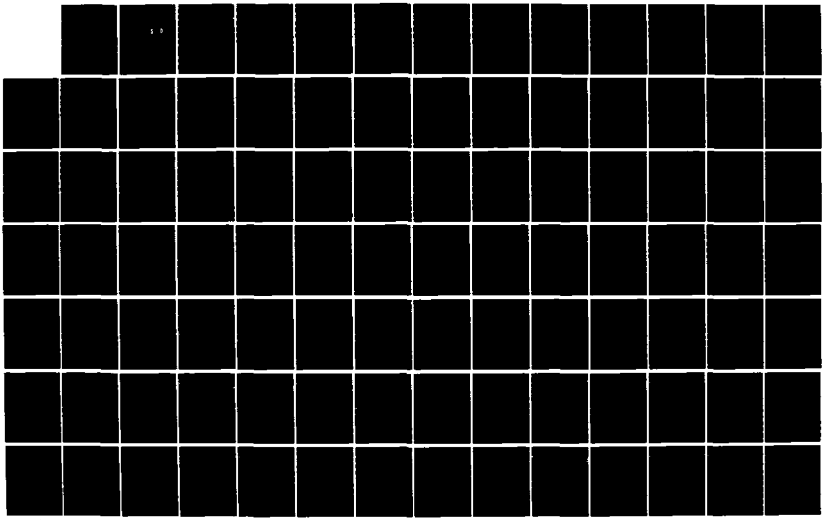
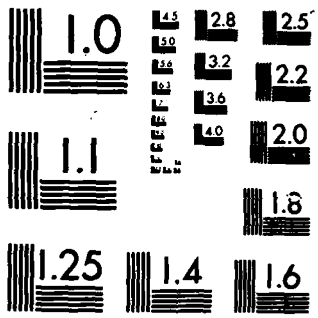
AD-A164 070 RESEARCH ON TURBINE FLOWFIELD ANALYSIS METHODS(U) 1/2
CALSPAN ADVANCED TECHNOLOGY CENTER BUFFALO NY W J RAE
JAN 85 CALSPAN-7177-A-3 AFOSR-TR-85-1220
UNCLASSIFIED F49620-83-C-0096 F/G 20/4 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(2)

AFOSR-TR- 85-1220

AD-A164 070

RESEARCH ON TURBINE FLOWFIELD ANALYSIS METHODS

William J. Rae (State Univ. of New York @ Buffalo)
Physical Sciences Department
Calspan Advanced Technology Center
P.O. Box 400
Buffalo, New York 14225

January 1985

Prepared for:

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
BOLLING AIR FORCE BASE, DC 20332

CONTRACT NO. F49620-83-C-0096
AFOSR FINAL SCIENTIFIC REPORT

86 2 11 072

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A164 070

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for Public Release; Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): 7177-A-3	5. MONITORING ORGANIZATION REPORT NUMBER(S): AFOSR-TR- 85-1220
6a. NAME OF PERFORMING ORGANIZATION: Calspan Advanced Technology Center; 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION: AFOSR/NA
6c. ADDRESS (City, State and ZIP Code): P. O. Box 400, Buffalo, NY 14225	7b. ADDRESS (City, State and ZIP Code): Bolling AFB DC 20332-6448
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Air Force Office of Scientific Research; 8b. OFFICE SYMBOL (If applicable): NA	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: F49620-83-C-0096
8c. ADDRESS (City, State and ZIP Code): Bolling AFB, DC 20332	10. SOURCE OF FUNDING NOS.: PROGRAM ELEMENT NO. 61102F; PROJECT NO. 2307; TASK NO. A1; WORK UNIT NO.
11. TITLE (Include Security Classification): Research on Turbine Flowfield Analysis Methods	
12. PERSONAL AUTHOR(S): William J. Rae (State University of New York at Buffalo)	
13a. TYPE OF REPORT: Final; 13b. TIME COVERED: FROM 1Apr83 TO 30Nov84	14. DATE OF REPORT (Yr., Mo., Day): January 1985; 15. PAGE COUNT: 110
16. SUPPLEMENTARY NOTATION	
17. COSATI CODES: FIELD / GROUP / SUB. GR.	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Turbomachinery, Compressors, Grid Generation, Transonic Flow, Conformal Mapping.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report contains a brief description of several procedures which use conformal mapping to generate computational grids for turbomachinery flowfield calculations, and to find the incompressible potential flow on such a grid.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: Dr James D Wilson	22b. TELEPHONE NUMBER (Include Area Code): 202/767-4935; 22c. OFFICE SYMBOL: AFOSR/NA

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

ABSTRACT

This report contains a description of conformal-mapping procedures that can be used to generate computational grids for turbomachinery flowfield calculations, and to determine the incompressible potential flow on such a grid. The mapping procedures represent an extension of the Ives transformation to blade rows having a high solidity. The flowfield solution takes advantage of the fact that one of the mapping steps takes the blade row into a unit circle; by writing down the classical source/sink/vortex solution in this circle, it is possible to find the incompressible potential flow in the original cascade. This solution is of interest in its own right, and provides a useful initial condition for iterative or time-marching calculational methods.

Accesion For	
NTIS CRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

FOREWORD

This document contains a final report of research done on Contract No. F49620-83-C-0096, entitled "Research on Turbine Flowfield Analysis Methods". The initial effort was presented in an interim report (Ref. 1a), and a recent summary was given in Ref. 1b. Technical effort was curtailed for a period of time when the Principal Investigator (Dr. W. J. Rae) became a member of the faculty in the Department of Mechanical and Aerospace Engineering at the State University of New York at Buffalo. Arrangements were made to resume the effort at the University on a subcontract basis and a no-cost time extension through the end of 1984 was granted to facilitate completion of the program. The program remained under the direction of Dr. Paul V. Marrone, Head, Physical Sciences Department, Calspan ATC.

This document contains the completion of the work initiated in Ref. 1. The results presented make it possible to generate computational grids for cascades of high solidity, and to calculate the imcompressible potential flow on such a grid.

1a. Rae, W.J., "Revised Computer Program for Evaluating the Ives Transformation in Turbomachinery Cascades" Calspan Report No. 7177-A-1 (AFOSR Report No. TR-83-1284) July 1983

1b. Rae, W.J. (State University of New York at Buffalo) and Marrone, P.V. "Research on Turbine Flowfield Analysis Methods" Calspan Report No. 7177-A-2 April 1984

TABLE OF CONTENTS

Section 1
INTRODUCTION

Some of the most important recent advances in computational fluid dynamics have been made possible by the use of algorithms which solve the equations of motion in boundary-conforming coordinates (see, for example, Refs. 2-4). Thus, the development of methods for carrying out these coordinate transformations has received considerable attention (for example, Ref. 5). The field of turbomachinery flows has been the object of a major fraction of this attention, partly because the constraints imposed by internal flows make the application of certain contemporary algorithms more difficult than is the case for external flows. In particular, the presence of non-orthogonality or shearing in the computational grid tends to make the algorithms less stable, and the solutions are more sensitive to the methods used for treating, in the computational plane, the regions corresponding to trailing edges and the points at upstream and downstream infinity. References 6 & 7 are typical of one approach to the resolution of the latter class of problems. The net conclusion drawn from that work is that there is a need for improved grid-generation techniques for turbomahcinery, particularly with respect to the treatment of trailing-edge regions and to the achievement of orthogonal grids.

The grid-generation methods in use today can be grouped into three categories: those which apply algebraic shearing or stretching of the coordinates, those which are generated numerically by solving a Poisson equation, and those which are based on conformal mapping. The current research is concerned with an example from the third category, namely the transformation introduced by Ives and Liutermoza.[8,9,10] Attempts to use this method, as reported in Ref. 7 encountered the problem of non-orthogonality, especially when applied to the high solidity blade rows typical of present designs (slant gap/chord ratios on the order of 0.7 or less). The primary goal of the present research was to extend the Ives-Liutermiza technique so as to apply to turbine blade rows of high solidity.

As the first step in achieving this goal, an Interim Report[1] was prepared, documenting a number of improvements that had been made since issuance of Ref. 9. The version of the method presented in Ref. 1 also segregated the portions of the method in which the improvements, to achieve orthogonal grids at high solidity, were to be made.

The principal modification contained in the present report is the replacement of the Theodorsen-Garrick mapping (used as one of the steps in Refs. 1,8-10) with a derivative form, as suggested by Bauer et al[(11)]. Use of this replacement makes it possible to handle blade rows of high solidity. In addition, a further step involving the Schwarz-Christoffel transformation can be used, to generate a fully orthogonal grid. These new developments are described in Section 2 (along with an updated version of Ref. 1, and 8-10, for completeness).

The potential flow through a cascade of blades can be written down algebraically, once the cascade has been mapped into a unit circle. This information has been added as Section 3, using one of the mapping steps in which the blade-surface image is mapped into a circle.

The final sections contain a description of the computer code and comments on its range of applicability.

Section 2
CONFORMAL TRANSFORMATION METHOD

The method of Ives & Liutermoza consists of a sequence of transformations, which map a two-dimensional cascade into a rectangle. The notation and coordinate system used to define the cascade are shown in Fig. 1. The x-coordinate is measured in the axial direction, and y is perpendicular to x.

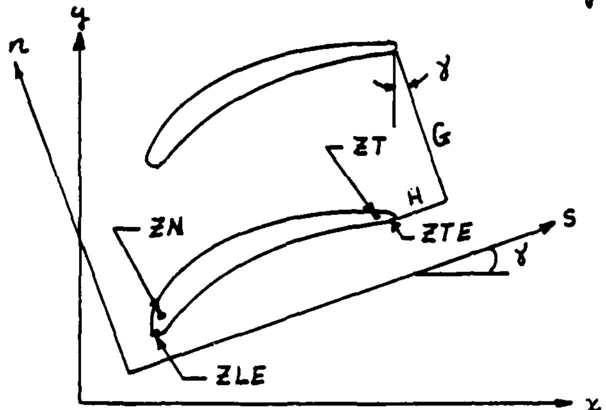


Figure 1. Coordinate System

The quantities s, n and the angle γ denote the "streamwise, normal" coordinates, in terms of which the blade profiles are sometimes defined. These reduce to the x, y set if H is taken as zero. The origins of both of these coordinate systems are arbitrary.

These coordinates define complex variables z and z_{xy}

$$z = s + in \quad , \quad z_{xy} = x + iy \tag{2-1}$$

The points ZN and ZT are taken anywhere near the centers of curvature of the leading and trailing edges, while ZLE and ZTE are points which divide the "pressure" side of the blade (i.e., its concave surface) from the "suction" side (its convex surface). These points can be chosen anywhere on the leading- and trailing-edge contours; ZTE is the point that will be connected to the "point" at downstream infinity by one of the grid lines, if that option is chosen (i.e., ISHEAR=1).

For the case of a sharp trailing edge (ITE=0), ZT must equal ZTE, and for a sharp leading edge (ILE=0), ZN must equal ZLE. The included angle at a sharp trailing edge, τ , must be specified. This is illustrated in Fig. 2, for the cascade used in Ref. 7.

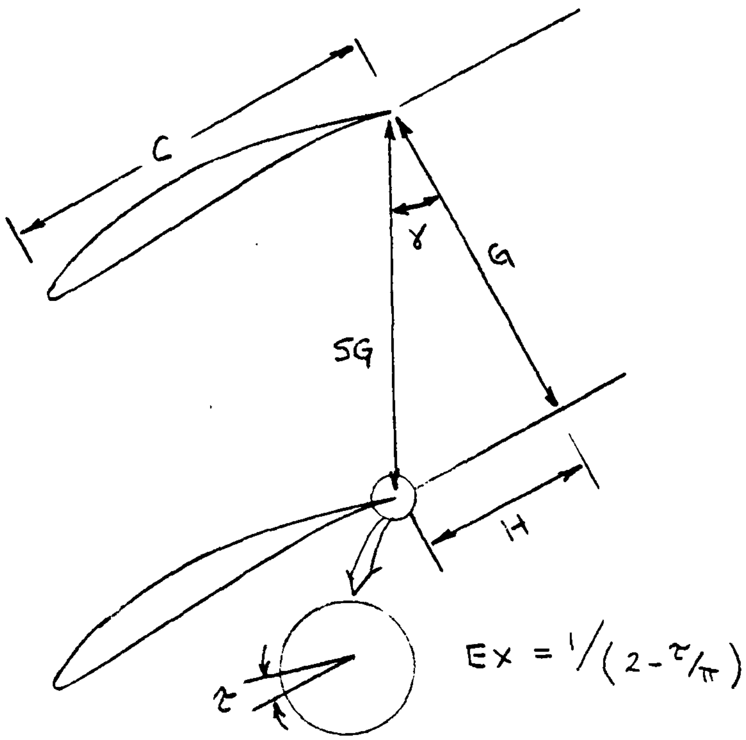


Figure 2. Blade Geometry of Ref. 7

This blade row uses the NACA 65(12)10 profile, has a slant-gap chord ratio $SG/C = 1$, where SG is the slant gap

$$SG = \sqrt{H^2 + G^2} \tag{2-2}$$

and a stagger angle γ of 28.4°. This geometry is used, below, to illustrate the steps in the Ives transformation. A comparable set of illustrations, for a cascade of turbine blades with rounded trailing edges, is given in Ref. 10.

Figure 2 shows the relation between the trailing-edge angle τ and an exponent Ex (which is used in one of the transformation steps described below). This relation applies only for a sharp trailing edge. For a rounded trailing edge, Ex is not related to the 180-degree trailing-edge angle, but is chosen as a number in the range 0.2 to 0.4, as described below.

The blade shape is input as two tables of coordinate pairs, one for the pressure surface, and one for the suction surface. These coordinates, plus the leading- and trailing-edge points ZLE and ZTE are then arranged in an array indexed by KJ, where KJ=1 at the trailing edge, and where the numbering proceeds around the pressure side to the leading edge (KJLE), and then along the suction side to the trailing edge, where the point denoted by KJMX is a repeat of that denoted by KJ=1. This notation is shown in Fig. 3.

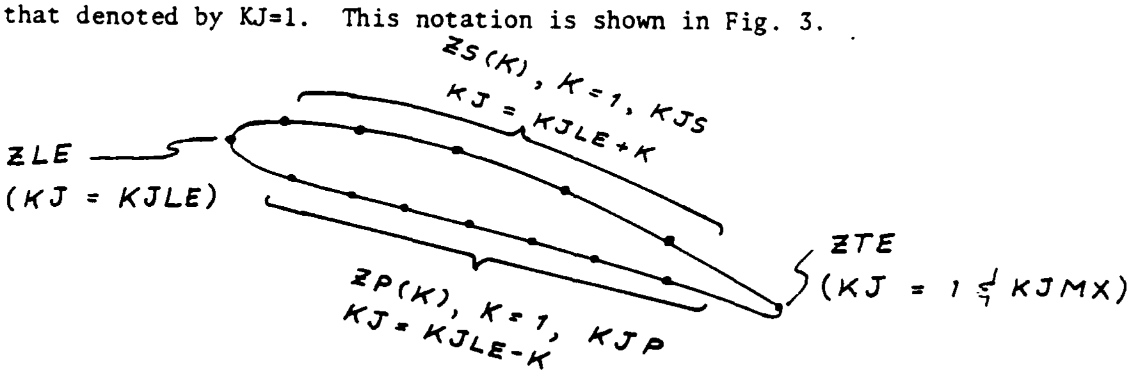


Figure 3. Notation for Blade-Surface Coordinates

The quantities KJS and KJP need not be equal; they are limited to a maximum value of 80 by a dimension statement in the current version of the program.

The first step in the Ives transformation is

$$q(z) = \frac{\sin\left\{\pi\,\dfrac{z - z_T}{H + iG}\right\}}{\sin\left\{\pi\,\dfrac{z - z_N}{H + iG}\right\}} \equiv \frac{\sin\zeta_1}{\sin\zeta_3} \equiv \frac{\zeta_2}{\zeta_4} \qquad (2\text{-}3)$$

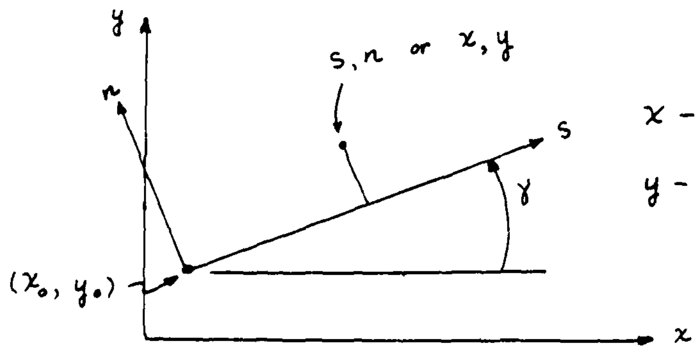
The fact that only differences of z -values are used is what accounts for the arbitrariness of the origin in Fig. 1. On the suction (S) and pressure (P) sides, the function $q(z)$ has the Fortran equivalents:

$$q(z) = \begin{cases} RDS(K)\, e^{i\,THS(K)} \\ RDP(K)\, e^{i\,THP(K)} \end{cases} \qquad (2\text{-}4)$$

The arguments of the sine functions can be written in a simpler form, as follows:

$$\begin{aligned} \zeta_{1,3} &= \pi\,\frac{\Delta z\,(H - iG)}{H^2 + G^2} \\ &= \frac{\pi}{SG}\Big\{\mathrm{Re}\,\Delta z\cdot\sin\gamma + \mathrm{Im}\,\Delta z\cdot\cos\gamma \\ &\qquad + i\left[\mathrm{Im}\,\Delta z\cdot\sin\gamma - \mathrm{Re}\,\Delta z\cdot\cos\gamma\right]\Big\} \end{aligned} \qquad (2\text{-}5)$$

where Δz stands for $z - z_T$ or $z - z_N$ in the expressions for ζ_1 and ζ_3 , respectively. By noting that (see Fig. 4)



$$x - x_0 = S\cos\gamma - n\sin\gamma$$

$$y - y_0 = S\sin\gamma + n\cos\gamma$$

Figure 4. Coordinate Relations

it follows that

$$\zeta_1 = \frac{\pi}{SG}\left\{(s - s_T)\sin\gamma + (n - n_T)\cos\gamma + i\left[(n - n_T)\sin\gamma - (s - s_T)\cos\gamma\right]\right\}$$

$$= \frac{\pi}{SG}\left\{y - y_T - i\,(x - x_T)\right\} \tag{2-6}$$

Similarly,

$$\zeta_3 = \frac{\pi}{SG}\left\{y - y_N - i\,(x - x_N)\right\} \tag{2-7}$$

Thus the function g can be written as

$$g = \frac{\sinh\left[\frac{\pi}{SG}\left(z_{XY} - z_{XYT}\right)\right]}{\sinh\left[\frac{\pi}{SG}\left(z_{XY} - z_{XYN}\right)\right]}$$

where $$z_{XYT,N} = x_{T,N} + i\,y_{T,N}$$

The shape of the blade-surface image in the g-plane can best be illustrated by considering a cascade of airfoils consisting of flat plates with circular leading and trailing edges:

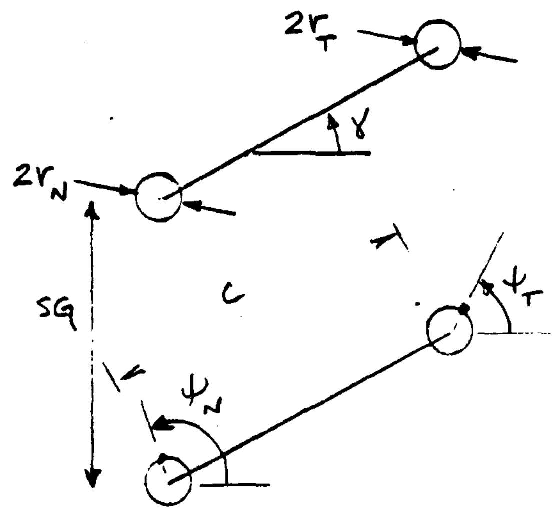


FIGURE 5 NOTATION FOR FLAT-PLATE CASCADE

For points z_{xy} that lie on the circle at the trailing edge, the corresponding location in the q -plane, to first order in r_T , is

$$q \approx \frac{\pi}{sG} \frac{r_T}{R} e^{i(\psi_T - \beta)}$$

where

$$R^2 = \sinh^2\left(\frac{\pi c}{sG} \cos\gamma\right) \cos^2\left(\frac{\pi c}{sG} \sin\gamma\right) + \cosh^2\left(\frac{\pi c}{sG} \cos\gamma\right) \sin^2\left(\frac{\pi c}{sG} \sin\gamma\right)$$

$$\beta = \tan^{-1} \frac{\cosh\left(\frac{\pi c}{sG} \cos\gamma\right) \sin\left(\frac{\pi c}{sG} \sin\gamma\right)}{\sinh\left(\frac{\pi c}{sG} \cos\gamma\right) \cos\left(\frac{\pi c}{sG} \sin\gamma\right)}$$

The variation of β with γ and sG/c is shown in Fig. 6. For points z_{xy} that lie on the circle at the leading edge, the corresponding relation, to first order in r_N , is

$$q \approx \frac{sG}{\pi} \frac{R}{r_N} e^{i(\pi + \beta - \psi_N)}$$

Along the upper surface of the flat-plate portion of the airfoil, the coordinates in the physical plane have the values:

$$z_{xy} - z_{xyT} = r\, e^{i(\gamma + \pi)} \; ; \; z_{xy} - z_{xyN} = \sigma\, e^{i\gamma}$$

where r is measured from z_{xyT} and σ from z_{xyN} , and

$$r + \sigma = c$$

The resulting formula for q shows that at midchord (i.e., $r = \sigma = c/2$), $q = -1$.

These general features of the q-plane are shown in Fig. 7, for a negative stagger angle. Note that a clockwise path around the airfoil, starting from the trailing edge, leads to a clockwise path around the small circle in the q -plane, followed by a transit to the large circle along one side of the flat-plate image, and then a counter-clockwise path around the large circle, and back

