol. 11, no. 2, Mar. 1962, pp. 225-234.

NASA-Langley, 1964 E-2469