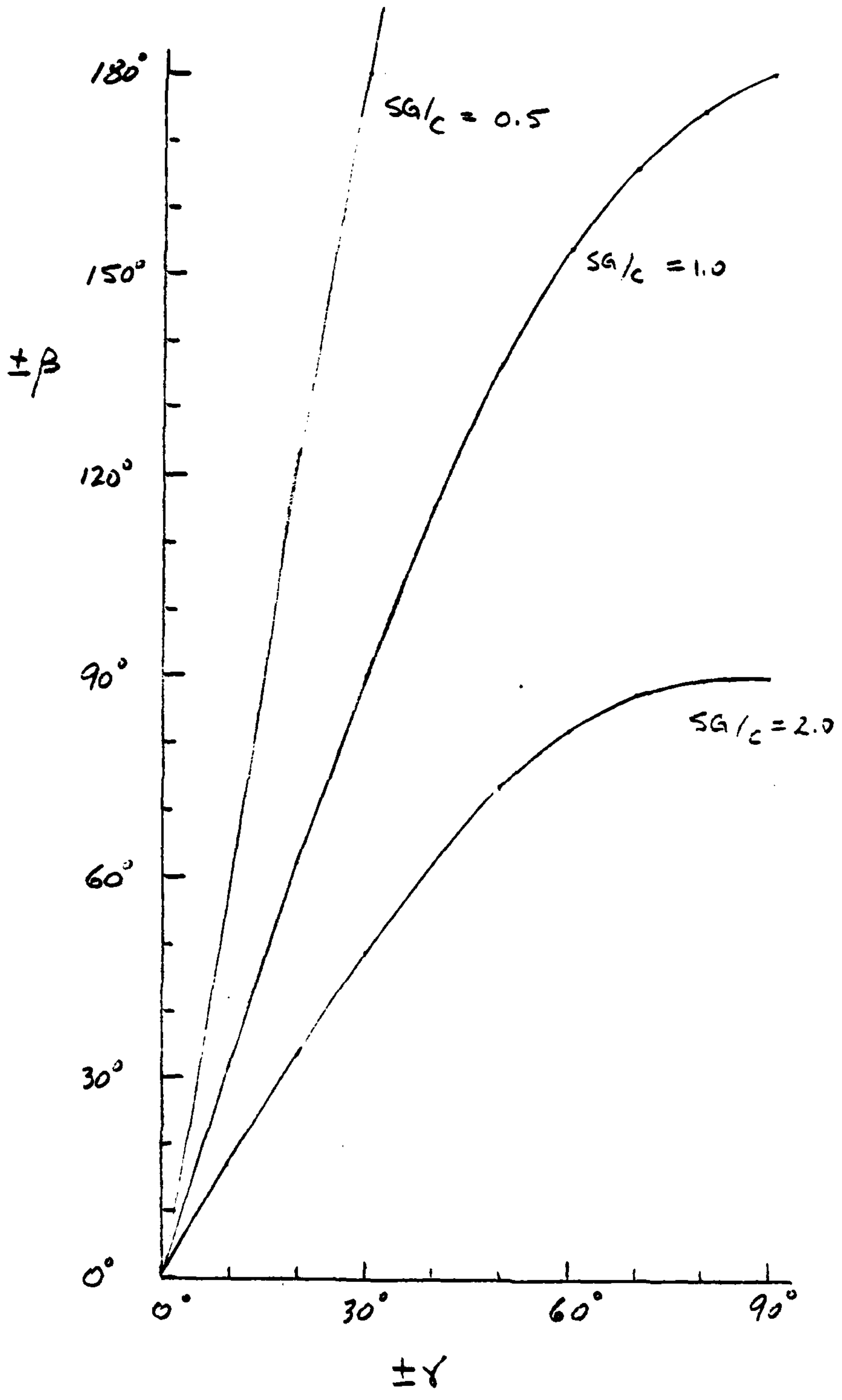


FIGURE 6 β, γ RELATION FOR FLAT-PLATE CASCADE

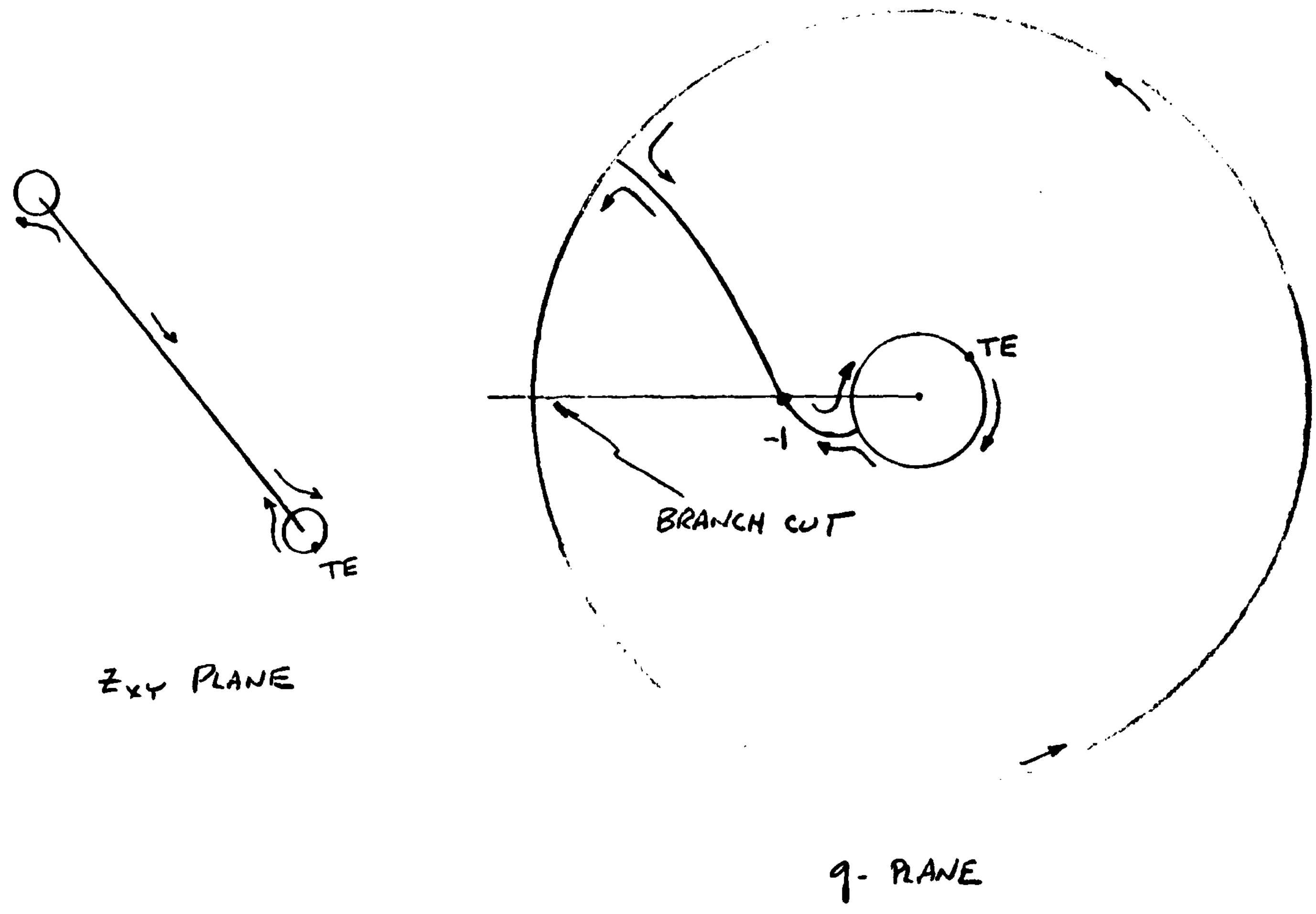


FIGURE 7 THE MAPPING $g(z)$

to the small circle along the other side of the flat-plate image. For blades of finite thickness, the flat-plate image becomes a pair of lines, which fair smoothly into the large and small circles.

In writing the polar angle of any point in the g-plane, care must be taken to retain continuity across the cut (along the negative real axis) that is used by the FORTRAN ATAN2 function. This is achieved by writing:

$$\arg[g] = \mathrm{ATAN2}\left[\mathrm{IMAG}(g),\ \mathrm{REAL}(g)\right] + 2\pi \cdot BR$$

where the 'branch number' BR is set equal to zero at the trailing-edge point $KJ=1$ (or $KJ=2$ for a sharp trailing edge) if the value returned by ATAN2 at this point is positive (and BR is set equal to -1, if the value returned is negative), and where BR is incremented/decremented each time the branch cut is crossed in the counterclockwise/clockwise direction. For a sharp leading edge, the increment occurring at the leading-edge point must be added specifically, since the radius of the large circle in the g-plane is infinite for this case.

The next step is the exponentiation, defined as:

$$\Omega = \left[g(z)\right]^{1/K} \quad ; \quad K = 2 - \frac{\tau}{\pi} = EXINV \equiv \frac{1}{EX} \qquad (2\text{-}8)$$

The result of this step is shown in Fig. 8, for the cascade depicted in Fig. 2. The value of EX for a sharp trailing edge is fixed by the angle τ ; for a round trailing edge, a value of EX in the neighborhood of 0.2 to 0.5 will usually reduce the variation of $\arg\Omega$ so that the resulting curve in the Ω-plane will have a polar radius that is a single-valued function of $\arg\Omega$ near the trailing edge.

The next transformation is a bilinear one, whose purpose is to produce a curve in the ω -plane that can be mapped into a unit circle in a subsequent step:

$$c\,\frac{\omega - a}{\omega - b} = \Omega \quad ; \quad \omega = \frac{a - b\,\frac{\Omega}{c}}{1 - \frac{\Omega}{c}} \qquad (2\text{-}9)$$

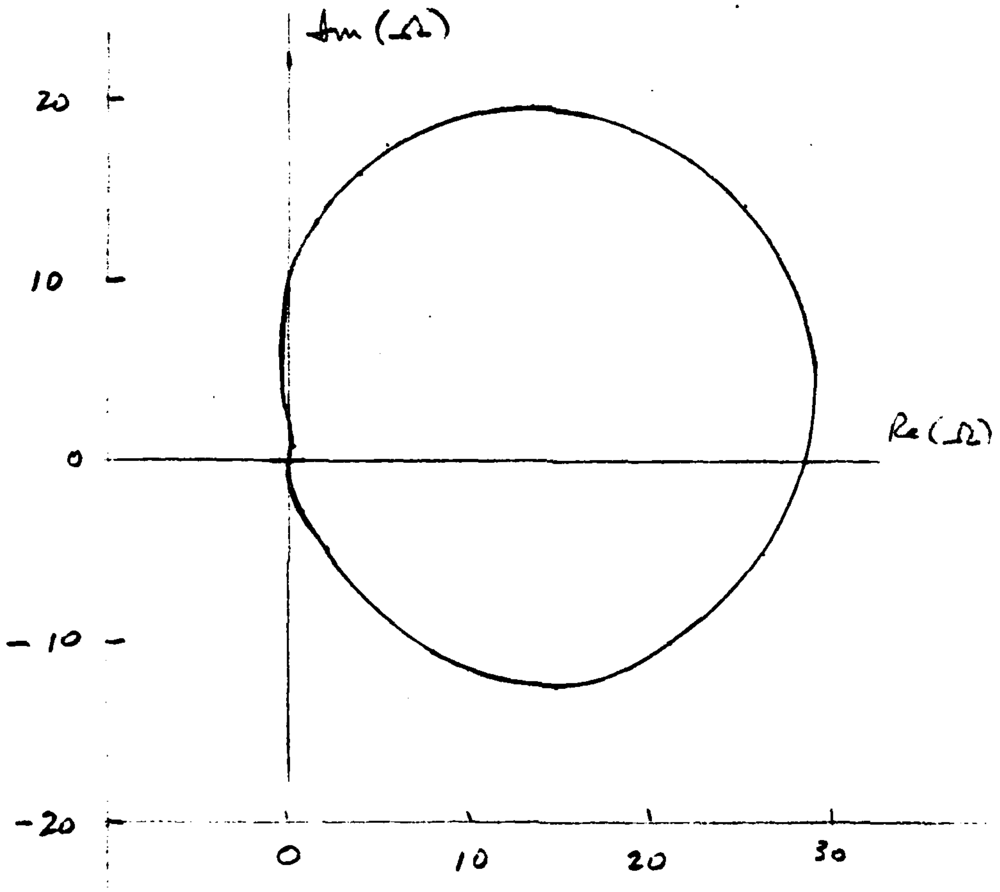


FIGURE 8 THE MAPPING $\Omega(z)$

where a, b, and c are complex constants. Three conditions must be assigned, in order to evaluate these constants: Ives suggests,[8] for two of them, that the images of upstream and downstream infinity be mapped into $\omega = \pm 1$. As the third condition, he recommends that the centroid of the blade-surface image in the ω-plane be forced to be close to the origin. This condition was applied in Ref. 9; however, it has been found simpler to impose a condition on the ratio between the maximum and minimum radii in the ω-plane, as outlined below. The calculation of the centroidal location has been retained in the present code, for informational purposes. (Details on how this is calculated can be found in Ref. 9).

The locations of the images of upstream and downstream infinity in the g- and Ω-planes can be expressed as follows: in general,

$$g(z_{xy}) = \frac{\sin\left[\frac{\pi}{SG}(y-y_T)\right]\cosh\left[\frac{\pi}{SG}(x-x_T)\right] - i\cos\left[\frac{\pi}{SG}(y-y_T)\right]\sinh\left[\frac{\pi}{SG}(x-x_T)\right]}{\sin\left[\frac{\pi}{SG}(y-y_N)\right]\cosh\left[\frac{\pi}{SG}(x-x_N)\right] - i\cos\left[\frac{\pi}{SG}(y-y_N)\right]\sinh\left[\frac{\pi}{SG}(x-x_N)\right]}$$

As $x \to -\infty$ at finite y (see line 1 of Fig. 9), the large-argument approximations to the hyperbolic functions give

$$g(-\infty) = \exp\left\{\frac{\pi}{SG}(z_T - z_N)\right\}$$

Similarly, as $x \to +\infty$ at finite y (see line 2 of Fig. 9), the corresponsing limit is

$$g(+\infty) = \exp\left\{-\frac{\pi}{SG}(z_T - z_N)\right\}$$

These are written as

$$g(-\infty) = e^{\mathrm{Im}(\zeta_1 - \zeta_3)}\, e^{-i\,\mathrm{Re}\,\zeta_1} \equiv e^{X \cdot K}\, e^{i X_A \cdot K} \tag{2-10}$$

$$g(+\infty) = e^{-X \cdot K}\, e^{-i X_A \cdot K} \tag{2-11}$$

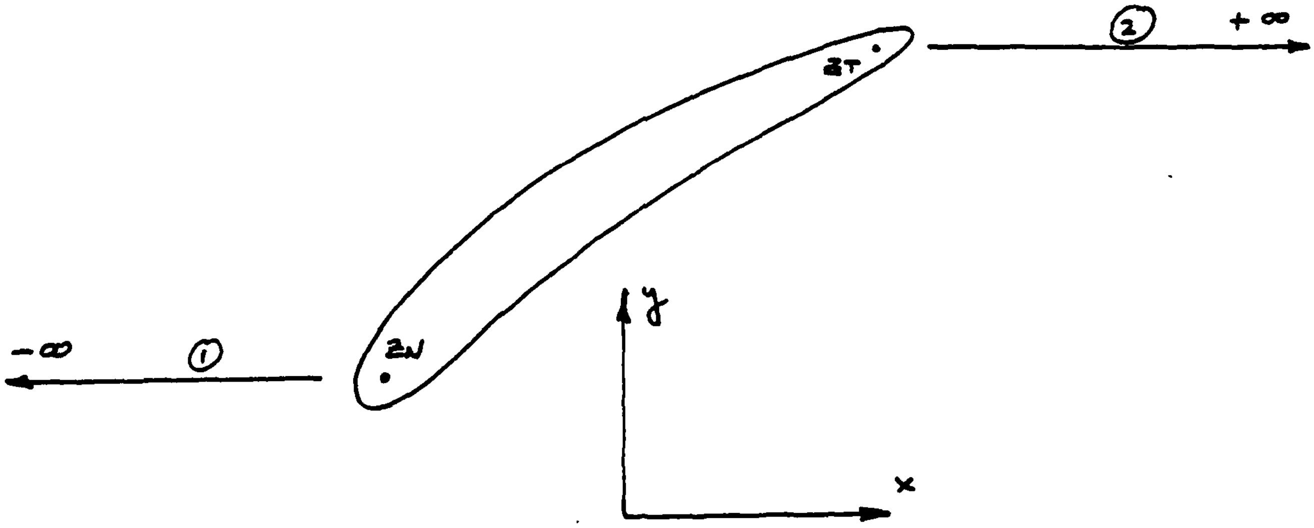


Figure 9. Location of the Points at Infinity

where

$$\chi \cdot K \equiv \mathrm{Im}(\zeta_1 - \zeta_3) = \frac{\pi}{SG}\left\{ \mathrm{Re}(Z_T - Z_N)\cos\gamma - \mathrm{Im}(Z_T - Z_N)\sin\gamma \right\}$$

$$\chi_A \cdot K \equiv \mathrm{Re}(\zeta_1) = -\frac{\pi}{SG}\left\{ \mathrm{Re}(Z_T - Z_N)\sin\gamma + \mathrm{Im}(Z_T - Z_N)\cos\gamma \right\}$$

Thus

$$\Omega^- \equiv [g(-\infty)]^{1/K} = e^{\chi} e^{i\chi_A}, \quad \Omega^+ \equiv [g(+\infty)]^{1/K} = e^{-\chi} e^{-i\chi_A} \tag{2-12}$$

Note that

$$\Omega^+ \Omega^- = 1$$

The constants a and b can be expressed in terms of C by the two equations:

$$C \frac{-1-a}{-1-b} \equiv \Omega^-$$

$$C \frac{1-a}{1-b} \equiv \Omega^+$$

The solution is:

$$b = \frac{-2c + \Omega^- + \Omega^+}{\Omega^+ - \Omega^-} \quad ; \quad a = 1 - (1-b)\frac{\Omega^+}{c}$$

$$= -Ec + F \quad ; \quad = \frac{E}{c} - F \tag{2-13}$$

where E and F are known quantities:

$$E \equiv \frac{2}{\Omega^+ - \Omega^-}, \quad F \equiv \frac{\Omega^+ + \Omega^-}{\Omega^+ - \Omega^-}. \tag{2-14}$$

In terms of the parameters E, F, and c, the transformation can be written as

$$\omega = \frac{E - Fc + (Ec - F)\,\Omega}{c - \Omega} \tag{2-15}$$

$$\Omega = \frac{(\omega + F)c - E}{\omega - F + Ec} \tag{2-16}$$

The latter relations can be used in an iteration procedure to find a value of c that will minimize the ratio RMAX/RMIN, where RMAX and RMIN are the maximum and minimum values of $|\omega|$ over the set defined by KJ=1 to KJMX. The iteration process is as follows: on alternate iterations, values of c are chosen that will either reduce RMAX or increase RMIN. This is done by solving Equation 2-16 for c:

$$c = \frac{\Omega\omega + E - F\Omega}{\omega + F - E\Omega} \tag{2-17}$$

On the first iteration, $c = 1 + i$ is used; the values of RMIN and the index KJ=KJMN at which it occurs are then found. Next, a new value of c is calculated, using Eq. 2-17, such that the new value of ω (KJMN) will equal 1.1 times the value just found. For this new mapping into the ω-plane, the value of RMAX and the index KJ=KJMXX at which it occurs are found. For the third iteration, a value of c is used such as to generate a new value of ω (KJMXX) equal to 0.9 times the previous one. This alternating cycle is then continued until the ratio RMAX/RMIN is less than the tolerance RTOL. This tolerance is assigned a default value of 3.0; values up to 6.0 have been handled successfully by the subsequent steps in the transformation.

The iteration on c can be bypassed, if c is already known, by setting IGOT=1 and reading in the value of c. Figure 10 is the ω-plane for the cascade shown in Figure 2.

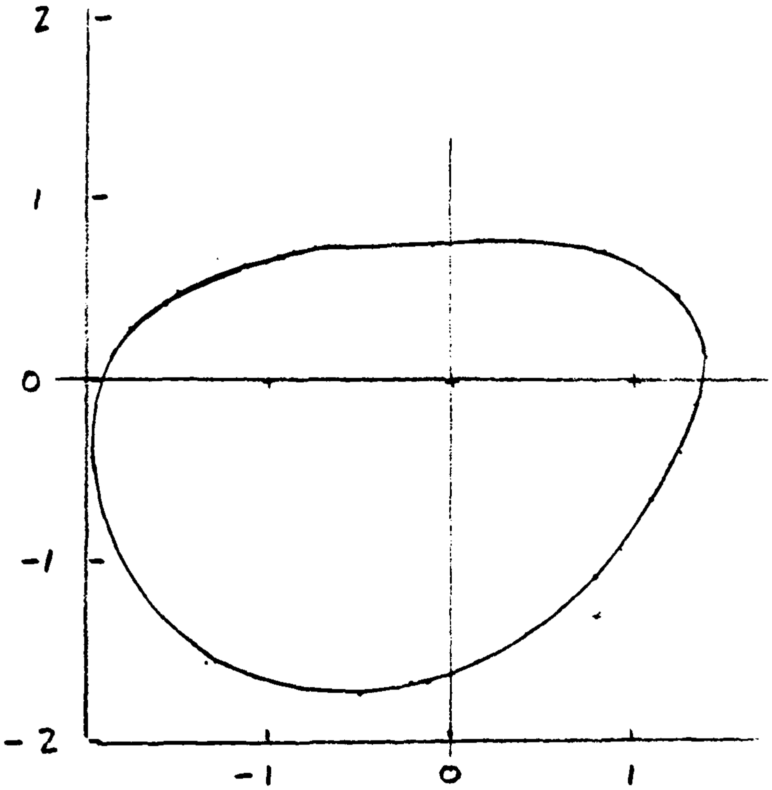


FIGURE 10 THE MAPPING $\omega(z)$

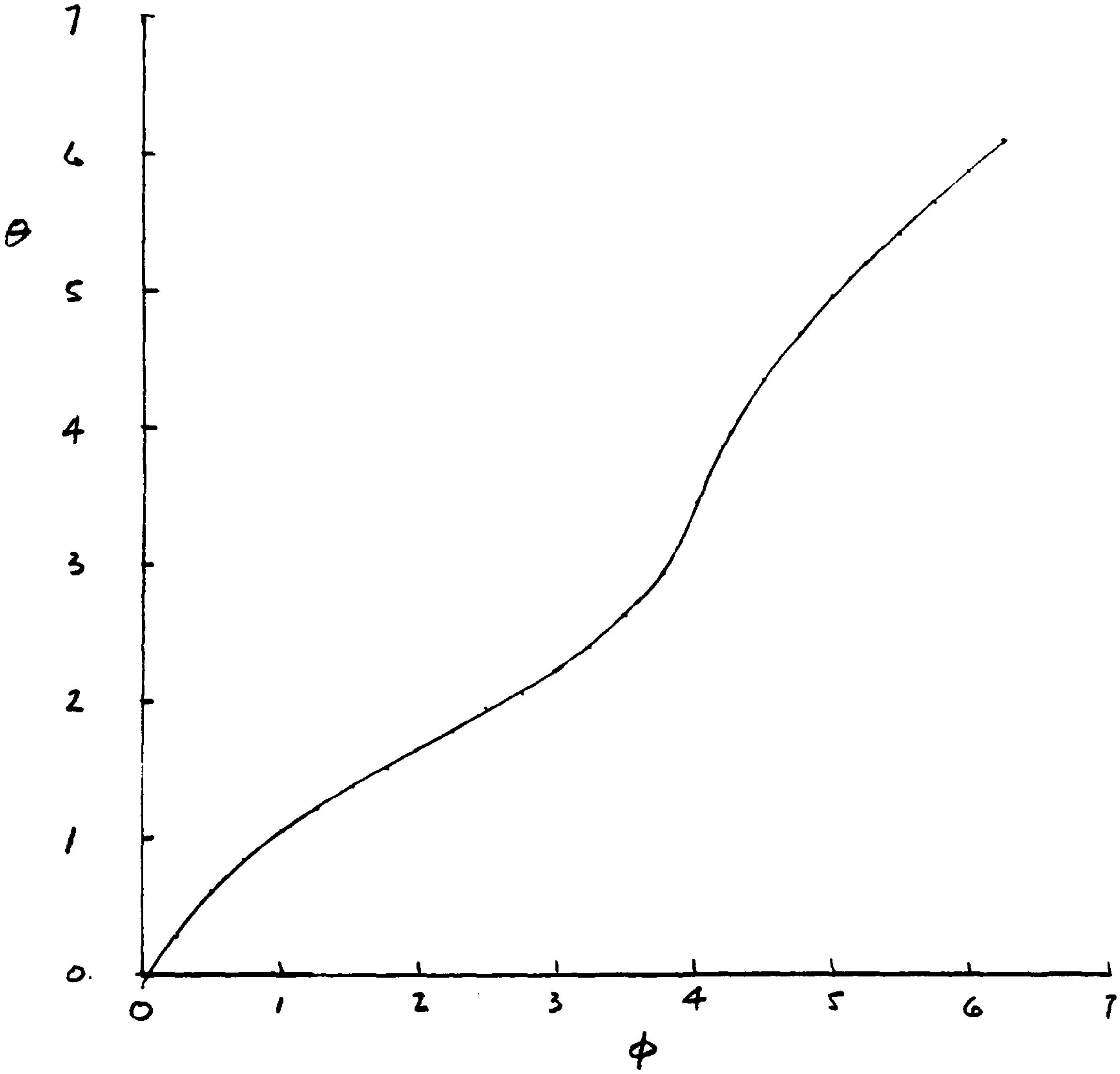


FIGURE 11 VARIATION OF θ vs. ϕ ON THE BLADE SURFACE

Next, the blade-surface image in the ω-plane is mapped into the unit circle in the ζ-plane with the trailing edge at $\zeta = 1$ by either of two methods. The first is the Theodorsen-Garrick transformation:

$$\omega = \zeta \exp\left\{ \sum_{j=0}^{N} (A_j + iB_j)\, \zeta^{j} \right\} \tag{2-18}$$

To determine the coefficients, the values of ω and ζ on the blade surface are written as

$$\omega = r(\theta)\, e^{i\theta} \quad , \quad \zeta = 1\, e^{i\phi} \tag{2-19}$$

Then the real and imaginary parts of the transformation are

$$\ln r = A_0 + \sum_{j=1}^{N} \left[A_j \cos j\phi - B_j \sin j\phi \right] \tag{2-20}$$

$$\theta = \phi + B_0 + \sum_{j=1}^{N} \left[A_j \sin j\phi + B_j \cos j\phi \right] \tag{2-21}$$

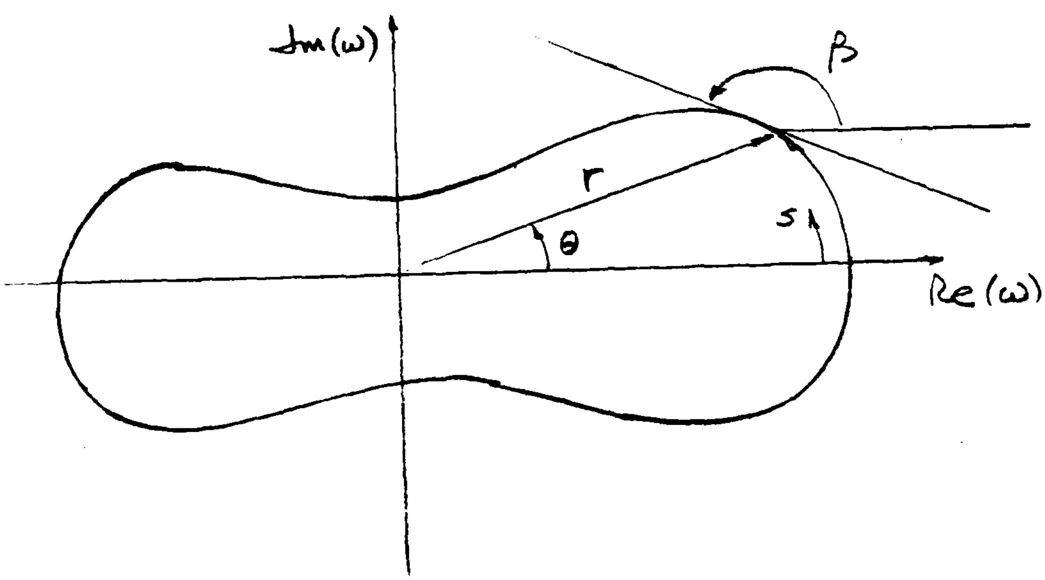


FIGURE 12 INTRINSIC COORDINATES

The coefficients are then determined by the following iteration procedure: an equally-spaced array of values of ϕ is set up, and all the coefficients are initially set equal to zero. Then the second equation gives $\theta = \phi$ as the first approximation for θ . For each of these values of θ , a corresponding value of $\ln r$ can be found, from a spline fit to the coordinates of the blade-surface image in the ω-plane. These known values of $\ln r$ are then used in the first of the equations above, to find the next approximation to the A_j and B_j coefficients. These coefficients can then be used to give the next approximation to $\theta(\phi)$, and the process is continued until convergence is reached, to some preassigned tolerance. Fast Fourier transform tech- techniques[12] can be used in the processes of evaluating the second equation, for known values of the coefficients, and of determining the coefficients from the first equation with known values of $\ln r$. These techniques were applied in the present calculations. The details are given in Appendix A.

Sufficient conditions for convergence of this iteration process have been discussed by Warschawski.[13] For the present case, these conditions are not met; in particular, it is required that the maximum and minimum values of $r(\theta)$ obey the relation:

$$\sqrt{\frac{RMAX}{RMIN}} - 1 < 0.295 \qquad \text{or} \qquad \frac{RMAX}{RMIN} < 1.678$$

This condition is seldom met in practice. It was found, however, that the iteration process would converge if a relaxation parameter was used. i.e.,

12. Cooley, J.W., Lewis, P.A.W., and Welch, P.D., "The Fast Fourier Transform Algorithm: Programming Considerations in the Calculations of Sine, Cosine and Laplace Transforms", Journal of Sound and Vibration 12, (1970) pp. 315-337.

13. Warschawski, S.E., "On Theodorsen's Method of Conformal Mapping of Nearly Circular Regions", Quarterly of Applied Mathematics 3, (1945) pp. 12-28.

if a relaxation parameter was used, i.e., the values of θ called for by the second equation [called θ^*] were not used in the first equation, but were replaced by $\theta_{new} = 0.1\,\theta^* + 0.9\,\theta_{old}$. With this relaxation factor, the iterations were convergent: after 68 iterations, the maximum change in any of the values of θ was less than 10^{-2} radians. The variation of θ with ϕ is shown in Fig. 11. Calculations for other cases, not shown here, have required relaxation factors as low as 0.02 for convergnece. A recent review paper by Henrici (Reference 14) calls attention to the applicability of under-relaxation in this problem.

When the solidity is high, the curve of θ vs ϕ becomes extremely steep; typical results, for a gap/chord ratio of 0.8, are shown in Ref. 10. For such cases, a preferable approach is to use a derivative form of the transformation[11]:

$$\frac{d\omega}{d\zeta} = \exp\left\{ \sum_{j=0}^{N} D_j \zeta^j \right\}$$

To determine the coefficients, the ω-plane is expressed in terms of the intrinsic coordinates s and β, where s is arclength measured from the trailing edge, and β is the angle between the real axis and the tangent to the curve (see Fig. 12). The relations between the intrinsic coordinates and the polar angle ϕ in the ζ-plane are:

$$s = \int_{\omega_{TE}}^{\omega} |d\omega| = \int_0^{\phi} \left| \frac{d\omega}{d\zeta} \right| d\phi$$

since $|d\zeta| = d\phi$ and

$$\arg \frac{d\omega}{d\zeta} = \arg d\omega - \arg d\zeta = \beta - (\phi + \pi/2)$$

If the Fourier coefficients are written as

$$D_j = DR_j + i\, DI_j$$

Then the basic equation can be split into its real and imaginary parts as:

14. Henrici, P., "Fast Fourier Methods in Computational Complex Analysis", SIAM Review 21, (1979) pp. 481-527.

$$\ln\left|\frac{dw}{d\zeta}\right| = DR_0 + \sum_{j=1}^{N} \left(DR_j \cos j\phi - DI_j \sin j\phi\right)$$

$$\arg \frac{dw}{d\zeta} = DI_0 + \sum_{j=1}^{N} \left(DI_j \cos j\phi + DR_j \sin j\phi\right)$$

The coefficients D_j are then found by the following steps:

1. β and S are found from the numerical data that define the blade-surface image in the ω -plane
2. A first guess at $|dw/d\zeta|$ is made, at equally spaced points on the unit circle in the ζ -plane.
3. The corresponding values of the arclength are found, from

$$s = \frac{S_{max}}{\int_0^{2\pi}\left|\frac{dw}{d\zeta}\right| d\phi} \cdot \int_0^{\phi}\left|\frac{dw}{d\zeta}\right| d\phi$$

4. A spline fit of the β, s data is used to evaluate, at the equi-spaced ϕ -values, the quantity:

$$\arg \frac{dw}{d\zeta} = \beta - \left(\phi + \pi/2\right)$$

5. The Fast Fourier Transform is then used to find the coefficients which fit the data of step 4.
6. These coefficients are then used (again with the FFT) to evaluate the next approximation to $|dw/d\zeta|$, and steps 3 through 6 are repeated until convergence is achieved.

The details of this procedure are very similar to those involved in the Theodorsen-Garrick technique; the FFT steps described in Appendix A can be taken over, with only minor changes.

This procedure was found to converge in 15 iterations (to an arclength error less than .03) without relaxation. The resulting variations of s and β with ϕ are shown in Figure 13.

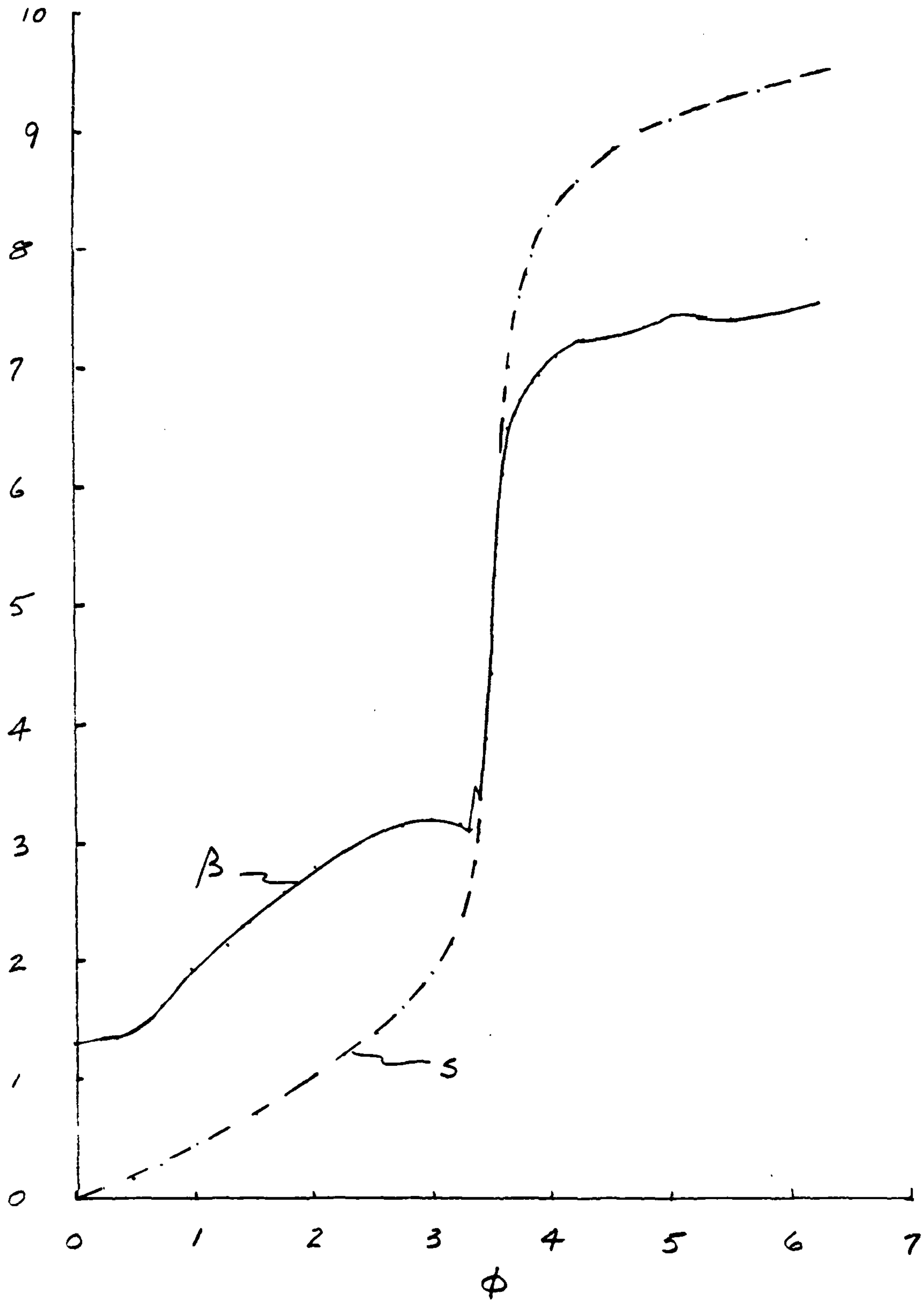


FIGURE 13 INTRINSIC COORDINATES FOR CASCADE OF FIG. 2

The next transformation is

$$\eta = \gamma \, \frac{\zeta - \alpha}{\zeta - \beta} \tag{2-22}$$

where α, β, and γ are chosen so as to place the images of $z = \pm\infty$ at $\eta = \pm S$, while the blade surface continues to be the unit circle. These images are located, respectively, at $\omega = \pm 1$, and $\zeta = \zeta_A, \zeta_B$. Explicit formulas for α, β, and γ in terms of ζ_A and ζ_B are given by Ives[8] as

$$x = \frac{2 - |\zeta_A + \zeta_B|^2 + 2\,|\zeta_A \zeta_B|^2}{|\zeta_A - \zeta_B|^2}$$

$$S = \min\left\{ \sqrt{\left| x + \sqrt{x^2 - 1} \right|} \;,\; \sqrt{\left| x - \sqrt{x^2 - 1} \right|} \right\}$$

$$\alpha = \frac{2\zeta_A \zeta_B + [S^2(\zeta_A - \zeta_B) - (\zeta_A + \zeta_B)] / \tilde{\zeta}_A}{S^2(\zeta_A - \zeta_B) + (\zeta_A + \zeta_B) - 2/\tilde{\zeta}_A}$$

$$\beta = \frac{2\zeta_A \zeta_B - \alpha(\zeta_A + \zeta_B)}{\zeta_A + \zeta_B - 2\alpha}$$

$$\gamma = S \, \frac{\zeta_A - \beta}{\zeta_A - \alpha} \tag{2-23}$$

Mokry (Reference 15) has pointed out that the formula for $\mathcal{S}$ can be simplified, as follows: define

$$C \equiv \frac{\zeta_A \bar{\zeta}_B - 1}{\zeta_A - \zeta_B}$$

Then

$$\mathcal{S} = |C| - \sqrt{|C|^2 - 1} \quad , \quad \gamma = \frac{C - \mathcal{S}|C|\bar{\zeta}_B}{|C|\bar{\zeta}_B - \mathcal{S}C}$$

$$\alpha = \frac{\zeta_B C - \mathcal{S}|C|}{C - \mathcal{S}|C|\bar{\zeta}_B} \quad , \quad \beta = \frac{|C| - \mathcal{S}C\zeta_B}{|C|\bar{\zeta}_B - \mathcal{S}C} \qquad (2\text{-}24)$$

The computer program described below does not use these simplifications.

Next, it is necessary to find ζ_A and ζ_B, given $\omega = \pm 1$. This was done by Newton-Raphson iteration:

$$\zeta^{(n+1)} = \zeta^{(n)} - \frac{G(\zeta)}{\frac{dG}{d\zeta}}$$

where for the Theodorsen-Garrick procedure

$$G(\zeta) = \zeta \exp\left\{\sum_{j=0}^{N} (A_j + iB_j)\zeta^j\right\} - \omega$$

$$\frac{dG}{d\zeta} = \frac{\omega}{\zeta}\left[1 + \sum_{j=1}^{N} j(A_j + iB_j)\zeta^j\right]$$

15. Mokry, M., "Comment on Analysis of Transonic Cascade Flow Using Conformal Mapping and Relaxation Techniques", AIAA Journal 16, No. 1, (January 1978) p. 96.

In order to find an initial guess for ζ, a preliminary calculation was made, for

$$|\zeta| = 0.49\ (.1)\ 0.99, \qquad \arg\zeta = -60^\circ\ (10^\circ)\ +60^\circ$$

From this set, the value of ζ which gave ω nearest to +1 was chosen as the initial guess. This set was then repeated with ζ replaced by $-\zeta$, to obtain the value of ζ that gave ω nearest to -1. The locations of key points in the ζ and η planes are shown in Fig. 14.

When using the derivative form of the ω, ζ transformation, it is necessary to integrate the derivative $d\omega/d\zeta$; this was done using Simpson's Rule, starting at the trailing edge:

$$\omega - \omega_{TE} = \int_1^\zeta \frac{d\omega}{d\zeta}\, d\zeta$$

The equations used in this case for the Newton-Raphson procedure are

$$G(\zeta) = \omega_{TE} + \int_1^\zeta \frac{d\omega}{d\zeta}\, d\zeta - \omega$$

$$\frac{dG}{d\zeta} = \frac{d\omega}{d\zeta} = \exp \sum_{j=0}^{N} D_j \zeta^j .$$

When the values of ζ_A and ζ_B are known, the blade-surface-image points can be mapped from the ζ -plane to the η -plane. In both planes, these points are located on unit circles, so it is only necessary to interpolate in Fig. 12 to find the ζ -values for the points defining the blade surface. This is done with a call to the spline-fit subroutine, and the values returned by it are replaced by linear interpolation in regions where the θ, ϕ curve is so steep that the spline fit returns non-monotonic values.

The final transformation now uses an elliptic function to map the unit circle in the η -plane (with cuts along the real axis from $\pm S$ to the circle) into a rectangle:

$$\eta = S \, sn\,(\tilde{\xi}, k) \tag{2-25}$$

where the parameter k is given by

$$k = S^2 \tag{2-26}$$

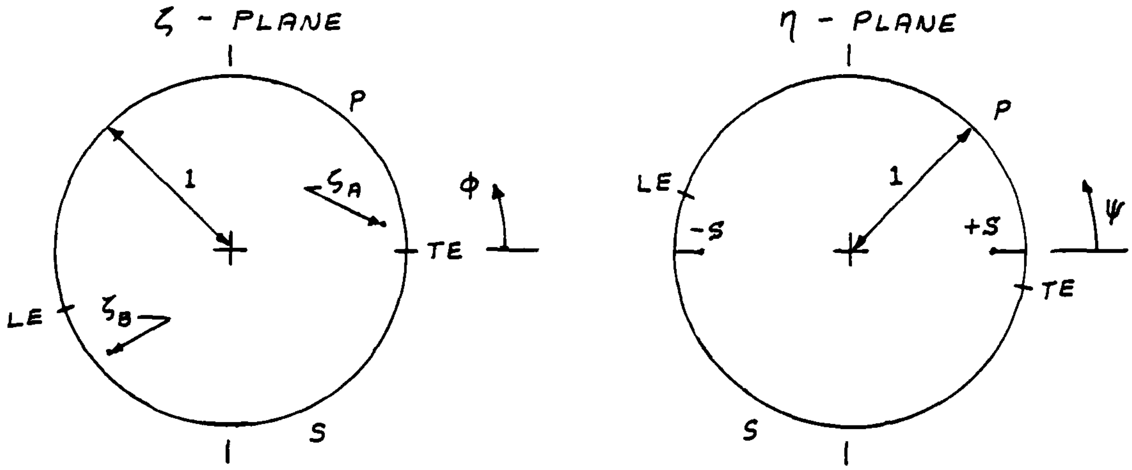


Figure 14 The ζ and η Planes

The inverse of this transformation is

$$\tilde{\xi} = \int_0^{\eta/S} \frac{dt}{\sqrt{1-t^2}\;\sqrt{1-k^2t^2}} \qquad (2\text{-}27)$$

This latter transformation is used to find the images, in the $\tilde{\xi}$ -plane, of the blade-surface points. This requires an expression for the real and imaginary parts of the incomplete elliptic integral of the first kind; convenient formulae for this purpose are given by Nielsen and Perkins[16] who show that, if

$$\eta = S(\tau + i\delta) \qquad (2\text{-}28)$$

then

$$\tilde{\xi} = F(\sqrt{\lambda}, k) + i F(\sqrt{\sigma}, k') \qquad (2\text{-}29)$$

where

$$k' = \sqrt{1-k^2} = \sqrt{1-S^4} \qquad (2\text{-}30)$$

16. Nielsen, J.N. and Perkins, E.W., "Charts for the Conical Part of the Downwash Field of Swept Wings at Supersonic Speeds", NACA Technical Note 1780, (December 1948), Appendix C.

and where F (· · ·) denotes the incomplete elliptic integral of the first kind, with real arguments λ and σ given as functions of τ, δ and k :

$$\lambda = \left[1 + \tau^2 + \delta^2 - \sqrt{(1-\tau^2)^2 + \delta^2(\delta^2 + 2 + 2\tau^2)} \right] \left[1 + k^2(\tau^2 + \delta^2) \right.$$

$$\left. - \sqrt{(1 - k^2\tau^2)^2 + k^2\delta^2(2 + 2k^2\tau^2 + k^2\delta^2)} \right] / 4k^2\tau^2$$

$$\sigma = \left[\tau^2 + \delta^2 - \lambda \right] / \left[\tau^2 + \delta^2 - \lambda + 1 - \lambda k^2(\tau^2 + \delta^2) \right] \quad (2\text{-}31)$$

These formulae are equivalent to those following Eqn. 115.01 of Byrd and Friedman;[17] the formula for λ has been rearranged slightly from the form given by Nielsen and Perkins, to avoid the occurence of negative values under the square root sign, which can sometimes happen in the numerical evaluations when $\delta = 0$. These formulas are correct along the branch cuts, where $\delta = 0$ and $|\eta|/\delta > 1$.

Numerical evaluations of these elliptic integrals were done using twelve terms in the formulae of Luke:[18]

$$F(y, k) \equiv \int_0^y \frac{dt}{\sqrt{1-t^2}\,\sqrt{1-k^2t^2}} = \int_0^{\phi = \sin^{-1} y} \frac{d\psi}{\sqrt{1 - k^2 \sin^2\psi}}$$

$$\approx F_{12}(\phi, k) = \frac{1}{25}\left[\phi + 2\sum_{m=1}^{12} \frac{\tan^{-1}(\sigma_m \tan\phi)}{\sigma_m} \right] \quad (2\text{-}32)$$

---

17. Byrd, P.F., and Friedman, M.D., Handbook of Elliptic Integrals for Engineers and Physicists. Springer Verlag, Berlin (1954).

18. Luke, Y.L., "Approximations for Elliptic Integrals", Mathematics of Computation, 22 (1968) 627-634.

where

$$\phi < \pi/2 \ , \quad \sigma_m = \sqrt{1 - k^2 \sin^2 \theta_m} \ , \quad \theta_m = m\pi/25 \ .$$

For the case where $\phi = \pi/2$ (the complete integral) the approximate formula is

$$F_{12}\left(\frac{\pi}{2}, k\right) = \frac{\pi}{50}\left[1 + 2\sum_{m=1}^{12} \frac{1}{\sigma_m}\right] \tag{2-33}$$

The signs in the formulas above pertain to the first quadrant in the η-plane ($\tau \geq 0$, $\delta \geq 0$), which maps into a rectangle in the first quadrant of the $\tilde{\xi}$-plane, with sides located on the lines

$$Re(\tilde{\xi}) = K(k) = \int_0^1 \frac{dt}{\sqrt{1-t^2}\sqrt{1-S^4 t^2}}$$

$$Im(\tilde{\xi}) = \frac{1}{2} K'(k) = \frac{1}{2}\int_0^1 \frac{dt}{\sqrt{1-t^2}\sqrt{1-(1-S^4)\,t^2}} \tag{2-34}$$

The remaining three quadrants in the η-plane map into the remaining three quadrants in the $\tilde{\xi}$-plane, as described in Reference (19), p. 377; the cuts from $\pm S$ to the unit circle along the real axis become the left and right sides of the rectangle in the $\tilde{\xi}$-plane:

19. Erdelyi, A., et al. Higher Transcendental Functions, Volume 2, p. 377, McGraw-Hill Book Company, New York (1953).

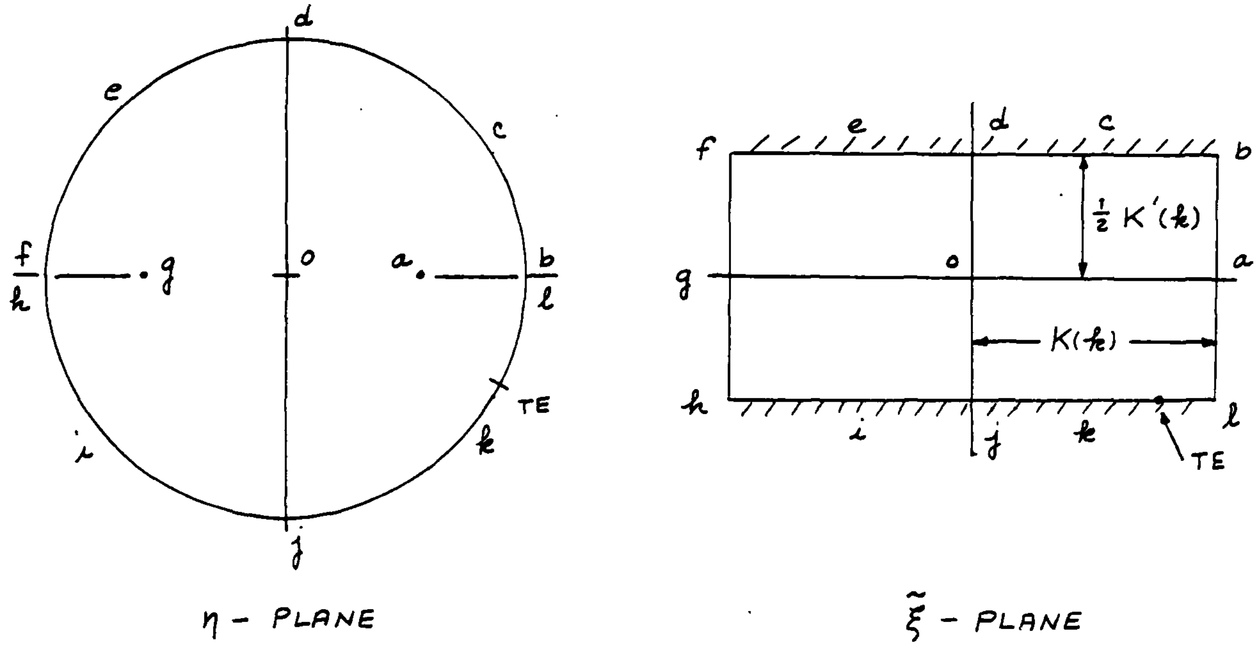


Figure 15 The η and $\tilde{\xi}$ Planes

Finally, the $\tilde{\xi}$ - plane is re-normalized, so as to lie between -1 and +1 on both axes:

$$z_{MAPPED} = \frac{Re(\tilde{\xi})}{K(k)} + i\,\frac{Im(\tilde{\xi})}{\frac{1}{2}K'(k)} \tag{2-35}$$

A grid is now to be set up in the $\tilde{\xi}$ plane, and mapped back to the z - plane. This process is facilitated by first rearranging the quadrants in the $\tilde{\xi}$ - plane, by using the periodicity of the elliptic functions as follows: in the first and second quadrants, let $\hat{\xi} = \tilde{\xi}$, and in the third and fourth let $\hat{\xi} = -(\tilde{\xi} + 2K(k))$ and use the relations (see for example, Eqs. 122.00 and 122.04 of Reference 17.

$$sn(\tilde{\xi}) = -sn(\tilde{\xi}+2K) = sn\left[-(\tilde{\xi}+2K)\right] = sn\left[-K-(\tilde{\xi}+K)\right] \qquad (2\text{-}36)$$

The $\hat{\xi}$ plane then has the form:

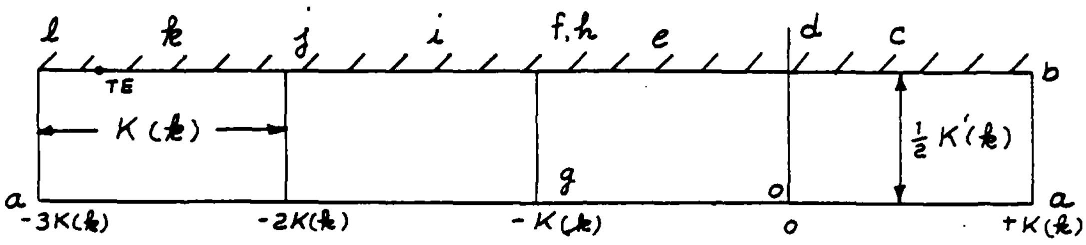


Figure 16 The $\hat{\xi}$ Plane

Two types of grid can be selected in the $\hat{\xi}$ plane: a rectangular one (if ISHEAR=0) or a sheared one (if ISHEAR=1). The latter is the default.

For the rectangular grid, equally spaced points are assigned, according to

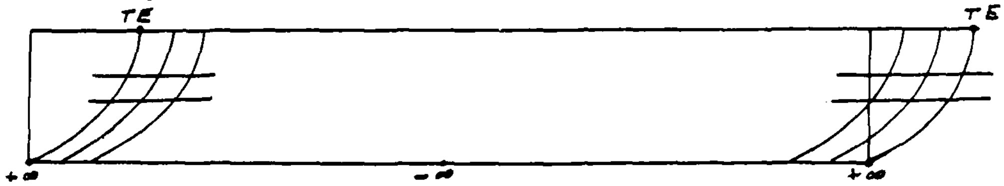
$$\hat{\xi}_R \equiv \mathcal{R}e(\hat{\xi}) = (K-1)\,\Delta\hat{\xi}_{real}\,, \quad K = 1,2,\ldots,KMX \qquad (2\text{-}37)$$
$$\hat{\xi}_I \equiv \mathcal{I}m(\hat{\xi}) = \frac{1}{2}K'(k) - (L-1)\,\Delta\hat{\xi}_{imag}\,, \quad L = 1,2,\ldots,LMX$$

where

$$\Delta\hat{\xi}_{real} = \frac{4K(k)}{KMX-1}\,, \quad \Delta\hat{\xi}_{imag} = \frac{\frac{1}{2}K'(k)}{LMX-1} \qquad (2\text{-}38)$$

Becuase the mapping is conformal, the images of these grid lines will intersect at right angles when mapped back to the physical plane.

The rectangular grid has the property that in general the trailing edge is not connected, by a grid line, to the point at downstream infinity. However, such a connection is a desirable feature in certain flowfield codes (see Ref. 7, for example). To allow this feature, the ISHEAR=1 option establishes a sheared grid, consisting of the same $\mathcal{I}m(\hat{\xi})$ lines as above, but replacing the $\mathcal{R}e(\hat{\xi})$ lines by a set of parabolas which intersect the blade surface at 90 degrees, and are displaced from a base parabola that connects the trailing edge to the image of downstream infinity:



This grid is given by

$$\hat{\xi}_I = \frac{1}{2} K'(k) - (L-1)\,\Delta\hat{\xi}_{imag} \quad , \quad L = 1, 2, \ldots, LMX$$

$$\hat{\xi}_R = \hat{\xi}_R(TE) + (K-1)\,\Delta\hat{\xi}_{real} + \frac{\left[\hat{\xi}_I - \frac{1}{2}K'(k)\right]^2}{const} \quad , \quad \begin{array}{l} K = 1, 2, \ldots, KMX \\ L = 1, 2, \ldots, LMX \end{array} \tag{2-39}$$

where

$$const = \frac{-\left[K'(k)\right]^2}{4\left[3K(k) + \hat{\xi}_R(TE)\right]} \quad ; \quad \hat{\xi}_R(TE) = \mathcal{R}e\left\{\hat{\xi}\left[KJ = 1\right]\right\} \tag{2-40}$$

Some points on these parabolas will lie outside the range

$$-3K(k) \le \hat{\xi}_R \le +K(k)$$

When this occurs, equivalent points are found by adding or subtracting $4K(k)$, the period of the elliptic sine. In addition, the base parabola is always joined to the lower-left corner of Fig. 15, by subtracting $4K(k)$ from the real part of $\hat{\xi}_{TE}$, if the latter is greater than $-K(k)$.

An alternate method of joing the images of the trailing edge and downstream infinity, while retaining an orthogonal grid, is described in Section 4. It involves use of the Schwarz-Christoffel transformation.

This completes the definition of the grid in the $\hat{\xi}$-plane. Each of these grid points must now be mapped back to the physical plane. The first transformation is:

$$\eta = \mathcal{S}\, sn(\tilde{\xi}\,;\, k) = \mathcal{S}\, sn(\xi\,;\, k) \tag{2-41}$$

The elliptic sine of a complex argument is expressed as (Ref. 11, Eq. 125.01)

$$sn(u + iv, k)$$

$$= \frac{sn(u,k)\, dn(v,k') + i\, cn(u,k)\, dn(u,k)\, sn(v,k')\, cn(v,k')}{1 - sn^2(v,k')\, dn^2(u,k)} \tag{2-42}$$

The functions in this expression are evaluated by the Arithmetic-Geometric Mean method (see Ref. 20, p. 571):

Set

$$a_o = 1 \quad , \quad b_o = k' \quad , \quad c_o = k$$

and then calculate

$$a_n = \frac{1}{2}(a_{n-1} + b_{n-1}) \quad , \quad b_n = \sqrt{a_{n-1}\, b_{n-1}}$$

$$c_n = \frac{1}{2}(a_{n-1} - b_{n-1}) \tag{2-43}$$

until $c_n = 0$ to a prescribed tolerance (10^{-7} was used in the present case.) Then form

$$\varphi_N = 2^N \cdot a_n \cdot u$$

and calculate $\varphi_{N-1}, \varphi_{N-2}, \cdots, \varphi_o$ from

$$\varphi_{n-1} = \frac{1}{2}\left\{\varphi_n + \arcsin\left(\frac{c_n}{a_n}\sin\varphi_n\right)\right\} \tag{2-44}$$

Then the desired results are given by

---

20. Abramowitz, M. and Stegun, I.A., Handbook of Mathematical Functions, National Bureau of Standards, Applied Mathematics Series 55 (1964).

$$sn(u,k) = \sin\varphi_0 \ , \quad cn(u,k) = \cos\varphi_0 \ ,$$

$$dn(u,k) = \frac{\cos\varphi_0}{\cos(\varphi_1 - \varphi_0)} \tag{2-45}$$

These values of η are then mapped back to the ζ - plane by

$$\zeta = \frac{\beta\eta - \alpha\gamma}{\eta - \gamma} \tag{2-46}$$

and thence to the ω- plane by either

$$\omega = \zeta \exp\left[\sum_{j=0}^{N} (A_j + i B_j)\zeta^j\right] \tag{2-47}$$

if the Theodorsen-Garrick transformation is benig used or by

$$\omega = \omega_{START} + \int_{\zeta_{START}}^{\zeta} \frac{d\omega}{d\zeta} d\zeta$$

if the derivative form of the ω, ζ transformation is being used. The values used for the starting point of this integration are as follows: at first, the equation is integrated from $\omega = -1$ to $\omega = +1$, $(\zeta = \zeta_B \text{ to } \zeta_A)$ so as to establish the periodic boundary. Then a series of integrations is done, at constant K, for L ranging from the blade surface to the periodic boundary (the values of ζ and ω on the blade-surface image were found earlier in the FFT procedure).

These integrations are overdetermined, in the sense that both end points are known, as well as the values of the derivatives in between. Moreover, the integrations used to establish the end points have been done by several techniques (among them, Simpson's rule and the FFT procedure). Thus the integrations from a given starting point do not always terminate at the desired end point; when this occurs, a small adjustment is made, in the values of the derivatives, so as to guarantee the proper end points.

Finally, the value of z_{XY} is found by inverting;

$$g(z) = \Omega^K = \left(c\,\frac{\omega - a}{\omega - b}\right)^K = \frac{\sinh\left[\frac{\pi}{SG}\left(z_{XY} - z_{XYT}\right)\right]}{\sinh\left[\frac{\pi}{SG}\left(z_{XY} - z_{XYN}\right)\right]}$$

whose inverse is

$$z_{XY} = \frac{SG}{2\pi} \ln \frac{\exp\left(\frac{\pi}{SG} z_{XYT}\right) - g\exp\left(\frac{\pi}{SG} z_{XYN}\right)}{\exp\left(-\frac{\pi}{SG} z_{XYT}\right) - g\exp\left(-\frac{\pi}{SG} z_{XYN}\right)}$$

The final result of this process is shown in Figure 17, for the cascade of Figure 2. This grid was found using the derivative form of the transformation; a comparable grid found with the Theodorsen-Garrick technique is shown in Ref. 7.

For higher solidities, the derivative form of the ω, ζ transformation can be used; Figure 18 shows the grid that results for a gap/chord ratio of 0.5. The grid points calculated for the region near the leading edge have been omitted from this figure, because they cannot be located with sufficient accuracy. The basic reason for this loss of accuracy can already be seen at a gap/chord ratio as high as 1.0 (see Fig. 13): the portion of the blade-surface image near the leading edge has a nearly vertical slope in this figure. Thus while the derivative form of the transformation does coverage for high solidities, the grid which it leads to suffers from a loss of accuracy near the leading edge.

This loss is aggravated, in the present case, by the adjustment of derivatives that is made, to enforce periodicity. Near the leading edge, the adjustments required become significant, and their cumulative effect is to produce a non-orthogonality at the blade surface. Further study is required, in order to minimize this problem of inaccuracy near the leading edge.

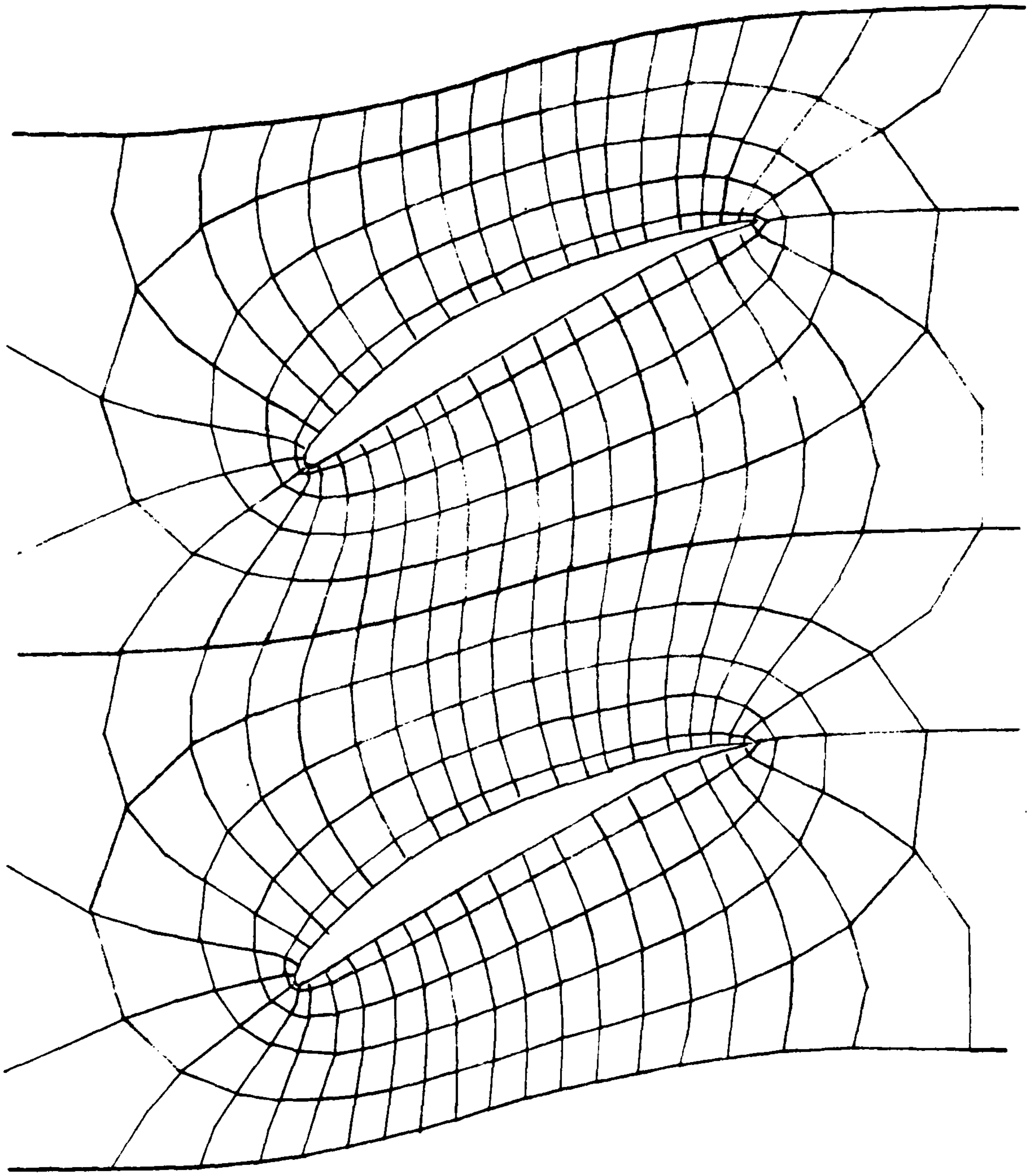


FIGURE 17 COORDINATE MAPPING SOLIDITY = 1

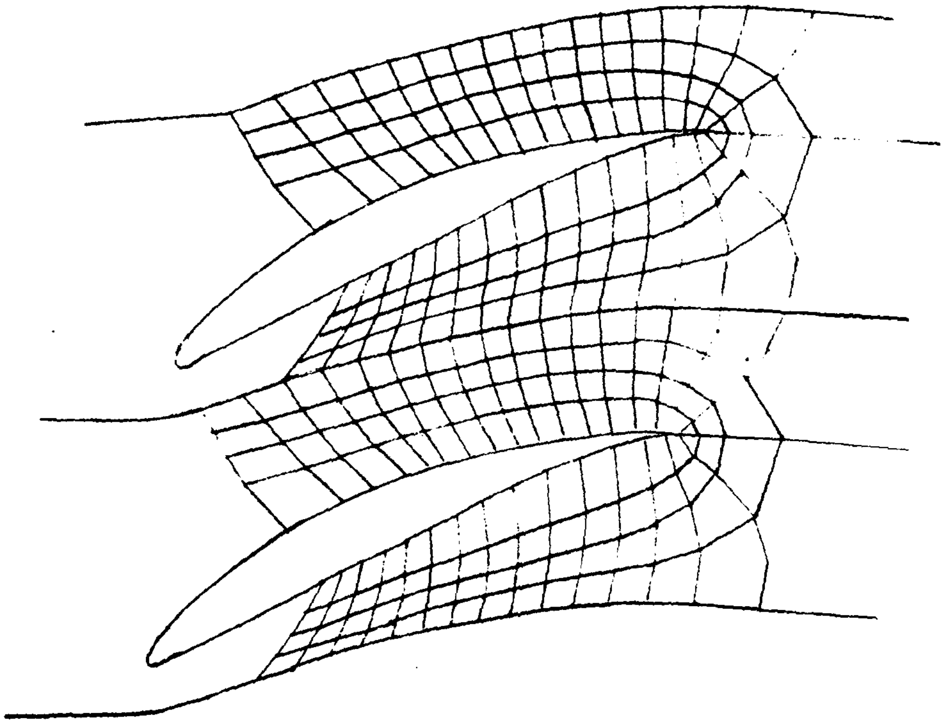


FIGURE 18 COORDINATE MAPPING HIGH SOLIDITY CASE

Section 3
POTENTIAL FLOW FIELD

It is possible to write down in algebraic form the complex potential for inviscid, incompressible flow in the η-plane, and this solution then gives the flow in the physical plane. The complex potential is (see, for example, Ref. 21).

$$W(\eta) = \phi + i\psi = Q\left[e^{-i\alpha}\ln\frac{1+s\eta}{1-s\eta} + e^{i\alpha}\ln\frac{\eta+s}{\eta-s}\right] + iG\ln\frac{(1+s\eta)(1-s\eta)}{(\eta+s)(\eta-s)}$$

where ϕ and ψ are the velocity potential and stream function, Q and G are source and vortex strengths, α is the angle of attack, and s the parameter described earlier. The complex velocity is:

$$\frac{dW}{d\eta} = Q\left[e^{-i\alpha}\left(\frac{s}{1+s\eta} + \frac{s}{1-s\eta}\right) + e^{i\alpha}\left(\frac{1}{\eta+s} - \frac{1}{\eta-s}\right)\right] + iG\left[\frac{s}{1+s\eta} - \frac{s}{1-s\eta} - \frac{1}{\eta+s} - \frac{1}{\eta-s}\right]$$

The Kutta condition, requiring that $dW/d\eta = 0$ at η_{TE} gives the following relation for G & Q :

$$\frac{iG}{Q} = \left.\frac{s\left\{e^{-i\alpha}(\eta^2-s^2) - e^{i\alpha}(1-s^2\eta^2)\right\}}{\eta(1-s^4)}\right|_{\eta=\eta_{TE}}$$

Putting this back into the expression for $dW/d\eta$, it is possible to find a second stagnation point (near the leading edge) at

$$\eta_{ST.PT.} = \frac{\frac{iG}{Q}(1-s^4) + \sqrt{-\left(\frac{G}{Q}\right)^2(1-s^4)^2 + 4s^2\left(1+2s^2\cos 2\alpha + s^4\right)}}{2s\left(e^{-i\alpha} + s^2 e^{i\alpha}\right)}$$

The values of G and Q are now found from conditions on the velocity at infinity; in general, the velocity components u and v in the x, y plane are given by:

$$u - iv = \frac{dW}{dz_{XY}} = \frac{dW}{d\eta}\frac{d\eta}{d\zeta}\frac{d\zeta}{d\omega}\frac{d\omega}{d\Omega}\frac{d\Omega}{dz_{XY}}$$

As $z_{XY} \to +\infty$, the image in the η-place is $\eta \to +S$; then:

$$\frac{dW}{d\eta} \cong \left[-Qe^{i\alpha} - iG\right]/(\eta - S)$$

Thus

$$\frac{dW}{dz_{XY}} = -\frac{Qe^{i\alpha}+iG}{\eta - S}\frac{d\eta}{dz_{XY}} = -\frac{Qe^{i\alpha}+iG}{\eta - S}\frac{d\eta}{d\zeta}\frac{d\zeta}{d\omega}\frac{d\omega}{d\Omega}\frac{d\Omega}{dz_{XY}}$$

But

$$\eta - S = \left.\frac{d\eta}{d\zeta}\right)_{\zeta_A}(\zeta - \zeta_A) = \left.\frac{d\eta}{d\zeta}\right)_{\zeta_A}\left.\frac{d\zeta}{d\omega}\right)_{+1}(\omega - 1) = \left.\frac{d\eta}{d\zeta}\right)_{\zeta_A}\left.\frac{d\zeta}{d\omega}\right)_{+1}\left.\frac{d\omega}{d\Omega}\right)_{\Omega(+\infty)}(\Omega - \Omega(\infty))$$

Thus

$$(u - iv)_{+\infty} = -\left[Qe^{i\alpha} + iG\right]\frac{d\Omega}{dz_{XY}}\Big/\left[\Omega - \Omega(\infty)\right]$$

By using the approximate form of the Ω, z_{XY} relation as $z_{XY} \to \infty$, it is possible to show that

$$\frac{d\Omega/dz_{XY}}{\Omega - \Omega(\infty)} = -\frac{2\pi}{SG}$$

Thus

$$(u - iv)_{+\infty} = \frac{Q\cdot 2\pi}{SG}\left[e^{i\alpha} + i\frac{G}{Q}\right]$$

A similar analysis shows that

$$(u - iv)_{-\infty} = \frac{Q\cdot 2\pi}{SG}\left[e^{i\alpha} - i\frac{G}{Q}\right]$$

To satisfy the continuity requirement that $u_{+\infty} = u_{-\infty}$ take $Q = U\cdot SG/2\pi$.

Then: $$u_{-\infty} = u_{+\infty} = U\cos\alpha$$

$$v_{-\infty} = +U\left[-\sin\alpha + G/Q\right] \qquad v_{+\infty} = U\left[-\sin\alpha - G/Q\right]$$

$$(u^2 + v^2)^{1/2}_{\pm\infty} = U\left\{1 + \frac{G^2}{Q^2} \pm 2\frac{G}{Q}\sin\alpha\right\}^{1/2}$$

These relations are shown in Figure 19; note that α is < 0 for positive incidence.

Once the constants G and Q are established, the velocities at any point in the flow can be found, by using

$$u - iv = \frac{dW}{dz_{XY}} = \frac{dW}{d\eta}\,\frac{d\eta}{d\zeta}\,\frac{d\zeta}{d\omega}\,\frac{d\omega}{d\Omega}\,\frac{d\Omega}{dz_{XY}}$$

Each of the derivatives in this formula can be evaluated exactly. A typical result is the streamline pattern shown in Figure 20.

This solution is of value in its own right (especially since it is found from purely algebraic formulae), and can also be useful as a set of initial conditions for a time-marching or iterative solution of the flow pattern.

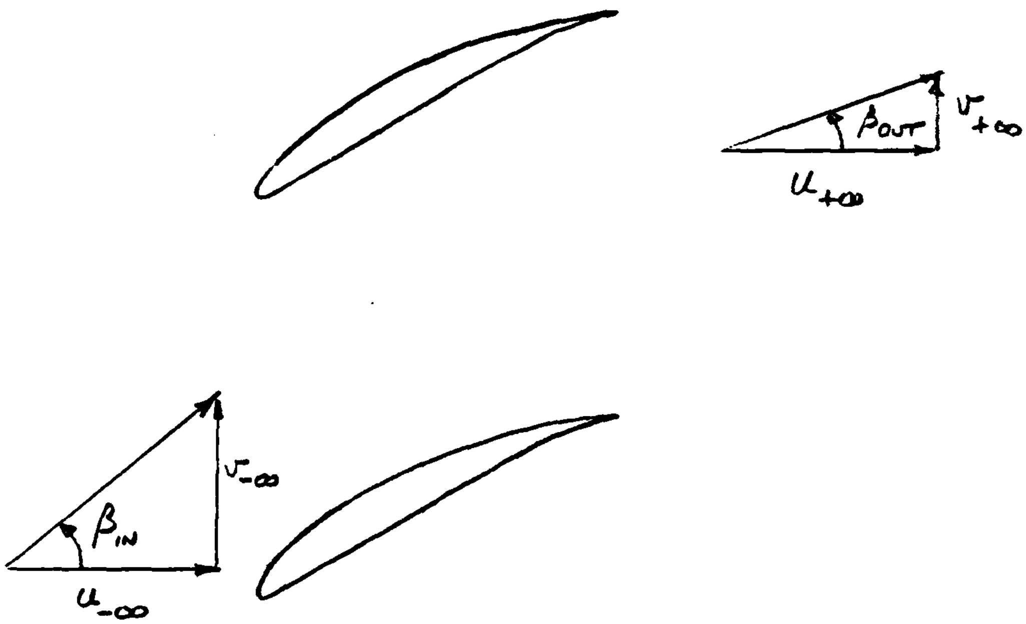


FIGURE 19 VELOCITY COMPONENTS
FAR UPSTREAM AND DOWNSTREAM OF CASCADE

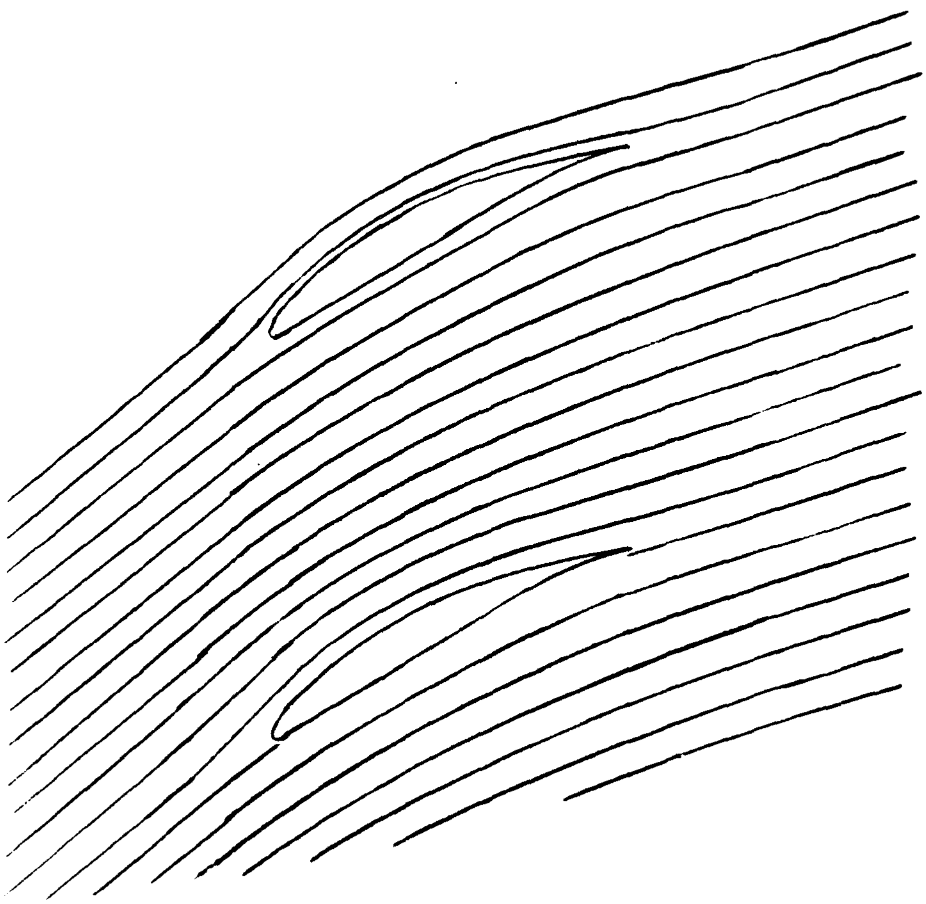


FIGURE 20 FLOW PATTERN

Section 4
ORTHOGONAL GRIDS

It was pointed out above that many current flowfield codes require a grid, in the computational plane, having the property that the images of the trailing edge and the point at downstream infinity lie at corners of the grid. This property is not realized by the Ives transformation; the point at downstream infinity occurs at a corner of the $\hat{\xi}$ -plane, but the trailing edge does not. The sheared grid discussed in Section 2 is one way to achieve the desired property, but it has the disadvantage that the resulting grid is not orthogonal.

The Schwarz-Christoffel transformation can be used, in place of the shearing, to achieve an orthogonal grid. A number of recent publications[22-25] have discussed the use of this transformation; its great versatility offers several possible ways to use it, within the Ives Transformation. For example, it would be possible to replace the elliptic-integral step between the η and $\hat{\xi}$ planes by a hyper-elliptic integral (which is a form of the Schwarz-Christoffel formula). However, a much more straightforward application has been made here, namely to map the parallelogram formed by joining the trailing-edge and downstream infinity points in the $\hat{\xi}$ -plane into a rectangle. This step can be used on other grid generators having similar properties, such as that of Ref. 26, for example.

Figure 21 shows the parallelogram formed by joining the images of the trailing edge and the point at downstream infinity in the $\hat{\xi}$ plane:

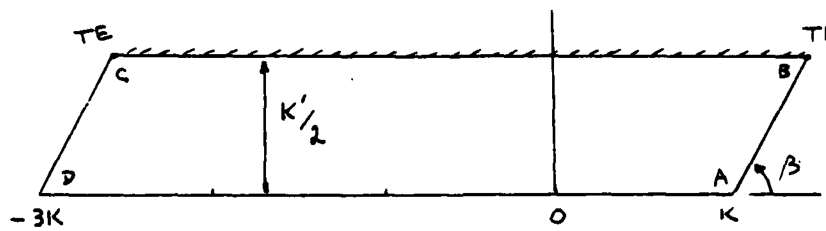


FIGURE 21 PARALLELOGRAM IN THE $\hat{\xi}$ -PLANE

The particular values of the coordinates shown on this figure are those peculiar to the Ives transformation; the method applies to arbitrary locations.

This figure is mapped into the upper half of the t -plane by the Schwarz-Christoffel transformation:

$$\frac{d\hat{\xi}}{dt} = M\,(t-t_A)^{-\beta/\pi}\,(t-t_B)^{-1+\beta/\pi}\,(t-t_C)^{-\beta/\pi}\,(t-t_D)^{-1+\beta/\pi}$$

where the parameters t_A through t_D are pure real:

The exponents in the above equation guarantee that an integration path along the real axis in the t -plane will produce a four-sided figure with the proper angles. The four parameters t_A through t_D and the complex constant M must be chosen so as to produce the desired parallelogram.

It turns out that only two parameters are needed: t_A can be set equal to $-t_D$, t_B to $-t_C$, and t_C can be set equal to 1.0 without loss of generality:

$$\frac{d\hat{\xi}}{dt} = M\,(t+t_D)^{-\beta/\pi}\,(t+1)^{-1+\beta/\pi}\,(t-1)^{-\beta/\pi}\,(t-t_D)^{-1+\beta/\pi}$$

This set of parameters produces a genuine parallelogram, i.e., the lengths of opposite sides are equal. Thus, all that is needed is to select t_D so as to give the desired ratio of the adjacent sides, and then set the parameter M so as to give the proper absolute size.

Numerical integrations of this formula can be carried out using the method of Ref. 22; the integral over a distance Δt can be written as:

$$\hat{\xi}_{m+1} - \hat{\xi}_m = \frac{M}{(\Delta t)^3} \prod_{i=1}^{4} \left\{ \left. \frac{(t-t_i)^{P_i}}{P_i} \right|_{t_m}^{t_{m+1}} \right\}$$

where

$$\Delta t = t_{m+1} - t_m$$

and where t_i and P_i are the parameters:

$$P_1 = P_3 = 1 - \beta/\pi \quad ; \quad P_2 = P_4 = \beta/\pi$$

$$t_4 = -t_1 = t_D \quad ; \quad t_3 = -t_2 = 1$$

The length/width ratios of the parallelograms produced by typical cascades are generally 5.0 or larger. Numerical work using the formula above shows that such ratios require values of t_D that are only slightly larger than 1:

$$t_D = 1 + \epsilon$$

Asymptotic formulas for the lengths of the slant side (called side 1) and the horizontal side (called side 2) of the parallelogram can be found by taking the limit as $\epsilon \to 0$:

$$\frac{1}{M} \Delta \hat{\xi} \Big)_{SIDE\,1} = \int_{-1-\epsilon}^{-1} (t+1+\epsilon)^{-\beta/\pi} (t+1)^{-1+\beta/\pi} (t-1)^{-\beta/\pi} (t-1-\epsilon)^{-1+\beta/\pi} dt$$

The change of variable $t = -1-\epsilon x$ leads to:

$$\frac{1}{M} \Delta \hat{\xi} \Big)_{SIDE\,1} = \frac{1}{2} e^{i\beta} \int_0^1 (1-x)^{-\beta/\pi} x^{-1+\beta/\pi} F(x;\epsilon)\, dx$$

where

$$F = \left(1 + \frac{\epsilon x}{2}\right)^{-\beta/\pi} \left(1 + \frac{\epsilon}{2}[x+1]\right)^{-1+\beta/\pi}$$

$$= 1 - \frac{\epsilon}{2}\left(\frac{\beta}{\pi} x + \left[1 - \frac{\beta}{\pi}\right](x+1)\right) + O(\epsilon^2)$$

The leading term can be expressed in terms of gamma functions; using the relations (Ref 27)

$$\int_0^1 t^{x-1} (1-t)^{y-1} dt = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$$

$$\Gamma(z+1) = z\Gamma(z)$$

$$\Gamma(z)\Gamma(1-z) = \pi \csc(\pi z)$$

leads to

$$\frac{1}{M} \Delta \hat{\xi} \Big)_{SIDE\,1} = \frac{1}{2} e^{i\beta} \frac{\pi}{\sin\beta} \left[1 + O(\epsilon)\right]$$

To find the analogous result for side 2:

$$\frac{1}{M}\Delta\hat{\xi}\Big)_{\text{SIDE }2} = \int_{-1}^{+1} (t+1+\epsilon)^{-\beta/\pi} (t+1)^{-1+\beta/\pi} (t-1)^{-\beta/\pi} (t-1-\epsilon)^{-1+\beta/\pi} dt$$

Let

$$\int_{-1}^{+1} \cdots = \int_{-1}^{0} \cdots + \int_{0}^{+1} \cdots$$

In the first of these integrals, let $t+1 = \epsilon x$, and in the second, let $t-1 = -\epsilon y$; this leads to

$$-\frac{2}{M}\Delta\hat{\xi}\Big)_{\text{SIDE }2} = \int_{0}^{1/\epsilon} (x+1)^{-\beta/\pi} x^{-1+\beta/\pi} \left[1+O(\epsilon)\right] dx$$

$$+ \int_{0}^{1/\epsilon} (y+1)^{-1+\beta/\pi} y^{-\beta/\pi} \left[1+O(\epsilon)\right] dy$$

The first of these integrals can be estimated by an integration by parts, while the second can be approximated, for $\epsilon = 0$, as a Mellin transform; the leading term in each integral is $\ln\epsilon$, with the result:

$$\frac{1}{M}\Delta\hat{\xi}\Big)_{\text{SIDE }2} = -\ln\left(\frac{1}{\epsilon}\right) + O(1)$$

The practical application of these asymptotic formulas is to provide a first guess at ϵ for a given ratio of the length of the adjacent sides of the parallelogram. Values of ϵ on the order of 10^{-5} are needed, for side ratios of 5 or so; these small values required comparably small step sizes in the numerical evaluations, in order to preserve accuracy.

Once the parameters for mapping of the parallelogram are established, the next step is make the upper half of the t-plane into a rectangle; this is achieved by using the same values of ϵ and M, with $\beta/\pi = 1/2$.

An orthogonal grid can then be set up in the rectangle plane, and mapped (numerically) to the t-plane. This grid can then be mapped, again numerically, from the t-plane to the parallelogram in the $\hat{\xi}$-plane, and then back to the physical plane, as explained in Section 2.

Section 5
METRIC EVALUATIONS

The coordinate transformation enters the flowfield solution algorithm only in the metric derivatives. These can be evaluated by differencing the coordinates themselves, or in the case of a conformal transformation, by evaluating the analytic expressions for them. These analytic expressions are derived in Ref. 1, and the code listed in that report contains the Fortran statements required to evaluate these expressions. However, as pointed out in Ref. 7, the truncation error resulting from the use of analytic metrics in the finite-difference flowfield code is large enough to cause major instabilities in the solution algorithm. It is highly preferable to use metrics that are found by differencing the coordinates in the same manner as the flowfield variables are differenced. A program to achieve this, for the grid conventions used in Ref. 7, is given in Appendix D.

Section 6
COMPUTER PROGRAM

A listing of the computer program is given in Appendix B, and a dictionary of variables is given in Appendix C. This section contains a general description of the program, plus some specific details.

In order to handle complex arithmetic, all variables beginning with the letter Z are declared to be complex by an implicit type specification at the beginning of the program.

The input is generally described by comment cards in the deck. Certain blade-shape parameters must be read in: EX, G, H, ZLE, ZTE, ZN, ZT. Also, if IGOT=1, ZC must be read in. The blade shape itself is defined by pairs of coordinates in the s, n plane - KJS on the suction side and KJP on the pressure side. In the version shown here, these were read in as a table in subroutine SHAPE.

The blade surface coordinates are numbered from 1 to KJMX; point number 1 is the trailing edge, 2 through KJLM are on the pressure side from trailing edge to leading edge, point KJLE is the leading-edge point (ZLE), KJLP through KJMXM are on the suction side from leading edge to trailing edge, and point KJMX repeats the trailing edge.

The complex sine functions are calculated next; then the iterations to determine the parameter C are done. The initial guess provided for C is $ZC = 1+i$. It may happen in some cases that a better guess is required: in particular, it is necessary that the value of C must lie outside the blade-surface curve in the Ω-plane. If it does not, then the interior of this curve in the Ω-plane is mapped to the exterior of the blade-surface image in the ω-plane. This fact can be seen from the discussion by Kober (Ref. 28 of the bilinear transformation applied to circles.

---

28. Kober, H., Dictionary of Conformal Transformations, Dover Publications, New York (1957).

After the parameter c has been found, the transformation to the ζ plane is carried out, using the fast Fourier transfrorm procedure (see Appendix A). The actual calculations of the Fourier coefficients are done in subroutine FFT2, a proprietary program of International Mathematical and Statistical Libraries, Inc. (IMSL). This routine computes the fast Fourier transform of a complex vector of length equal to a power of two (here 2^6). The coefficients of the input vector are given in normal order by the array named as the first argument of the call; the coefficients of the output vector are overstored in this array, in reverse binary order. The subroutine SHUFL is then used to restore this output to the normal order. The coefficients in the series expression for ζ are determined iteratively in a relaxation process that is terminated when the maximum change in θ falls below the tolerance ANGERR, or when IMX iterations are done. If the derivative form of the transformation is being used, the maximum change in S is required to be less than ANGERR.

In doing these iterations, it is necessary to know values of $\ln r$ at given values of θ; these are found by a spline fit in subroutine CISPLN, which is a straightforward implementation of the formulas given by Ahlberg, et al.[29] Similarly, β at given values of s is found by a spline fit.

In certain cases (typically when RMAX/RMIN is large) the calculated variation of θ with ϕ may be non-monotonic; if this occurs, the calculation should be repeated, with a smaller relaxation factor. The progress of the ϕ/θ iterations is printed, showing at each iteration the largest change in θ and the number of reversals (i.e., the number of occurrences of non-monotonic variation). The iterations using the derivative form usually do not display a similar problem, and converge very quickly, with OM = 1.

---

29. Ahlberg, J.H., Nilson, E.N., and Walsh, J.L., The Theory of Splines and Their Applications Academic Press, New York (1967).

Next, the parameters ζ_A and ζ_B are found, starting with "best-guess" values calculated in the sectors described in Section 2. Once these are found, the mapping to the η -plane follows. The calculations that link the Ω -plane and the ζ -plane are done in subroutine OMETA, which sums the Theodorsen-Garrick series, using complex arithmetic. This subroutine has been modified slightly from that appearing in Ref. 9, as follows: the previous code evaluated sums of the form

$$ZSUM = \sum_{JP=1}^{65} ZCC(JP)\ \zeta^{JP-1} \tag{4-1}$$

by a sequence of multiplications and additions:

$$ZSUM = ZCC(65)\ \zeta^{64} + ZCC(64)\ \zeta^{63} + \cdots + ZCC(1) \tag{4-2}$$

The current version uses (Ref. 30, p. 28)

$$ZSUM = \left\langle \left\{ \left[ZCC(65)\zeta + ZCC(64) \right] \zeta + ZCC(63) \right\} \zeta + \cdots \right. \tag{4-3}$$

Finally, the blade-surface image is mapped into the $\hat{\xi}$ plane, using the elliptic-function formulas of Nielsen and Perkins[16] and of Luke,[18] as outlined in Section 2. This completes the mapping of the blade surface.

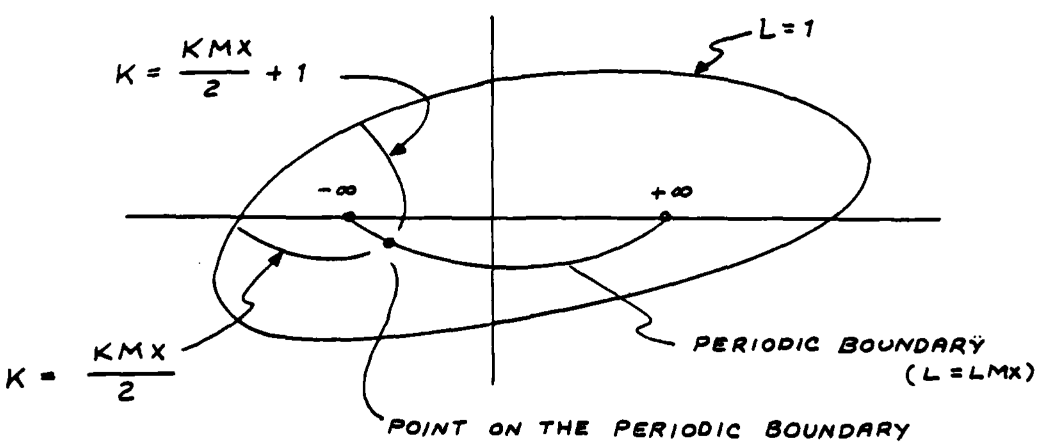
It is now possible to set up a grid in the $\hat{\xi}$ - plane, and map it back into the physical plane. This involves straightforward evaluations of the transformation functions. The only complication is the need to evaulate the Jacobian elliptic sine (done in subroutine JCELFN).

30. Hartree, D.R., Numerical Analysis, 2nd Edition, Oxford, Clarendon Press (1958).

The calculation of the images of the grid points in the various planes is bypassed for the points at upstream and downstream infinity, and at the trailing edge. (The trailing-edge point will be a grid point if ISHEAR=1). The z -plane locations of upstream and downstream infinity are arbitrarily assigned to finite locations given by linear extrapolation from the two adjacent L -values.

The point at upstream infinity will be a grid point only if KMX is odd; in this case IOE=1, and the image calculations are bypassed.

Adjustments to the grid-point image locations in the z -plane are sometimes required for points on the periodic ouundary (L=LMX) for values of K near 1, KMX/2, and KMX. At these points, the formula used in going from the Ω -plane to the z -plane sometimes cannot distinguish between points that are separated by $H + i\,G$. The problem can be seen best in the ω -plane. (The sketch below is for an even value of KMX; the same picture applies for an odd value):



Note that the same point in the ω-plane (and thus also in the Ω-plane) can map into either of two points in the Z-plane, which differ by $H + iG$. The selection is guided toward the correct value by starting on the blade (L=1) and working toward the periodic boundary (L=LMX), but it can happen that the wrong branch is chosen during the iterations. To avoid this, the imaginary parts of Z and ZN are compared, for K values near the leading edge, and the imaginary parts of Z and ZT are compared near the trailing edge, and the quantity $i \cdot SG$ is added or subtracted (depending on the value of K) where necessary.

Finally, the real and imaginary parts of Z are written to unit 7 (if PNCHZA=TRUE). These values can be used, in a separate program, to calculate the metrics of the transformation (see Appendix D).

Section 7
CONCLUDING REMARKS

The method described above is capable of generating grids, and the associated incompressible, inviscid flow-field, for blade rows with gap/chord ratios as low as 0.8. Below that value, the Theodorsen-Garrick mapping may not converge. In such cases, a derivative form of the transformation usually will converge, but the grid that results may not be useful, in the region near the leading edge. Further development of this part of the technique is required.

As a separate mapping step, the Schwarz-Christoffel technique has been applied to the problem of generating orthogonal grids. Based on the analysis given above, several grid-generated techniques can now be generalized to give orthogonal grids.

APPENDIX A

DETAILS OF THE FAST FOURIER TRANSFORM PROCEDURES

The details given in this Appendix apply specifically to the Theodorsen-Garrick mapping; with obvious changes in notation, they also are valid for the derivative form of the transformation.

Equations 2-20 and 2-21 contain $2N+2$ constants, which are evaluated as follows: first, they are satisfied at a discrete number of points, denoted by ϕ_K :

$$\phi_K = \frac{2\pi(K-1)}{2N} \quad , \quad K = 1, 2, \ldots, 2N \qquad \text{(A-1)}$$

where N was chosen to be 64, in the present case. Thus:

$$\ln r_K = A_0 + \sum_{j=1}^{N-1} (A_j \cos j\phi_K - B_j \sin j\phi_K) + (-1)^K A_N \qquad \text{(A-2)}$$

$$\theta_K - \phi_K = B_0 + \sum_{j=1}^{N-1} (B_j \cos j\phi_K + A_j \sin j\phi_K) + (-1)^K B_N \qquad \text{(A-3)}$$

Each right-hand side now contains $2N$ coefficients. The correspondence between either of these and the Fast Fourier Transform (FFT) as presented in Ref. 12 is given by the next two equations: consider the expression

$$Y(j) = \sum_{n=0}^{2N-1} C(n)\, W_{2N}^{nj} \quad ; \quad j = 0, 1, 2, \ldots, 2N-1 \, ; \quad W_{2N} = e^{i\frac{2\pi}{2N}} \qquad \text{(A-4)}$$

where values $Y(\cdot)$ are real, and the $2N$ values of $C(\cdot)$ are in general complex, but must satisfy the following redundancy condition, in order that the $Y(\cdot)$ values be real:

$$C(n) = \tilde{C}(2N-n) \quad , \quad n = 1, 2, \ldots, N-1 \qquad \text{(A-5)}$$

$C(0)$ and $C(N)$ pure real

where the tilde denotes the complex conjugate. When these conditions are met, Eq. A-4 can be written as

$$Y(\ell) = C_R(0) + (-1)^{\ell} C_R(N) + 2\sum_{n=1}^{N-1} \left\{ C_R(n) \cos\frac{n\ell\pi}{N} - C_I(n) \sin\frac{n\ell\pi}{N} \right\}$$

$$\ell = 0, 1, \ldots, 2N-1 \qquad \text{(A-6)}$$

This form can now be used, in conjunction with Eq. 2-20 or 2-21, to facilitate application of the FFT to the complex form given in Eq. A-4.

In the case of Eq. A-2, values of the coefficients A_0 through A_N and B_1 through B_{N-1} are found, from given values of $\ln r$. This is procedure 4 of Ref. 6, which takes the following steps:

1. Set
$$\left.\begin{aligned} X_1(k) &= Y(2k) = (\ln r)_{2k} \\ X_2(k) &= Y(2k+1) = (\ln r)_{2k+1} \end{aligned}\right\} k = 0, 1, \ldots, N-1 \qquad \text{(A-7)}$$

2. Set
$$X(j) = X_1(j) + i X_2(j) \;, \quad j = 0, 1, \ldots, N-1 \qquad \text{(A-8)}$$

3. Calculate the N-point Discrete Fourier Transform of

$$A(n) = A_1(n) + iA_2(n) = \frac{1}{N} \sum_{j=0}^{N-1} X(j)\, W_N^{-nj} \;; \quad n = 0, 1, \ldots, N-1$$
$$W_N = e^{i \frac{2\pi}{N}} \qquad \text{(A-9)}$$

4. By periodicity, set $A(N) = A(0)$

5. Apply Eq. 34 of Ref. 12, in order to extract $A_1(n)$ and $A_2(n)$ from $A(n)$:

$$\left.\begin{aligned} A_1(n) &= \frac{1}{2}\left\{\tilde{A}(N-n) + A(n)\right\} \\ A_2(n) &= \frac{i}{2}\left\{\tilde{A}(N-1) - A(n)\right\} \end{aligned}\right\} n = 0, 1, \ldots, \frac{N}{2} \qquad \text{(A-10)}$$

Note that these expressions use $A(n)$ for $n = 0, 1, \ldots, N$ to give $A_1(n)$ and $A_2(n)$ for $n = 0, 1, \ldots, N/2$

6. These values of $A_1(n)$ and $A_2(n)$ then give $C(n)$ for the same range:

$$C(n) = \frac{1}{2}\left[A_1(n) + W_{2N}^{-n} A_2(n)\right], \quad n = 0, 1, \ldots, N/2 \qquad \text{(A-11)}$$

For the range of n from $\frac{N}{2}+1$ to $N-1$, use Eq. 36 of Ref. 12, with n replaced by $N-n$ (and noting that $W_{2N}^{-N} = -1$):

$$\begin{aligned} C(n) &= \tilde{C}(2N-n) \\ &= \frac{1}{2}\left\{A_1(N-n) + W_{2N}^{n} A_2(N-n)\right\} \quad n = \frac{N}{2}+1, \frac{N}{2}+2, \ldots, N-1 \end{aligned} \qquad \text{(A-12)}$$

This equation, applied for $n = \frac{N}{2}+1, \frac{N}{2}+2, \ldots, N-1$ uses A_1 and A_2 with index $\frac{N}{2}-1, \frac{N}{2}-2, \ldots, 1$ to get $C(n)$ for $n = \frac{N}{2}+1, \frac{N}{2}+2, \ldots, N-1$. The process is completed by setting

$$C(N) = \frac{1}{2}\left\{A_1(0) - A_2(0)\right\} \tag{A-13}$$

7. The A_j and B_j coefficients are retrieved from:

$$\begin{aligned} A_0 &= \mathrm{Re}\left[C(0)\right], \quad A_N = \mathrm{Re}\left[C(N)\right] \\ A_j &= 2\,\mathrm{Re}\left[C(j)\right], \quad B_j = 2\,\mathrm{Im}\left[C(j)\right], \quad j = 1, 2, \ldots, N-1 \end{aligned} \tag{A-14}$$

At this point, B_0 and B_N are undetermined. Following Ives, B_N is set equal to zero, and B_0 is chosen so as to place the trailing edge at $\phi = 0$ (this latter selection of B_0 is actually carried out in a subsequent step, noted below).

In the case of Eq. A-3, the A's and B's are considered known, and are used to evaluate Θ_k. The coefficient B_0 can be found from

$$\Theta_{TE} = B_0 + \sum_{j=1}^{N-1} B_j \quad (B_N = 0) \tag{A-15}$$

Actually, it is simpler to evaluate the right-hand side of

$$\Theta_k - \phi_k - B_0 = \sum_{j=1}^{N-1} (B_j \cos j\phi_k + A_j \sin j\phi_k), \quad k = 0, 1, \ldots, N-1 \tag{A-16}$$

and then find B_0 from

$$B_0 = \Theta_0 - RHS)_{k=0} \tag{A-17}$$

The actual evaluation of the right-hand side takes the following steps (Procedure 5 of Ref. 12): by comparison of Eqs. A-3 and A-6:

1. Set $C(0) = 0, \quad C(N) = 0$

$$C(n) = \frac{1}{2}\left[B_n - i A_n\right], \quad n = 1, 2, \ldots, N-1 \tag{A-18}$$

Note that the C's determined here are different from those used in Procedure 4; the A_j and B_j values are the same, but their relation to C_j is different.

2. Values of C_j are then used to find $A_1(n)$ and $A_2(n)$: Equations 40 and 41 of Ref.12 are rewritten, using

$$C(n) = \tilde{c}(2N-n)$$

Replace n by $N+n$:

$$C(N+n) = \tilde{c}(2N-N-n) = \tilde{c}(N-n)$$

Thus Eqs. 40 and 41 of Ref. 12 are

$$\left.\begin{aligned} A_1(n) &= C(n) + \tilde{c}(N-n) \\ A_2(n) &= \left[C(n) - \tilde{c}(N-n)\right] W_{2N}^{n} \end{aligned}\right\} n = 0, 1, \ldots, N/2 \qquad \text{(A-19)}$$

These are all the values needed for A_1 and A_2.

3. Find $A(n)$, $n=0,1,\ldots,N-1$ from Eqs. 42 and 43 of Ref. 12:

$$\left.\begin{aligned} A(n) &= A_1(n) + iA_2(n) \\ A(N-n) &= \tilde{A}_1(n) + i\tilde{A}_2(n) \end{aligned}\right\} n = 0, 1, \ldots, N/2 \qquad \text{(A-20)}$$

This gives, on the left-hand side, all values from $n=0$ to $n=N$.

4. Calculate

$$X(j) = \sum_{n=0}^{N-1} A(n)\, W_N^{nj}, \quad j = 0, 1, \ldots, N-1 \qquad \text{(A-21)}$$

5. Finally:

$$\left.\begin{aligned} \theta_{2k} - \phi_{2k} - B_0 &= \mathrm{Re}\left[X(k)\right] \\ \theta_{2k+1} - \phi_{2k+1} - B_0 &= \mathrm{Im}\left[X(k)\right] \end{aligned}\right\} k = 0, 1, \ldots, N-1 \qquad \text{(A-22)}$$

The first of these equations, with $k=0$, is used to find B_0.

In order to apply these formulas, it is necessary to have a relation between r and θ:

$$r_K = r(\theta_K) \;, \quad K = 1, 2, \ldots, 2N \tag{A-23}$$

which is found from a spline fit to the blade-surface image in the ω- plane.

The discrete Fourier transform and its inverse are given by

$$A(n) = \frac{1}{N} \sum_{j=0}^{N-1} X(j)\, W_N^{-nj} \;, \qquad n = 0, 1, 2, \ldots, N-1 \tag{A-24}$$

$$X(j) = \sum_{n=0}^{N-1} A(n)\, W_N^{nj} \;, \qquad j = 0, 1, 2, \ldots, N-1 \tag{A-25}$$

The IMSL routine FFT2 evaluates the second of these, i.e., it returns $X(j)$, given the values $A(n)$. To evaluate the first of these, FFT2 is used with input $\tilde{X}(j)/N$, and with output interpreted to be $\tilde{A}(n)$; this is Procedure 1 of Reference 6:

$$N\tilde{A}(n) = \sum_{j=0}^{N-1} \tilde{X}(j)\, W_N^{nj} \tag{A-26}$$

In the FORTRAN version of these and other procedures, it is convenient to use indices that begin at one, rather than zero, by setting $j+1 = j'$ (FORTRAN symbol JP). The corresponding table, for example of the coefficient B_j, is

j	JP	B_j	B(JP)
0	1	B_o	B(1)
1	2	B_1	B(2)
2	3	B_2	B(3)
N-1	N	B_{N-1}	B(N)
N	N+1	B_N	B(N+1)

In addition, care must be taken with the argument $N-n$; for example, Eqs. are written as

$$\left.\begin{aligned} ZA1(NP) &= ZCC(NP) + \widetilde{ZCC}(N+2-NP) \\ ZA2(NP) &= \left[ZCC(NP) - \widetilde{ZCC}(N+2-NP)\right] W_{2N}^{n} \end{aligned}\right\} \quad NP = 1,2,\ldots,\frac{N}{2}+1 \tag{A-27}$$

It can be verified that the quantity N + 2 - NP in the arguments above preserves the correct ordering - for example when N = 64, and n = 0, $\tilde{C}(N-n) = \tilde{C}(64)$ This would be stored in ZCC(64 + 2 - 1) = ZCC(65).

Appendix B

COMPUTER PROGRAM LISTING

```
      PROGRAM RAVES(INPUT,OUTPUT,GRID,TAPE5=INPUT,TAPE6=OUTPUT,
     *   TAPE7=GRID)
C
C  PROGRAM RAVES:  THE IVES-LIUTERMOZA CONFORMAL TRANSFORMATION FOR
C  TURBOMACHINERY CASCADES (AIAA JOURNAL,VOL. 5,1977, PP 647 - 652)
C  DOCUMENTATION IS GIVEN IN:
C       W. J. RAE, A COMPUTER PROGRAM FOR THE IVES TRANSFORMATION
C       IN TURBOMACHINERY CASCADES, CALSPAN CORPORATION REPORT 6275-A-3,
C       NOVEMBER 1980
C       W.J. RAE, MODIFICATIONS OF THE IVES - LIUTERMOZA CONFORMAL-
C       MAPPING PROCEDURE FOR TURBOMACHINERY CASCADES, ASME PAPER
C       83-GT-116, MARCH 1983
C       W.J. RAE, REVISED COMPUTER PROGRAM FOR EVALUATING THE IVES
C       TRANSFORMATION IN TURBOMACHINERY CASCADES, CALSPAN CORPORATION
C       REPORT 7177-A-1, JULY 1983
C       W.J. RAE AND P.V.MARRONE, RESEARCH ON TURBINE FLOWFIELD
C       ANALYSIS METHODS, CALSPAN CORPORATION REPORT 7177-A-3,
C       JANUARY 1985
C
      IMPLICIT REAL(A-H,O-Y),COMPLEX(Z)
      LOGICAL PNCHZA
      COMMON/TGINTG/N,NP1,NP2,N2,NB2,NB2P1,IP,IPMX,ITP,IWK,IMX,KJMX
      COMMON/TGCMPX/Z1,ZW2N,ZI,ZNN,ZA,ZCC,ZA1,ZA2
      COMMON/TGDBLE/PBN,OM,OMM,ANGERR,A,B,E,F,THT,PHI,X,Y,BETA,S,DR,DI
      DIMENSION ZS(80),ZP(80),ZOMS(80),ZOMP(80)
      DIMENSION RDS(80),RDP(80),THS(80),THP(80)
      DIMENSION RDSX(80),RDPX(80),THSX(80),THPX(80)
      DIMENSION X(160),Y(160),E(160),F(160),THT(160),
     * BETA(160),S(160)
      DIMENSION PHI(130),A(65),B(65),ZCC(65),DR(65),DI(65),
     *   ZA(165),ZA1(165),ZA2(165),
     *   ZEE(165),ZFF(165),ZSOMGA(165),
     *   ZXI(10,40),ZEETA(10,40),ZETA(10,40),ZOMA(10,40),
     *   ZFNL(10,40),ZXY(10,40),ZVEL(10,40),ZFNP(10,40),
     *          IWK(7)
      DIMENSION ITP(100),ID(36)
      DIMENSION XX(50,20),YY(50,20)
      DO 10 I = 1,7
   10 IWK(I) = 0
C
      NAMELIST/INPUTS/ ANGERR,EX,G,H,IGOT,ILE,ITE,KMX,ALP,
     *            IMX, LMX,  OM, PNCHZA, RTOL, ZC, ZLE, ZN, ZT, ZTE,
     *            IPMX,ISHEAR,KJS,KJP,ICIRC,NS
C
C --- SET NAMELIST DEFAULT VALUES
```

```
C
      ICIRC=1
      ANGERR=0.024
      ALP = 0.
      IGOT=0
      ILE = 1
      ITE=0
      KMX = 40
      LMX = 10
      KJS = 20
      KJP = 20
      IMX = 400
      ISHEAR = 1
      IPMX = 1
      NS = 40
      PNCHZA=.FALSE.
      RTOL = 3.
      OM = 1.
C
C     NOTE: EX, G, H, ZC, ZLE, ZN, ZT, ZTE HAVE NO DEFAULT VALUES
C     (ZC IS NOT NEEDED IF IGOT=0)
C
      READ(5,103)(ID(I),I=1,36)
  103 FORMAT(18A4)
C
      PI = 4.0  *ATAN(1.0  )
      TPI = 2.0  *PI
      PB2 = PI/2.
      ZPI = CMPLX(PI,0.0  )
      Z1 = CMPLX(1.0  ,0.0)
      ZERO = CMPLX(0.0  ,0.0)
      ZMGA = CMPLX(+1.0  ,0.0)
      ZMGB = CMPLX(-1.0  ,0.0)
      ZEPS = CMPLX(0.,1.E-08)
      ZBG = CMPLX(1.E10,1.E10)
C  READ PNCHZA=T IF (ZA(L),L=1,LMX) IS TO BE PUNCHED FOR ALL VALUES
C  OF K, OTHERWISE READ PNCHZA=F
C  IGOT = 1 IF THE VALUE OF ZC IS KNOWN.  OTHERWISE, IGOT = 0.
C  KMX  AND LMX  ARE THE GRID SIZES IN THE FINAL TRANSFORMED PLANE.
C  KJS AND KJP ARE THE NUMBERS OF POINTS ON THE SUCTION AND PRESSURE
C  SIDES AT WHICH PAIRS OF BLADE COORDINATES WILL BE INPUT.
C  ILE = 0 OR 1 FOR A SHARP OR ROUNDED LEADING EDGE, RESPECTIVELY
C  ITE = 0 OR 1 FOR A SHARP OR ROUNDED TRAILING EDGE, RESPECTIVELY.
C  FOR A SHARP LEADING EDGE (ILE=0) ZN MUST EQUAL ZLE
C  FOR A SHARP TRAILING EDGE (ITE=0) ZT MUST EQUAL ZTE
C  IMX IS THE MAXIMUM NUMBER OF ITERATIONS ALLOWED FOR THE PHI/THETA(/S)
C  ITERATIONS
C  IPMX.NE.1    CAN BE USED TO DISPLAY THE VALUES OF THETA/S DURING THE
C  PHI/THETA(/S) ITERATIONS, AT THE ITERATION NUMBERS READ INTO THE
C  ITP ARRAY BELOW.
C  ISHEAR = 0 GIVES AN ORTHOGONAL GRID.  THE LINES K=1 AND K=KMX, WHICH
C  START AT THE TRAILING EDGE, DO NOT GO TO DOWNSTREAM INFINITY.
C  ISHEAR = 1 SHEARS THE GRID UNIFORMLY; IN THIS CASE, THE K=1 AND KMX
C  LINES DO GO TO DOWNSTREAM INFINITY.
```

```
C  OM IS A RELAXATION FACTOR USED IN THE PHI/THETA MAPPING:
C   - FOR ICIRC = 0, USE 0.1, OR A SMALLER VALUE IF THE A AND B
C      ITERATIONS FAIL TO CONVERGE
C   - FOR ICIRC = 1, OM = 1.0 SHOULD BE SATISFACTORY.
C  ANGERR IS THE ANGULAR(IN RADIANS)(/ARC LENGTH) TOLERANCE
C  FOR THE PHI/THETA(/S) TRANSFORMATION.
C  A REASONABLE VALUE IS .01
C  RTOL IS THE TOLERANCE FOR THE MAX/MIN RADIUS RATIO IN THE
C  OMEGA PLANE
C  ALP IS THE ANGLE OF INCIDENCE, IN DEGREES (NEGATIVE), IN
C  THE ETA-PLANE.
C  ICIRC = 0 FOR THE THEODORSEN-GARRICK TECHNIQUE
C  ICIRC = 1 FOR THE BAUER ET AL TECHNIQUE
C  NS IS THE NUMBER OF STEPS TO BE USED IN THE SIMPSON'S-RULE
C  INTEGRATION OF D(SMALL OMEGA)/D(ZETA) WHEN ICIRC = 1.
C
C --- READ NAMELIST INPUT DATA
C
      READ(5,INPUTS)
      ITP(1) = 999
      IF(IPMX.NE.1) READ(5,102)(ITP(IP),IP=1,IPMX)
  102 FORMAT(20I4)
      IP = 1
C
C
      KMXH=(KMX+1)/2
      KMXL=KMXH
      IF(MOD(KMX,2).NE.0) KMXL=KMXH-1
      IOE = MOD(KMX,2)
      KOUNT = 0
C
      DO 310 K=1,50
      DO 311 L=1,20
      XX(K,L) = 0.
      YY(K,L) = 0.
  311 CONTINUE
  310 CONTINUE
C
C  INPUT VARIABLES EX, G, H, ZLE, ZN, ZT, ZTE,
C  THE FORTRAN STATEMENTS IN SUBROUTINE SHAPE,
C  AND THE VARIABLES LISTED IN COMMON BLOCK GEOM (IF ONE IS BEING USED)
C  ARE ALL SPECIFIC TO THE BLADE SHAPE BEING USED.
C
C
C  CALCULATION OF THE BLADE SHAPE
C
      D = CABS(ZTE-ZLE)
      CALL SHAPE(D,H,G,EX,ZP,ZS,KJS,KJP)
      SG = SQRT(H*H+G*G)
      ZDA = CMPLX(G/SG,-H/SG)
      ZGAMMA=CONJG(ZDA)
      ALP = ALP*TPI/360.
      ZALP = CMPLX(COS(ALP),SIN(ALP))
      ZAM =  CONJG(ZALP)
      ZAP =  ZALP
      WRITE(6,207)
  207 FORMAT(1H1)
```

```
      WRITE(6,206)(ID(I),I=1,36)
  206 FORMAT(30X,18A4)
      WRITE(6,240) D,H,G,EX,SG
  240 FORMAT(//10X,'BLADE-GEOMETRY PARAMETERS ARE:',
     *   //5X,'ABS(ZTE-ZLE) =',
     *                  F10.5,'  H = ',F10.5,'  G = ',F10.5,'  EX = ',
     *   F10.5,'  SLANT GAP =',F10.5,//)
      WRITE(6,INPUTS)
      KJLE = KJP + 2
      KJMX = KJLE + KJS + 1
      WRITE(6,209)
  209 FORMAT( 8X,'BLADE COORDINATES:',
     *   //9X,'SUCTION SIDE',/3X,'KJ',7X,'S',13X,'N',13X,'X',13X,'Y',/)
      ZZXY = ZGAMMA*ZLE
      WRITE(6,270) KJLE,ZLE,ZZXY
  270 FORMAT(I5,1P4E14.5)
      DO 64 K = 1,KJS
      KJ = KJLE + K
      ZZXY = ZGAMMA*ZS(K)
      WRITE(6,270) KJ,ZS(K),ZZXY
   64 CONTINUE
      ZZXY = ZGAMMA*ZTE
      WRITE(6,270) KJMX,ZTE ,ZZXY
      WRITE(6,271)
  271 FORMAT(//8X,'PRESSURE SIDE',/3X,'KJ',7X,'S',13X,'N',
     *   13X,'X',13X,'Y',/)
      ZZXY = ZGAMMA*ZLE
      WRITE(6,270) KJLE,ZLE,ZZXY
      DO 61 K = 1,KJP
      KJ = KJLE - K
      ZZXY = ZGAMMA*ZP(K)
      WRITE(6,270) KJ,ZP(K),ZZXY
   61 CONTINUE
      KJ = 1
      ZZXY = ZGAMMA*ZTE
      WRITE(6,270) KJ,ZTE,ZZXY
C
      ZDN = CMPLX(H,G)
      ZZ = (ZT-ZN)/ZDN
      CHI = PI*EX*(G*REAL(ZT-ZN)-H*AIMAG(ZT-ZN))/(SG*SG)
      XA = PI*EX*(H*REAL(ZT-ZN)+G*AIMAG(ZT-ZN))/(SG*SG)
      R = EXP(-CHI)
      ZPLUS = CMPLX(R*COS(XA),-R*SIN(XA))
      R = 1.0/R
      ZMINUS= CMPLX(R*COS(XA),R*SIN(XA))
      DO 20 K = 1,KJS
      ZZT = (ZS(K)-ZT)/ZDN
      ZETA1S = ZPI*ZZT
      ZETA2S = CSIN(ZETA1S)
      ZZN = (ZS(K)-ZN)/ZDN
      ZETA3S = ZPI*ZZN
      ZETA4S = CSIN(ZETA3S)
      ZFS = ZETA2S/ZETA4S
```

```
      RDS(K) = CABS(ZFS)
      THS(K)= ATAN2(AIMAG(ZFS),REAL(ZFS))
   20 CONTINUE
      DO 24 K = 1,KJP
      ZZT = (ZP(K)-ZT)/ZDN
      ZETA1P = ZPI*ZZT
      ZETA2P = CSIN(ZETA1P)
      ZZN = (ZP(K)-ZN)/ZDN
      ZETA3P = ZPI*ZZN
      ZETA4P = CSIN(ZETA3P)
      ZFP = ZETA2P/ZETA4P
      RDP(K) = CABS(ZFP)
      THP(K)= ATAN2(AIMAG(ZFP),REAL(ZFP))
   24 CONTINUE
C  NOW ADD THE LEADING- AND TRAILING-EDGE POINTS, AND STORE THE G(Z)
C  ARRAY AS E(KJ)*EXP(I*THT(KJ)),WHERE KJ=1,KJMX AS YOU GO FROM TE AROUND
C  THE PRESSURE SIDE TO THE LE (KJ=KJLE) AND THEN ALONG THE SUCTION SIDE
C  BACK TO THE TE AGAIN (KJ=KJMX).
      IF(ITE.EQ.1) GO TO 13
      E(1)=0.0
      THT(1)=0.0
      GO TO 14
   13 ZETA2=CSIN(ZPI*(ZTE-ZT)/ZDN)
      ZETA4=CSIN(ZPI*(ZTE-ZN)/ZDN)
      ZFP=ZETA2/ZETA4
      E(1)=CABS(ZFP)
      THT(1)=ATAN2(AIMAG(ZFP),REAL(ZFP))
   14 IF(ILE.EQ.1) GO TO 15
      E(KJLE) = 1.0E+08
      THT(KJLE) = 0.5*(THP(1)+THS(1))
      GO TO 16
   15 ZETA2=CSIN(ZPI*(ZLE-ZT)/ZDN)
      ZETA4=CSIN(ZPI*(ZLE-ZN)/ZDN)
      ZFP=ZETA2/ZETA4
      E(KJLE) = CABS(ZFP)
      THT(KJLE) = ATAN2(AIMAG(ZFP),REAL(ZFP))
   16 DO 17 K = 1,KJP
      KJ = KJLE - K
      E(KJ)=RDP(K)
      THT(KJ)=THP(K)
   17 CONTINUE
      DO 18 K = 1,KJS
      KJ = KJLE + K
      E(KJ)=RDS(K)
   18 THT(KJ)=THS(K)
      E(KJMX) = E(1)
      THT(KJMX) = THT(1)
C
C  NOW ADJUST THE BRANCHES OF G(Z) SO AS TO BE CONTINUOUS ACROSS
C  THE CUT    (ALONG THE NEGATIVE REAL AXIS) OF THE ATAN2 FUNCTION.
C
  251 BR=0.0
      KA=2
      KB = KJLE + 1
```

```
      IF(ITE.EQ.0) KA = 3
      IF(THT(KA-1).GT.0.) BR = -1.
      PO=THT(KA-1)
  252 DO 321 KJ=KA,KJLE
      CHG=THT(KJ)-PO
      PO=THT(KJ)
      IF(ABS(CHG).LE.PI) GO TO 321
      IF(CHG.GT.PI) BR=BR-1.0
      IF(CHG.LT.-PI) BR=BR+1.0
  321 THT(KJ)=THT(KJ)+BR*TPI
      IF(ILE.EQ.1) GO TO 326
      BR = BR + 1.
      PO = THT(KB)
      THT(KB) = THT(KB) + BR*TPI
      KB = KB + 1
  326 DO 327 KJ = KB,KJMX
      CHG = THT(KJ)-PO
      PO = THT(KJ)
      IF(ABS(CHG).LE.PI) GO TO 327
      IF(CHG.GT.PI) BR = BR - 1.
      IF(CHG.LT.-PI) BR = BR + 1.
  327 THT(KJ) = THT(KJ) + BR*TPI
      DO 323 K=1,KJP
      THP(K) = THT(KJLE-K)
  323 CONTINUE
      DO 322 K = 1,KJS
      THS(K) = THT(KJLE+K)
  322 CONTINUE
C
      DO 25 K = 1,KJS
      ARS = EX*THS(K)
      RS = RDS(K)**EX
      RDSX(K) = RS
      THSX(K) = ARS
   25 CONTINUE
      DO 26 K = 1,KJP
      ARP = EX*THP(K)
      RP = RDP(K)**EX
      RDPX(K) = RP
      THPX(K) = ARP
   26 CONTINUE
      WRITE(6,241)
  241 FORMAT(//10X,'BLADE-SURFACE IMAGES IN THE G - PLANE (RATIO OF',
     *    ' SINES) AND CAP OMEGA PLANE(G**1/KAPPA) ,AND',
     *    // 9X,' RADII AND ANGLES USED IN SELECTING THE PROPER',
     *    ' BRANCHES OF THE RATIO OF SINE FUNCTIONS ARE:',
     *    //3X,'KJ',15X,'G',25X,'CAP OMEGA',13X,'R',11X,'THETA',
     *    9X,'R**EX',7X,'THETA*EX',)
      XS=E( 1)*COS(THT( 1))
      YS=E( 1)*SIN(THT( 1))
      RDTX = E(1)**EX
      THTX = EX*THT(1)
      US = RDTX*COS(THTX)
      VS = RDTX*SIN(THTX)
```

```
      WRITE(6,325) XS,YS,US,VS,E(1),THT(1),RDTX,THTX
  232 FORMAT(I5,1P8E14.5)
  324 FORMAT('  LE ',1P8E14.5)
  325 FORMAT('  TE ',1P8E14.5)
      KJLM = KJLE - 1
      DO 65 KJ = 2,KJLM
      K = KJLE - KJ
      XP = RDP(K)*COS(THP(K))
      YP = RDP(K)*SIN(THP(K))
      UP = RDPX(K)*COS(THPX(K))
      VP = RDPX(K)*SIN(THPX(K))
      WRITE(6,232)KJ,XP,YP,UP,VP,RDP(K),THP(K),RDPX(K),THPX(K)
   65 CONTINUE
      XS = E(KJLE)*COS(THT(KJLE))
      YS = E(KJLE)*SIN(THT(KJLE))
      RDLX = E(KJLE)**EX
      THLX = EX*THT(KJLE)
      US = RDLX*COS(THLX)
      VS = RDLX*SIN(THLX)
      WRITE(6,324)XS,YS,US,VS,E(KJLE),THT(KJLE),RDLX,THLX
      KJMXM = KJMX - 1
      KJLP = KJLE + 1
      DO 31 KJ = KJLP,KJMXM
      K = KJ - KJLE
      XS = RDS(K)*COS(THS(K))
      YS = RDS(K)*SIN(THS(K))
      US = RDSX(K)*COS(THSX(K))
      VS = RDSX(K)*SIN(THSX(K))
      WRITE(6,232) KJ,XS,YS,US,VS,RDS(K),THS(K),RDSX(K),THSX(K)
   31 CONTINUE
      XP = E(KJMX)*COS(THT(KJMX))
      YP = E(KJMX)*SIN(THT(KJMX))
      RDTX = E(KJMX)**EX
      THTX = EX*THT(KJMX)
      UP = RDTX*COS(THTX)
      VP = RDTX*SIN(THTX)
      WRITE(6,325)XP,YP,UP,VP,E(KJMX),THT(KJMX),RDTX,THTX
      WRITE(6,242)ZPLUS,ZMINUS
  242 FORMAT(//10X,'POINTS AT INFINITY ARE LOCATED IN THE CAP OMEGA',
     *    ' PLANE AT:',
     *    //15X,'PLUS:',1P2E15.4,'   MINUS:',2E15.4,//)
C
C  DETERMINATION OF ZC SUCH AS TO MINIMIZE THE RATIO RMAX/RMIN IN THE
C  L.C. OMEGA PLANE
C
      M=1
      ZE = (ZMGA-ZMGB)/(ZPLUS-ZMINUS)
      ZF = (ZMGA*ZMINUS-ZMGB*ZPLUS)/(ZPLUS-ZMINUS)
      ZG = (ZMGA*ZPLUS-ZMGB*ZMINUS)/(ZPLUS-ZMINUS)
      WRITE(6,601)
  601 FORMAT('1ITER',11X,'ZD',22X,'ZB',22X,'ZC',18X,'ZOMSTR',19X,'ZNTRD'
     1/13X,'RMIN',8X,'RMAX',7X,'RATIO'//' (ZA(KJ),KJ=1,KJMX)')
  250 ITER=1
      RATIO=0.0
      IF(IGOT.EQ.1) GO TO 60
C
C  FOR A FIRST GUESS, USE ZC=(-1.0,+1.0)
C
```

```
      ZC=CMPLX(-1.0,1.0)
C
   60 ZB = (ZMGA*ZPLUS-ZMGB*ZMINUS-ZC*(ZMGA-ZMGB))/(ZPLUS-ZMINUS)
      ZD = (ZMGB*ZPLUS-ZMGA*ZMINUS+(ZMGA-ZMGB)/ZC)/(ZPLUS-ZMINUS)
   68 CONTINUE
      DO 75 K = 1,KJS
      RS = RDSX(K)
      ARS = THSX(K)
      ZOMS(K) = CMPLX(RS*COS(ARS),RS*SIN(ARS))
      ZOMS(K) = (ZD-ZB*ZOMS(K)/ZC)/(Z1-ZOMS(K)/ZC)
   75 CONTINUE
      DO 85 K = 1,KJP
      RP = RDPX(K)
      ARP = THPX(K)
      ZOMP(K) = CMPLX(RP*COS(ARP),RP*SIN(ARP))
      ZOMP(K) = (ZD-ZB*ZOMP(K)/ZC)/(Z1-ZOMP(K)/ZC)
   85 CONTINUE
      IF(ILE.EQ.1) GO TO 11
      ZOMLE = ZB
      GO TO 12
   11 RD = E(KJLE)
      TH = THT(KJLE)
      RS = RD**EX
      TH = EX*TH
      ZOMLE = CMPLX(RS*COS(TH),RS*SIN(TH))
      ZOMLE = (ZD-ZB*ZOMLE/ZC)/(Z1-ZOMLE/ZC)
   12 IF(ITE.EQ.1) GO TO 32
      ZOMTE = ZD
      ZA(1) = ZOMTE
      GO TO 33
   32 RD=E(1)
      TH=THT(1)
      RS=RD**EX
      TH=EX*TH
      ZOMTE=CMPLX(RS*COS(TH),RS*SIN(TH))
      ZOMTE=(ZD-ZB*ZOMTE/ZC)/(Z1-ZOMTE/ZC)
      ZA(1)=ZOMTE
   33 DO 76 KJ = 2,KJLM
      K = KJLE - KJ
   76 ZA(KJ) = ZOMP(K)
      ZA(KJLE) = ZOMLE
      DO 77 K = 1,KJS
      KJ = KJLE + K
   77 ZA(KJ) = ZOMS(K)
      ZA(KJMX) = ZA(1)
      ZNTRD = CMPLX(0.0,0.0)
      AREA = 0.0
      RMIN=CABS(ZA(1))
      RMAX=RMIN
      ZMAX=ZA(1)
      ZMIN=ZA(1)
      KJMXX = 1
      KJMN=1
      DO 78 KJ = 2,KJMX
      DAREA = ABS(REAL(ZA(KJ-1))*AIMAG(ZA(KJ))-REAL(ZA(KJ))*
     *   AIMAG(ZA(KJ-1)))/2.0
```

```
      ZBR = (ZA(KJ-1)+ZA(KJ))/3.0
      ZNTRD = ZNTRD + ZBR*DAREA
      RABS=CABS(ZA(KJ))
      IF(RABS.GE.RMIN) GO TO 79
      RMIN=RABS
      ZMIN=ZA(KJ)
      KJMN=KJ
      GO TO 78
   79 IF(RABS.LE.RMAX) GO TO 78
      RMAX=RABS
      ZMAX=ZA(KJ)
      KJMXX = KJ
   78 AREA = AREA + DAREA
      RATIO=RMAX/RMIN
      ZNTRD = ZNTRD/AREA
      ZOMSTR= ZC*(ZNTRD-ZD)/(ZNTRD-ZB)
      WRITE(6,602) ITER,ZD,ZB,ZC,ZOMSTR,ZNTRD,RMIN,RMAX,RATIO,
     1              (ZA(KJ),KJ=1,KJMX)
  602 FORMAT(/I5,1P10E12.4/E17.4,2E12.4/(10E13.5))
      IF(RATIO.LT.RTOL) GO TO 63
      IF(IGOT.EQ.1) GO TO 63
      ITER = ITER + 1
      IF(ITER.LE.30) GO TO 62
      WRITE(6,204)
  204 FORMAT(///10X,'TOLERANCE SPECIFIED FOR RMAX/RMIN NOT MET IN',
     *    ' 30 ITERATIONS')
      STOP
   62 CONTINUE
      IF(M.EQ.2) GO TO 66
      M=2
      ZDS=1.1*ZMIN
      KJ=KJMN
   67 RD=E(KJ)**EX
      TH=EX*THT(KJ)
      ZOM=CMPLX(RD*COS(TH),RD*SIN(TH))
      ZC=(ZOM*(ZDS-ZG)+ZE)/(ZDS+ZF-ZE*ZOM)
      GO TO 60
   66 M=1
      ZDS=0.9 *ZMAX
      KJ=KJMXX
      GO TO 67
   63 IGOT = 1
      WRITE(6,208) ZD,ZB,ZC,ZNTRD,ZOMSTR
  208 FORMAT(//10X,  'CONSTANTS FOR MAPPING FROM ',
     *    'CAP OMEGA - PLANE TO SMALL OMEGA - PLANE ARE',
     *    //20X,'A = ',1P2E20.5,/20X,'B = ',1P2E20.5,/20X,'C = ',
     *    1P2E20.5,//20X,'ZNTRD =  ',1P2E20.5,/20X,'ZOMSTR = ',1P2E20.5)
C
C  SET UP THE ARRAYS OF THETA AND LN(R)
C
      DO 41 KJ = 1,KJMXM
      X(KJ) = ATAN2(AIMAG(ZA(KJ)),REAL(ZA(KJ)))
   41 Y(KJ) = CABS(ZA(KJ))
      X(KJMX) = X(1) + TPI
      Y(KJMX) = Y(1)
```

```
C
C  NOW ADJUST THE ARGUMENTS OF THE THETA ARRAY, SO AS TO BE CONTINUOUS
C  ACROSS THE BRANCH CUT (ALONG THE NEGATIVE REAL AXIS) OF THE
C  ATAN2 FUNCTION.  THIS ADJUSTMENT ASSUMES THAT THE CONTOUR IS
C  TRAVERSED IN A COUNTERCLOCKWISE DIRECTION.
C
      BR = 0.0
      PO = X(1)
      KOUNT = 0
      DO 410 KJ = 2,KJMXM
      CHG = X(KJ) - PO
      IF(CHG.GT.0.) GO TO 409
      IF(CHG.LT.-PI) GO TO 408
      KOUNT = KOUNT + 1
      GO TO 409
  408 BR = BR + 1.
  409 CONTINUE
      PO = X(KJ)
      X(KJ) = X(KJ) + BR*TPI
  410 CONTINUE
      IF(KOUNT.GT.0) WRITE(6,407) KOUNT
  407 FORMAT(//5X,'WARNING: THERE ARE',I3,' NEGATIVE INCREMENTS',
     *    ' IN THE ANGULAR VALUES OF SMALL OMEGA ON THE BLADE SURFACE',
     *   //10X,'CHECK WHETHER THE ZS AND ZP ARRAYS ARE INTERCHANGED',
     *   //10X,'OR WHETHER R VS. THETA IS MULTIPLE-VALUED, IN WHICH',
     *    ' CASE:',//10X,' - IF THE OPTION ICIRC = 0 IS BEING USED,',
     *    ' THE NEGATIVE INCREMENTS MUST BE REMOVED.',//13X,'TRY A',
     *    ' SMALLER VALUE OF RTOL.',
     *   //10X,' - IF ICIRC = 1, NEGATIVE INCREMENTS ARE ALLOWED;',
     *    ' HOWEVER, IF THE PHI/S ITERATIONS FAIL TO CONVERGE',//13X,
     *    ' TRY A SMALLER VALUE OF RTOL OR A LARGER VALUE OF ANGERR',
     *    ' OR BOTH.',//)
C
      RMIN = 10.0
      RMAX = 0.0
      DO 49 K = 1,KJMX
      IF(Y(K).LT.RMIN) RMIN = Y(K)
   49 IF(Y(K).GT.RMAX) RMAX = Y(K)
      WARSCH = SQRT(RMAX/RMIN) - 1.0
      IF(WARSCH.LT.0.3 ) OM = 1.0
      WRITE(6,202)
      DO 43 KJ = 1,KJMX
      ZSOMGA(KJ) = CMPLX(Y(KJ)*COS(X(KJ)),Y(KJ)*SIN(X(KJ)))
   43 Y(KJ) = ALOG(Y(KJ))
C
C  ZSOMGA NOW CONTAINS KJMX VALUES OF SMALL OMEGA ON THE BLADES
C
C  USE FFT TO FIND THE FOURIER COEFFICIENTS IN THE SMALL-OMEGA/ZETA
C  MAPPING:
C
      ZI = CMPLX(0.0,1.0)
      N = 64
      N2 = 2*N
      NP1 = N + 1
      NP2 = N + 2
```

```
      PBN = PI/FLOAT(N)
      NB2 = N/2
      NB2P1 = NB2 + 1
      ZNN = CMPLX(FLOAT(N),0.0)
      ZW2N = CMPLX(COS(PBN),SIN(PBN))
      OMM = 1.0 - OM
      DO 298 II=1,160
      BETA(II)=0.
      S(II)=0.
  298 CONTINUE
C
C  FOR ICIRC = 1, SET UP THE CONSTANTS FOR THE LN(R), THETA SPLINE FIT
C
      IF(ICIRC.EQ.0) CALL CISPLN(Y,X,E,F,KJMX,1,128,1)
C
C   THIS SECTION (ENTERED WHEN ICIRC=1) COMPUTES THE BETA AND S
C   COORDINATES, AND SET UP THE CONSTANTS FOR THE BETA AND S SPLINE
C   FIT
C
      IF (ICIRC.NE.1) GO TO 297
      ZA1(1) = CMPLX(0.,0.)
      DO 289 KJ = 2,KJMX
  289 ZA1(KJ) = ZA1(KJ-1) + ABS(ZSOMGA(KJ)-ZSOMGA(KJ-1))
C     CALL ZISPLN(ZSOMGA,ZA1,ZEE,ZFF,KJMX,1,128,1)
      CALL ZISPLN(ZSOMGA,ZA1,ZEE,ZFF,KJMX,4,128,1)
      DO 290 KJ = 1,KJMX
      BETA(KJ) = REAL(ZEE(KJ))
  290 S(KJ) = REAL(ZFF(KJ))
  295 CALL CISPLN(BETA,S,E,F,KJMX,1,128,2)
  297 CONTINUE
C
      WRITE(6,243)
  243 FORMAT(3X, 'BLADE-SURFACE IMAGE IN THE SMALL-OMEGA PLANE:',
     *   /3X,'KJ',6X,'REAL',10X,'IMAG',12X,'R',9X,'THETA',
     * 14X,'S',14X,'BETA',/)
      DO 51 KJ = 1,KJMX
      WRITE(6,299) KJ,ZA(KJ),EXP(Y(KJ)),X(KJ),S(KJ),BETA(KJ)
  299 FORMAT(I5,1P6E14.5)
   51 CONTINUE
C
      IF(ICIRC.EQ.0) CALL THDGRK
      IF(ICIRC.EQ.1) CALL BAUGRK(RMAX,RMIN)
C
C  FIND ZETAA AND ZETAB
C
      ZABST = ZMGA
      ZBBST = ZMGB
      RABST = 1.0
      RBBST = 1.0
      DRR = 0.1
      DTH = 10.0*PI/180.0
      THA = -9.0*DTH
      DO 21 I = 1,9
      R = .19  + DRR*FLOAT(I-1)
```

```
      DO 22 J = 1,19
      TH = DTH*FLOAT(J-1)  + THA
      ZTAGS = CMPLX(R*COS(TH),R*SIN(TH))
      IF (ICIRC.EQ.0)
     *   CALL OMETA(A,B,ZMG,ZTAGS,ZTANSR,G5,1.0E-00,1)
      IF(ICIRC.EQ.1)
     *CALL OMDZETA(ZOMTE,Z1,DR,DI,ZMG,ZTAGS,ZTANSR,G5,1.0E-00,1,20)
      RA = CABS(ZMG-ZMGA)
      IF(RA.GT.RABST) GO TO 23
      RABST = RA
      ZABST = ZTAGS
   23 ZTAGS = -ZTAGS
      IF (ICIRC.EQ.0)
     *   CALL OMETA(A,B,ZMG,ZTAGS,ZTANSR,G5,1.0E-00,1)
      IF(ICIRC.EQ.1)
     *CALL OMDZETA(ZOMTE,Z1,DR,DI,ZMG,ZTAGS,ZTANSR,G5,1.0E-00,1,20)
      RB = CABS(ZMG-ZMGB)
      IF(RB.GT.RBBST) GO TO 22
      RBBST = RB
      ZBBST = ZTAGS
   22 CONTINUE
   21 CONTINUE
      ZTAGS = ZABST
      M = 0
      IF (ICIRC.EQ.0)
     *   CALL OMETA(A,B,ZMGA,ZTAGS,ZTANSR,G5,1.0E-05,M)
      IF(ICIRC.EQ.1)
     *CALL OMDZETA(ZOMTE,Z1,DR,DI,ZMGA,ZTAGS,ZTANSR,G5,1.0E-05,M,20)
      IF(M.NE.5) GO TO 260
      WRITE(6,261) ZTAGS,RABST
  261 FORMAT(//5X,'OMETA FAILED TO CONVERGE FOR ZETA A:',
     *   /10X,'ZTAGS = ',1P2E13.5,'  RABST =',E13.5)
      STOP
  260 CONTINUE
      ZETAA = ZTANSR
      ZTAGS = ZBBST
      M = 0
      IF (ICIRC.EQ.0)
     *   CALL OMETA(A,B,ZMGB,ZTAGS,ZTANSR,G5,1.0E-05,M)
      IF(ICIRC.EQ.1)
     *CALL OMDZETA(ZOMTE,Z1,DR,DI,ZMGB,ZTAGS,ZTANSR,G5,1.0E-05,M,20)
      IF(M.NE.5) GO TO 262
      WRITE(6,263) ZTAGS,RBBST
  263 FORMAT(//5X,'OMETA FAILED TO CONVERGE FOR ZETA B:',
     *   /10X,'ZTAGS = ',1P2E13.5,'  RBBST =',E13.5)
      STOP
  262 CONTINUE
      ZETAB = ZTANSR
C
C  FIND GAMMA, ALPHA, BETA, AND S FOR MAPPING TO ETA - PLANE
C
      AP = CABS(ZETAA + ZETAB)
      AM = CABS(ZETAA - ZETAB)
      AB = CABS(ZETAA*ZETAB)
```

```
      CHY = (2.0-AP*AP+2.0*AB*AB)/AM/AM
      RT = SQRT(CHY*CHY-1.0)
      CA = SQRT(ABS(CHY+RT))
      CB = SQRT(ABS(CHY-RT))
      SS = AMIN1(CA,CB)
      ZAL = (2.0*ZETAA*ZETAB+(SS*SS*(ZETAA-ZETAB)-ZETAA-ZETAB)
     *  /CONJG(ZETAA))/(SS*SS*(ZETAA-ZETAB)+ZETAA+ZETAB-2.0/
     *  CONJG(ZETAA))
      ZBT = (2.0*ZETAA*ZETAB-ZAL*(ZETAA+ZETAB))/(ZETAA+ZETAB-2.0
     *    *ZAL)
      ZGM = SS*(ZETAA-ZBT)/(ZETAA-ZAL)
C
  415 WRITE(6,245) ZABST
  245 FORMAT(//10X,'BEST GUESS FOR ZETA A IS ZABST = ',1P2E12.3)
      WRITE(6,246) ZBBST
  246 FORMAT(//10X,'BEST GUESS FOR ZETA B IS ZBBST = ',1P2E12.3)
      WRITE(6,215) ZETAA,ZETAB
  215 FORMAT(// 5X,'ZETAA = ',1P2E13.5,'  ZETAB = ',2E13.5)
      WRITE(6,216) ZAL,ZBT,ZGM,SS
  216 FORMAT(// 5X,'ALPHA = ',1P2E11.3,'  BETA = ',2E11.3,
     *    '  GAMMA = ',2E11.3,'  S = ',2E11.3)
C
C  FINDING THE LOCATION OF BLADE-SURFACE POINTS IN THE ZETA PLANE ONLY
C  INVOLVES PHI(THETA), SINCE R = 1.  USE SPLINE INTERPOLATION
C
      PHI(129) = PHI(1) + TPI
      IF (ICIRC.EQ.1) GO TO 211
      E(129)=E(1)+TPI
      CALL CISPLN(PHI,E,THT,F,129,1,1,2)
      CALL CISPLN(PHI,E,X,F,129,2,KJMX,2)
      GO TO 212
  211 CONTINUE
      E(129)=S(KJMX)
      CALL CISPLN(PHI,E,THT,F,129,1,1,2)
      CALL CISPLN(PHI,E,S,F,129,2,KJMX,2)
  212 CONTINUE
      WRITE(6,217)
  217 FORMAT(//10X,'MAPPING FROM SMALL OMEGA - PLANE TO ZETA - PLANE:',
     *    // 3X,'KJ',11X,'SMALL OMEGA',23X,'ZETA',//)
      DO 54 K = 1,KJMX
      R = EXP(Y(K))
      ZX = CMPLX(R*COS(X(K)),R*SIN(X(K)))
      ZYG = CMPLX(COS(F(K)),SIN(F(K)))
      IF(K.EQ.1.OR.K.EQ.KJMX) ZYG = Z1
      ZCC(K) = ZYG
   54 WRITE(6,218) K,ZX,ZYG
  218 FORMAT(I5,1P4E15.5)
C
C  ZCC NOW CONTAINS ZETA ON THE BLADE SURFACES
C
      WRITE(6,202)
C
C  NOW DO THE MAPPING FROM THE ETA - PLANE TO THE KSI TILDE - PLANE,
C    WHERE ETA/S = SN(KSI TILDE)
C
```

```
      AK = SS*SS
      AKQ = AK*AK
      AKP = SQRT(1.0-AKQ)
      AKM = AKP*AKP
      CALL ELLPT(PB2,AK,RL,1)
      TBK = 2.0*RL
      CTR = -RL
      CAPK = RL
      CALL ELLPT(PI,AKP,RL,1)
      CAPKPM = RL
      WRITE(6,222) AK,CAPK,AKP,CAPKPM
  222 FORMAT(//10X, 'COMPLETE ELLIPTIC INTEGRALS  OF K AND K PRIME',
     *   ' ARE AS FOLLOWS:',
     *  //10X,'K(',F10.6,') = ',F10.6,5X,'K(',F10.6,') = ',F10.6)
      DO 55 I = 1,KJMX
      ZA(I) = ZGM*(ZCC(I)-ZAL)/(ZCC(I)-ZBT)
C
C  ZA(I) NOW HOLDS ETA
C
      ZTD = ZA(I)/SS
      TAU = REAL(ZTD)
      DLT = AIMAG(ZTD)
      TSQ = TAU*TAU
      DSQ = DLT*DLT
      ART = 1.0 + AKQ*(TSQ + DSQ)
      RT = SQRT((1.0-AKQ*TSQ)*(1.0-AKQ*TSQ) + AKQ*DSQ*(2.0*
     *    (1.0+AKQ*TSQ)
     *    +AKQ*DSQ))
      BRQ = 1.0 + TSQ + DSQ
      BRT = SQRT((1.0-TSQ)*(1.0-TSQ) + DSQ*(DSQ+2.0+2.0*TSQ))
      ALM = (BRQ-BRT)*(ART-RT)/4.0/AKQ/TSQ
      SGA = (TSQ + DSQ -ALM)
      SGA = SGA/(SGA+1.0-ALM*AKQ*(TSQ+DSQ))
      IF(TAU.EQ.0.0) GO TO 403
      RTALM = SQRT(ALM)*TAU/ABS(TAU)
      GO TO 404
  403 RTALM = 0.0
  404 SGN = 1.0
      IF(DLT.LT.0.0) SGN = -1.0
      RTSGA = SGN*SQRT(SGA)
      RTALM = ASIN(RTALM)
      RTSGA = ASIN(RTSGA)
      CALL ELLPT(RTALM,AK,RL,0)
      CALL ELLPT(RTSGA,AKP,AG,0)
      IF(AG.GE.0.0) GO TO 57
      AG = -AG
      RL =-RL - TBK
   57 ZA1(I) = CMPLX(RL,AG)
   55 CONTINUE
C
C  ZA1(I) NOW HOLDS KSI HAT
C
      WRITE(6,219)
  219 FORMAT(////10X,'MAPPING FROM THE ETA - PLANE TO THE KSI HAT - ',
     *' PLANE',
     *   /// 3X,'KJ',15X,'ETA',25X,'KSI HAT',//)
```

```
      DO 56 K = 1,KJMX
      WRITE(6,218) K,ZA(K),ZA1(K)
   56 CONTINUE
C
C  NOW SET UP A GRID IN THE KSI-HAT PLANE, AND MAP IT BACK
C  TO THE Z - PLANE:
C
      ZA2(1) = ZTE
      ZA2(KJMX) = ZTE
      ZA2(KJLE) = ZLE
      DO 91 KJ = 2,KJLM
      K = KJLE - KJ
   91 ZA2(KJ) = ZP(K)
      DO 92 K = 1,KJS
      KJ = KJLE + K
   92 ZA2(KJ) = ZS(K)
C
C  THE ZA2 ARRAY NOW HOLDS THE BLADE-SURFACE COORDINATES, IN THE ORDER:
C    KJ = 1: TE
C    KJ = 2,KJLM: PRESSURE SIDE, FROM TE TO LE
C    KJ = KJLE: LE
C    KJ = KJLP,KJMXM: SUCTION SIDE, FROM LE TO TE
C    KJ = KJMX: TE AGAIN
C
      KMXM1 = KMX - 1
      LMXM1 = LMX - 1
      HCPKPM = CAPKPM/2.0
      TWOCPK = 2.0*CAPK
      THCPK = 3.0*CAPK
      FCPK = 4.0*CAPK
      ZAG = ZAL*ZGM
      EXINV = 1.0/EX
      SHXTE = REAL(ZA1(1))
      IF(SHXTE.GT.CTR) SHXTE = SHXTE-FCPK
      SHK = -CAPKPM*CAPKPM/4.0/(THCPK     +SHXTE)
      SHKINV = 1.0/SHK
      IF(ISHEAR.EQ.0) SHKINV = 0.0
      DSHX = 2.0/FLOAT(KMXM1)
      DSHY = 1.0/FLOAT(LMXM1)
      ZIG = ZI*SG
      ZXYN = ZGAMMA*ZN
      ZXYT = ZGAMMA*ZT
      ZEN = CEXP(PI*ZXYN/SG)
      ZET = CEXP(PI*ZXYT/SG)
      ZENI = 1./ZEN
      ZETI = 1./ZET
      SGP = SG/TPI
      SZ = 0.5*SG
      ZEETE = ZA(1)
      ZLA = 1. + SS*ZEETE
      ZLB = 1. - SS*ZEETE
      ZLC = ZEETE + SS
      ZLD = ZEETE - SS
      ZDQ = ZAM*(1./ZLA + 1./ZLB)*SS + ZAP*(1./ZLC - 1./ZLD)
      ZDG = ZI*(SS*(1./ZLA - 1./ZLB) - 1./ZLC - 1./ZLD)
```

```
      Q = SG/TPI
      GG = -Q*ZDQ/ZDG
      ZROOT = 4.*AK*(ZAM+AK*ZAP)*(ZAP+AK*ZAM)
      ZROOT = ZROOT-GG*GG*AKM*AKM/Q/Q
      ROOT = REAL(ZROOT)
      ROOT = SQRT(ROOT)
      ZSPB = (AKM*ZI*GG/Q - ROOT)/(2.*SS*(ZAM+AK*ZAP))
      ALPO = ALP*180./PI
      GQ = GG/Q
      WUP = SQRT(1.+GQ*GQ-2.*GQ*SIN(ALP))
      WDN = SQRT(1.+GQ*GQ+2.*GQ*SIN(ALP))
      UUP = COS(ALP)/WUP
      UDN = UUP
      VDN = -(SIN(ALP)+GQ)/WUP
      VUP = -(SIN(ALP)-GQ)/WUP
      WR = WDN/WUP
      BETA1 = ATAN(VUP/UUP)*180./PI
      BETA2 = ATAN(VDN/UDN)*180./PI
      WRITE(6,226) ALPO,Q,GG,ZSPB,UUP,VUP,BETA1,UDN,VDN,BETA2,WR
  226 FORMAT(//10X,'CONSTANTS FOR INCOMPRESSIBLE-FLOW SOLUTION ARE',
     *    //10X,'ANGLE OF ATTACK IN THE ETA - PLANE =',F10.5,
     *    //10X,'Q = ',F10.5,'  GG = ',F10.5,'  ETA(STAG.PT.) = ',
     *    2F10.5,//5X,'INLET U/WO,V/WO,BETA1 =',3F10.5,
     *    //5X,'OUTLET U/WO,V/WO,BETA2,W/WO =',4F10.5,//)
      K = 1
      L = 1
      WRITE(6,202)
      WRITE(6,205)
  205 FORMAT(/5X,'MAPPING OF A GRID IN THE KSI-HAT PLANE',
     *    /5X,'AND INCOMPRESSIBLE-FLOW SOLUTION IN THE X,Y PLANE',
     *    //3X,'K    L', 8X,'KSI HAT',
     *              15X,'ETA',16X,'ZETA',13X,'SMALL OMEGA',
     *    10X,'Z MAPPED',14X,'ZXY',
     *    /13X,'U/WO',6X,'V/WO',7X,'PHI',7X,'PSI',//)
C
      CALL ZISPLN(ZSOMGA,ZCC,ZEE,ZFF,KJMX,1,128,1)
C
C
C  K - LOOP STARTS HERE
C
  760 CONTINUE
      SHX = -1.0+DSHX*FLOAT(K-1)
C
C  L - LOOP STARTS HERE
C
  770 CONTINUE
      SHY=DSHY*FLOAT(L-1)
      XIM=HCPKPM*(1.0-SHY)
      XIR = (XIM-HCPKPM)**2
      XIR = SHXTE+    TWOCPK*(1.0+SHX)+XIR*SHKINV
      ZXI(L,K) = CMPLX(XIR,XIM)
```

```
C
C  BYPASS IMAGE CALCULATIONS FOR POINTS THAT FALL ON THE IMAGES OF
C  PLUS OR MINUS INFINITY OR THE T.E.
C
      IF(ISHEAR.EQ.0) GO TO 84
      IF(L.EQ.1.AND.K.EQ.1) GO TO 73
      IF(L.EQ.1.AND.K.EQ.KMX) GO TO 73
   84 CONTINUE
      IF(L.EQ.LMX.AND.K.EQ.1) GO TO 74
      IF(L.EQ.LMX.AND.K.EQ.KMX) GO TO 74
      IF(IOE.EQ.0) GO TO 80
      IF(L.EQ.LMX.AND.K.EQ.KMXH) GO TO 81
      GO TO 80
C
C
C
   73 ZEETA(L,K) = ZEETE
      ZETA(L,K) = (ZBT*ZEETA(L,K)-ZAG)/(ZEETA(L,K)-ZGM)
      ZOMA(L,K) = ZOMTE
      ZFNL(L,K) = CMPLX(SHX,SHY)
      ZXY(L,K) = ZTE*ZGAMMA
      ZVEL(L,K) = ZBG
      ZFNP(L,K) = ZBG
      GO TO 710
   74 ZEETA(L,K) = CMPLX(SS,0.0)
      ZETA(L,K) = ZETAA
      ZOMA(L,K) = ZMGA
      ZFNL(L,K) = CMPLX(SHX,SHY)
      ZVEL(L,K) = CMPLX(UDN,VDN)
      ZFNP(L,K) = ZBG
      GO TO 710
   81 ZEETA(L,K) = CMPLX(-SS,0.0)
      ZETA(L,K) = ZETAB
      ZOMA(L,K) = ZMGB
      ZFNL(L,K) = CMPLX(SHX,SHY)
      ZVEL(L,K) = CMPLX(UUP,VUP)
      ZFNP(L,K) = ZBG
      GO TO 710
C
C
C
   80 CONTINUE
      CALL JCELFN(XIR,XIM,AKQ,AKM,RLS,AGS,1)
      ZEETA(L,K) = SS*CMPLX(RLS,AGS)
      IF(L.NE.LMX) GO TO 86
      IF(K.LE.KMXH) ZEETA(L,K) = ZEETA(L,K) - ZEPS
      IF(K.GT.KMXH) ZEETA(L,K) = ZEETA(L,K) + ZEPS
   86 CONTINUE
      ZETA(L,K) = (ZBT*ZEETA(L,K)-ZAG)/(ZEETA(L,K)-ZGM)
      IF(L.NE.1) GO TO 82
      ZEE(1) = ZETA(L,K)
      CALL ZISPLN(ZSOMGA,ZCC,ZEE,ZFF,KJMX,2,1,1)
      ZOMA(L,K) = ZFF(1)
```

```
   82 IF (ICIRC.EQ.0)
     *   CALL OMETA(A,B,ZOMA(L,K),ZETA(L,K),ZTANSR,65,1.0E-00,1)
  710 CONTINUE
      L = L + 1
      IF(L.LE.LMX) GO TO 770
      K = K + 1
      L = 1
      IF(K.LE.KMX) GO TO 760
      IF(ICIRC.EQ.0) GO TO 763
C
C  AT THIS POINT, GRID CALCULATIONS IN THE ZXI, ZEETA, AND ZETA PLANES
C  ARE COMPLETE, FOR ALL L AND K, AND IF ICIRC = 0, THE SMALL-OMEGA-
C  PLANE GRID HAS ALSO BEEN FOUND.  THE FOLLOWING CALL CALCULATES THE
C  SMALL-OMEGA GRID FOR THE CASE ICIRC = 1:
C
      CALL OMZTDR(DR,DI,ZETAB,ZETA,ZOMA,KMX,LMX,IOE)
  763 CONTINUE
      K = 1
      L = 1
  761 CONTINUE
C
C  THE K AND L  - LOOPS RESUME HERE
C
      IF(L.EQ.1.AND.K.EQ.1) GO TO 711
      IF(L.EQ.1.AND.K.EQ.KMX) GO TO 711
      IF(L.EQ.LMX.AND.K.EQ.1) GO TO 712
      IF(L.EQ.LMX.AND.K.EQ.KMX) GO TO 712
      IF(IOE.EQ.0) GO TO 762
      IF(L.EQ.LMX.AND.K.EQ.KMXH) GO TO 712
  762 CONTINUE
      ZBOM=ZC*(ZOMA(L,K)-ZD)/(ZOMA(L,K)-ZB)
      RAD = CABS(ZBOM)
      ARG = ATAN2(AIMAG(ZBOM),REAL(ZBOM))
      RADO = RAD**EXINV
      ARGO = EXINV*ARG
      ZBOMK = CMPLX(RADO*COS(ARGO),RADO*SIN(ARGO))
C
C  NOW FIND ZXY, GIVEN ZBOMK
C
      ZXY(L,K) = SGP*CLOG((ZET-ZBOMK*ZEN)/(ZETI-ZBOMK*ZENI))
      IF(L.NE.1) GO TO 93
      IF((AIMAG(ZXY(L,K))-YY(K-1,L)).LT.-SZ) ZXY(L,K) = ZXY(L,K) + ZIG
      IF((AIMAG(ZXY(L,K))-YY(K-1,L)).GT.SZ) ZXY(L,K) = ZXY(L,K) -ZIG
      GO TO 90
   93 CONTINUE
      IF((AIMAG(ZXY(L,K))-YY(K,L-1)).LT.-SZ) ZXY(L,K) = ZXY(L,K) + ZIG
      IF((AIMAG(ZXY(L,K))-YY(K,L-1)).GT.SZ) ZXY(L,K) = ZXY(L,K) - ZIG
   90 CONTINUE
   72 CONTINUE
C
      ZFNL(L,K) = CMPLX(SHX,SHY)
      ZLA = 1. + SS*ZEETA(L,K)
      ZLB = 1. - SS*ZEETA(L,K)
      ZLC = ZEETA(L,K) + SS
      ZLD = ZEETA(L,K) - SS
```

```
      ZWE = Q*(SS*(1./ZLA+1./ZLB)*ZAM+ZAP*(1./ZLC-1./ZLD))
      ZWE = ZWE+ ZI*GG*(SS*(1./ZLA-1./ZLB)-1./ZLC-1./ZLD)
      ZLA = CLOG(ZLA)
      ZLB = CLOG(ZLB)
      ZLC = CLOG(ZLC)
      ZLD = CLOG(ZLD)
      XLC = REAL(ZLC)
      XLD = REAL(ZLD)
      YLC = AIMAG(ZLC)
      YLD = AIMAG(ZLD)
      IF(YLC.LT.O.) YLC = YLC + TPI
      IF(YLD.LT.O.) YLD = YLD + TPI
      ZLC = CMPLX(XLC,YLC)
      ZLD = CMPLX(XLD,YLD)
      ZFNP(L,K) = Q*(ZAM*(ZLA-ZLB)+ZAP*(ZLC-ZLD))
      ZFNP(L,K) = ZFNP(L,K)+ZI*GG*(ZLA+ZLB-ZLC-ZLD)
C
C  CALCULATE THE VELOCITY
C
       ZDR = ZDA
       ZDC = ZETA(L,K) - ZBT
       ZDC = ZGM*(ZAL-ZBT)/ZDC/ZDC
       ZDR = ZDR*ZDC
       IF (ICIRC.EQ.0)
      *   CALL OMETA(A,B,ZOMA(L,K),ZETA(L,K),ZTANSR,G5,1.,Z)
       IF(ICIRC.EQ.1)
      *CALL OMDZETA(ZOMTE,Z1,DR,DI,ZOMA(L,K),ZETA(L,K),ZTANSR,G5,1.,Z,Z0)
       ZDR = ZDR/ZTANSR
       ZDE = ZOMA(L,K) - ZB
       ZDE = ZC*(ZD-ZB)/ZDE/ZDE
       ZDR = ZDR/ZDE
       Z = ZXY(L,K)/ZGAMMA
       ZZT = (Z-ZT)/ZDN
       ZZN = (Z-ZN)/ZDN
       ZT2 = CSIN(ZPI*ZZT)
       ZT4 = CSIN(ZPI*ZZN)
       ZT1 = CSIN(ZPI*(ZZN-ZZT))
       ZDF = ZPI*ZBOM*EX*ZT1/ZDN/ZT2/ZT4
       ZDR = ZDR*ZDF
       ZVEL(L,K) = CONJG(ZWE*ZDR)/WUP
       GO TO 711
C
C
C
  712 ZXY(L,K) = 2.0*ZA(LMX-1) - ZA(LMX-2)
  711 WRITE(6,224) K,L,ZXI(L,K),ZEETA(L,K),ZETA(L,K),ZOMA(L,K),
     *   ZFNL(L,K),ZXY(L,K)
      WRITE(6,225)ZVEL(L,K),ZFNP(L,K)
  225 FORMAT(8X,12F10.5)
      ZA(L) = ZXY(L,K)
      XX(K,L) = REAL(ZXY(L,K))
      YY(K,L) = AIMAG(ZXY(L,K))
   71 CONTINUE
      L = L + 1
      IF(L.LE.LMX) GO TO 761
```

```
      WRITE(6,202)
      IF(PNCHZA) WRITE(7,210) (ZA(L),L=1,LMX)
      K = K + 1
      L = 1
      IF(K.LE.KMX) GO TO 761
   70 CONTINUE
C
      STOP
  100 FORMAT(8F10.4)
  202 FORMAT(///)
  210 FORMAT(1P4E20.13)
  224 FORMAT(2I4,12F10.5)
      END



      SUBROUTINE CISPLN(Y,X,E,F,NP,IRTN,NRTN,NPD)
C
C  THIS SUBROUTINE FITS A CUBIC SPLINE TO A FUNCTION Y(X), DEFINED BY
C  NP PAIRS OF POINTS.  THE SECOND DERIVATIVE OF THE FUNCTION
C  IS PERIODIC.  IF NPD = 1, THE FUNCTION ITSELF IS ALSO
C  PERIODIC, WHILE IF NPD = 2, THE FUNCTION INCREASES BY 2.*PI
C  EVERY PERIOD, IN WHICH CASE  THE DATA PASSED TO THIS
C  SUBROUTINE SHOULD HAVE THE PROPERTY THAT Y(NP) = Y(1) + 2.*PI;
C  THE SUBROUTINE WILL THEN SET :
C           Y(NP+1) = Y(2) + (NPD-1)*2.*PI, AND
C           X(NP+1) = X(NP) + X)2) - X(1)
C  SOLUTIONS FOLLOW PAGES 9 - 15 OF
C  THE THEORY OF SPLINES AND THEIR APPLICATIONS, BY J. H. AHLBERG,
C  E. N. NILSON, AND J. L. WALSH, ACADEMIC PRESS, 1967
C
C
C  NOTE THAT Y,X,E,AND F ARE, RESPECTIVELY, ORDINATE, ABSCISSA,ABSCISSA,
C  AND ORDINATE.
C
      IMPLICIT REAL(A-H,O-Y),COMPLEX(Z)
      DIMENSION Y(160),X(160),E(160),F(160),BDA(160),EM(160),H(160)
      DIMENSION S(160),T(160),V(160),D(160)
      DATA TPI/6.2831853071795/
C
C  THIS SECTION (ENTERED WHEN IRTN = 1) USES THE NP PAIRS OF INPUT
C  COORDINATES X AND Y TO FIND THE COEFFICIENTS OF THE SPLINE FIT.
C
C  THESE COEFFICIENTS - HERE CALLED EM(KJ) - ARE THE SECOND DERIVATIVES
C  OF THE FUNCTION.
C
C
      IF(IRTN.EQ.2) GO TO 20
      NPM = NP - 1
      N = NP + 1
      DO 1 KJ = 2,NP
    1 H(KJ) = X(KJ) - X(KJ-1)
      H(N) = H(2)
      DO 2 KJ = 2,NP
    2 BDA(KJ) = H(KJ+1)/(H(KJ)+H(KJ+1))
```

```
      E(1) = 0.0
      F(1) = 0.0
      S(1) = 1.0
      DO 3 KJ = 2,NPM
      DN = 2.0 + (1.0 - BDA(KJ))*E(KJ-1)
      E(KJ) = -BDA(KJ)/DN
      D(KJ) =
     * 6.0*((Y(KJ+1)-Y(KJ))/H(KJ+1)
     *    -(Y(KJ)-Y(KJ-1))/H(KJ))/(H(KJ)+H(KJ+1))
      S(KJ) = -(1.0-BDA(KJ))*S(KJ-1)/DN
    3 F(KJ) = (D(KJ)-(1.0-BDA(KJ))*F(KJ-1))/DN
      Y(N) = Y(2)
      IF(NPD.EQ.2) Y(N) = Y(N) + TPI
      D(NP) = 6.0*((Y(N)-Y(NP))/H(2)
     *    -(Y(NP)-Y(NPM))/H(NP))/(H(NP)+H(2))
      NPMM = NP - 2
      T(NP) = 1.0
      V(NP) = 0.0
      DO 6 I = 1,NPMM
      KJ = NP - I
      T(KJ) = E(KJ)*T(KJ+1) + S(KJ)
      V(KJ) = E(KJ)*V(KJ+1) + F(KJ)
    6 CONTINUE
      EM(NP) = (D(NP)-BDA(NP)*V(2)-(1.0-BDA(NP))*V(NPM))/
     *    (BDA(NP)*T(2)+(1.0-BDA(NP))*T(NPM) + 2.0)
      DO 4 I = 1,NPM
      KJ = NP - I
    4 EM(KJ) = E(KJ)*EM(KJ+1) + F(KJ)    + S(KJ)*EM(NP)
      SIZE = ABS(X(NP)-X(1))/10.0
      RETURN
C
C  THIS SECTION (ENTERED WHEN IRTN = 2) RETURNS NRTN INTERPOLATED
C  VALUES OF THE ORDINATE F AT ASSIGNED VALUES OF THE ABSCISSA E.
C
   20 KJ = 2
      DO 21 J = 1,NRTN
      A = E(J)
   24 IF(A.LE.X(KJ)) GO TO 23
      KJ = KJ + 1
      IF(KJ.LE.NP ) GO TO 24
      DF = A - X(NP)
      IF(DF.GT.SIZE) WRITE(6,200) J,E(J),NP,X(NP)
  200 FORMAT(//10X,'WARNING - ENTRY IN CISPLN EXCEEDS END OF BASE ',
     *    'ARRAY',/5X,'E(',I3,') = ',1PE16.8,'  EXCEEDS X(',I3,') = ',
     *    E16.8)
      DF = A - X(1)
      IF(DF.LT.(-SIZE)) WRITE(6,201)J,E(J),X(1)
  201 FORMAT(//10X,'WARNING - ENTRY IN CISPLN IS LESS THAN THE FIRST',
     *   ' BASE POINT',
     *    /5X,'E(',I3,') = ',1PE16.8,'  IS LESS THAN X(1) = ',E16.8)
      KJ = NP
   23 DXA = X(KJ) - A
      DXB = A - X(KJ-1)
      CBA = DXA*DXA*DXA
      CBB = DXB*DXB*DXB
      F(J) = (EM(KJ-1)*CBA+EM(KJ)*CBB)/6.0/H(KJ)
     *    + DXA*(Y(KJ-1)-EM(KJ-1)*H(KJ)*H(KJ)/6.0)/H(KJ)
     *    + DXB*(Y(KJ)-EM(KJ)*H(KJ)*H(KJ)/6.0)/H(KJ)
   21 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ELLPT(AR,AK,ANS,KOMP)
      IMPLICIT REAL(A-H,O-Y),COMPLEX(Z)
      DIMENSION SQ(12)
      DATA K/1/
      DATA PI/3.14159265358 97/
C
C  THIS SUBROUTINE EVALUATES THE ELLIPTIC INTEGRAL OF THE FIRST KIND,
C  WITH ARGUMENT AR (AN ANGLE IN RADIANS), AND PARAMETER AK (A REAL
C  NUMBER).
C  THIS EVALUATION USES EQ.(14) OF:  Y. L. LUKE, 'APPROXIMATIONS
C  FOR ELLIPTIC INTEGRALS', MATH. COMP., VOL. 22 (JULY 1968), PP 627-
C  634, WITH N = 12.
C  KOMP = 0,1 FOR THE INCOMPLETE, COMPLETE INTEGRAL, RESPECTIVELY.
C
      IF(K.GT.1) GO TO 11
      K = 2
      TNP = 25.0
      DO 10 M = 1,12
      THM = PI*FLOAT(M)/TNP
      S = SIN(THM)
   10 SQ(M) = S*S
   11 AKK = AK*AK
      SM = 0.0
      IF(KOMP.EQ.1) GO TO 40
      TN = TAN(AR)
      DO 20 M = 1,12
      SG = SQRT(1.0-AKK*SQ(M))
      T = ATAN(SG*TN)
   20 SM = SM + T/SG
      ANS = (AR + 2.0*SM)/TNP
      RETURN
   40 DO 41 M = 1,12
      SG = SQRT(1.0-AKK*SQ(M))
   41 SM = SM + 1.0/SG
      ANS = PI*(1.0+2.0*SM)/2.0/TNP
      RETURN
      END


      SUBROUTINE JCELFN(RL,AG,AKQ,AKM,RLS,AGS,M)
C
C  WHEN M.EQ.1,
C  THIS SUBROUTINE RETURNS THE JACOBIAN ELLIPTIC SINE
C  OF A COMPLEX ARGUMENT:
C       RLS + I*AGS = SN(RL + I*AG,K)
C  USING THE ARITHMETIC - GEOMETRIC MEAN FORMULA (SEE P.571, HANDBOOK
C  OF MATHEMATICAL FUNCTIONS, ED. BY M. ABRAMOWITZ AND I. A. STEGUN,
C  U.S. NATIONAL BUREAU OF STANDARDS, APPLIED MATHEMATICS SERIES, 55,
C  JUNE 1964) AND THE ADDITION FORMULA FOR THE SN (SEE EQUATION 125.01
C  P. 24, OF HANDBOOK OF ELLIPTIC INTEGRALS FOR ENGINEERS AND
C  PHYSICISTS, BY P. F. BYRD AND M. D. FRIEDMAN, SPRINGER VERLAG, 1954).
C  AKQ = K**2, AKM = 1. - K**2
C
C  WHEN M.EQ.2, THE QUANTITIES RETURNED ARE THE REAL AND IMAGINARY PARTS
C  OF THE PRODUCT CN(..)*DN(..), WHICH IS THE DERIVATIVE OF THE SN(..)
C
```

```
      IMPLICIT REAL(A-H,O-Y),COMPLEX(Z)
      DIMENSION A(20),B(20),C(20),PH(20)
      K = 1
      A(1) = 1.0
      B(1) = SQRT(AKM)
      C(1) = SQRT(AKQ)
    5 DO 6 I = 2,20
      A(I) = (A(I-1)+B(I-1))/2.0
      B(I) =SQRT(A(I-1)*B(I-1))
      C(I) = (A(I-1)-B(I-1))/2.0
      IF(ABS(C(I)).LT.1.0E-09) GO TO 7
    6 CONTINUE
      WRITE(6,200) RL,AG
  200 FORMAT(///10X,'JCELFN FAILED TO CONVERGE FOR Z = ',1P2E15.4)
      STOP
    7 NM = I-1
      N = I
      IF(K.EQ.2) GO TO 20
      PH(N) = A(N)*RL*2**NM
   15 DO 11 L = 1,NM
      J = N - L
   11 PH(J) = (PH(J+1)+ASIN(C(J+1)*SIN(PH(J+1))/A(J+1)))/2.0
      IF(K.EQ.2) GO TO 40
      SNK = SIN(PH(1))
      CNK = COS(PH(1))
      DNK = CNK/COS(PH(2)-PH(1))
      K = 2
      TMP = B(1)
      B(1) = C(1)
      C(1) = TMP
      GO TO 5
   20 PH(N) = A(N)*AG*2**NM
      GO TO 15
   40 SNP = SIN(PH(1))
      CNP = COS(PH(1))
      DNP = CNP/COS(PH(2)-PH(1))
      DNM = 1.0 - SNP*SNP*DNK*DNK
      IF(M.EQ.2) GO TO 50
      RLS = SNK*DNP/DNM
      AGS = CNK*DNK*SNP*CNP/DNM
      RETURN
   50 RLA = CNK*CNP
      AGA = SNK*DNK*SNP*DNP
      RLB = DNK*CNP*DNP
      AGB = AKQ*SNK*CNK*SNP
      RLS = (RLA*RLB-AGA*AGB)/DNM/DNM
      AGS = (-RLA*AGB-RLB*AGA)/DNM/DNM
      RETURN
      END
```

```
      SUBROUTINE OMETA(A,B,ZMGA,ZTAGS,ZTANSR,N,EPS,M)
      IMPLICIT REAL(A-H,O-Y),COMPLEX(Z)
      DIMENSION A(65),B(65),ZC(65)
C
C  IF M.EQ.0,
C  THIS SUBROUTINE USES NEWTON - RAPHSON TO FIND ZETA(OMEGA):  ZMGA IS A
C  KNOWN VALUE OF OMEGA, ZTAGS IS THE INITIAL GUESS AT ZETA, ZTANSR IS
C  THE SOLUTION, AND EPS IS THE TOLERANCE ON THE ANSWER.
C  M IS RESET TO 5 AS AN ERROR RETURN IF THESE ITERATIONS
C  DO NOT CONVERGE.
C
C  IF   M.EQ.1., THIS SUBROUTINE RETURNS THE VALUE OF OMEGA (IN ZMGA)
C  FOR A GIVEN VALUE OF ZETA (IN ZTAGS).
C
C  IF M.EQ.2, THE QUANTITY RETURNED (IN ZTANSR) IS D OMEGA/D ZETA, FOR
C  GIVEN VALUES OF OMEGA (IN ZMGA) AND ZETA (IN ZTAGS)
C
      Z1 = CMPLX(1.0,0.0)
      ZETA = ZTAGS
      NM = N - 1
      IT = 1
    5 CONTINUE
      ZSMA = CMPLX(A(N),B(N))
      IF(M.EQ.1) GO TO 22
      ZSMB = ZSMA*FLOAT(NM)
      IF(M.EQ.2) GO TO 40
      DO 10 J = 1,NM
      ZSMA = ZETA*ZSMA + CMPLX(A(N-J),B(N-J))
      ZSMB = ZETA*ZSMB + CMPLX(A(N-J),B(N-J))*FLOAT(NM-J)
   10 CONTINUE
      ZEXP = CEXP(ZSMA)
      ZG = ZETA*ZEXP - ZMGA
      ZDG = ZEXP*(Z1 + ZSMB)
      ZETOLD = ZETA
      ZETA = ZETOLD - ZG/ZDG
      IF(CABS(ZETA-ZETOLD).LT.EPS) GO TO 20
      IT = IT + 1
      IF(CABS(ZETA).GT.1.0) ZETA = 0.9 *ZETA/CABS(ZETA)
      IF(IT.LE.50) GO TO 5
      M = 5
      RETURN
   20 ZTANSR = ZETA
      RETURN
   22 DO 21 J = 1,NM
      ZSMA = ZETA*ZSMA + CMPLX(A(N-J),B(N-J))
   21 CONTINUE
      ZEXP = CEXP(ZSMA)
      ZMGA = ZETA*ZEXP
      RETURN
   40 DO 41 J = 1,NM
   41 ZSMB = ZETA*ZSMB + CMPLX(A(N-J),B(N-J))*FLOAT(NM-J)
      ZTANSR = ZMGA*(Z1+ZSMB)/ZTAGS
      RETURN
      END
```

```
      SUBROUTINE SHAPE(C,H,G,EX,ZP,ZS,KJS,KJP)
      IMPLICIT REAL(A-H,O-Y),COMPLEX(Z)
      DIMENSION ZS(80),ZP(80)
      DO 10 K = 1,KJS
   10 READ(5,100) ZS(K)
      DO 20 K = 1,KJP
   20 READ(5,100) ZP(K)
  100 FORMAT(8F10.0)
      RETURN
      END
```

```
      SUBROUTINE SHUFL(N,ZA,ZCC)
C
C  THIS SUBROUTINE TAKES THE N COMPLEX VALUES IN ARRAY ZA, WHICH WERE
C  COMPUTED AND STORED IN REVERSE BINARY ORDER BY FFT AND "SHUFFLES"
C  THEM INTO PROPER ORDER USING ARRAY ZCC FOR INTERMEDIATE STORAGE.
C  N IS ASSUMED TO HAVE THE FORM 2**M.
C
      IMPLICIT REAL(A-H,O-Y),COMPLEX(Z)
      DIMENSION ZA(65),ZCC(65),IAL(6),KR(64)
      DATA KALL/0/
      DATA IAL/6*0/
      IF (KALL.EQ.1) GO TO 10
      KALL = 1
      DO 341 JP = 1,N
      J = JP - 1
      IAL(6) = J/32
      J = J - 32*IAL(6)
      IAL(5) = J/16
      J = J - 16*IAL(5)
      IAL(4) = J/8
      J = J - 8*IAL(4)
      IAL(3) = J/4
      J = J - 4*IAL(3)
      IAL(2) = J/2
      J = J - 2*IAL(2)
      IAL(1) = J
  341 KR(JP) =
     *      32*IAL(1)+16*IAL(2)+8*IAL(3)+4*IAL(4)+2*IAL(5)+IAL(6)
   10 DO 342 J = 1,N
  342 ZCC(J) = ZA(KR(J)+1)
      DO 360 JP = 1,N
  360 ZA(JP) = ZCC(JP)
      RETURN
      END
```

```
      SUBROUTINE THDGRK
C
C  THIS SUBROUTINE MAPS AN OVAL TO A UNIT CIRCLE, USING A VARIANT OF
C  THE THEODORSEN-GARRICK TRANSFORMATION AND FAST FOURIER TRANSFORM
C  TECHNIQUES.  (SEE REFERENCES AT BEGINNING OF MAIN PROGRAM).
C
      IMPLICIT REAL(A-H,O-Y),COMPLEX(Z)
      COMMON/TGINTG/N,NP1,NP2,N2,NB2,NB2P1,IP,IPMX,ITP,IWK,IMX,KJMX
      COMMON/TGCMPX/Z1,ZW2N,ZI,ZNN,ZA,ZCC,ZA1,ZA2
      COMMON/TGDBLE/PBN,OM,OMM,ANGERR,A,B,E,F,THT,PHI,X,Y,BETA,S,DR,DI
      DIMENSION X(160),Y(160),E(160),F(160),THT(160),
     * BETA(160),S(160)
      DIMENSION PHI(130),A(65),B(65),ZCC(65),DR(65),DI(65),
     *   ZA(165),ZA1(165),ZA2(165),
     *          IWK(7)
      DIMENSION ITP(100)
C
  301 DO 300 I = 1,N2
  300 PHI(I)=PBN*FLOAT(I-1)
C
      DO 10 I=1,65
      ZCC(I) = 0.
      A(I) = 0.
      B(I) = 0.
   10 CONTINUE
      DO 11 I=1,160
   11 E(I) = 0.
C
      WRITE(6,401)
14937


              WRITE(6,401)
  401 FORMAT(//10X,'PROGRESS OF PHI / THETA ITERATIONS IS AS FOLLOWS:'//
     1 3X,'IT',4X,'DEMX',4X,'NO. OF THETA REVERSALS')
C
      IT = 1
C
C  FIRST GUESS IS THETA - THETA(TRAILING EDGE) = PHI
C
      DO 315 K = 1,N2
  315 E(K) = X(1) + PHI(K)
  305 DEMX = 0.0
C
C  USE AJ AND BJ TO GET NEXT APPROXIMATION TO THETA
C
      ZCC(1) = CMPLX(0.0,0.0)
      DO 302 JP = 2,N
  302 ZCC(JP) = CMPLX(B(JP)/2.0,-A(JP)/2.0)
      ZCC(NP1) = CMPLX(0.0,0.0)
      ZW = Z1/ZW2N
      DO 303 NP = 1,NB2P1
      ZA1(NP) = ZCC(NP) + CONJG(ZCC(NP2-NP))
      ZW = ZW*ZW2N
```

```
  303 ZA2(NP) = ZW*(ZCC(NP)-CONJG(ZCC(NP2-NP)))
      DO 304 NP = 1,NB2P1
  304 ZA(NP) = ZA1(NP) + ZI*ZA2(NP)
      DO 309 NP = 2,NB2
  309 ZA(NP2-NP) =
     *   CONJG(ZA1(NP)) + ZI*CONJG(ZA2(NP))
      CALL FFT(ZA,6,IWK)
      CALL SHUFL(N,ZA,ZCC)
      B(1) = X(1) - PHI(1) - REAL(ZA(1))
      DO 306 K = 1,64
      TMP = E(2*K-1)
      E(2*K-1) = REAL(ZA(K)) + PHI(2*K-1)     + B(1)
      THT(2*K-1) = E(2*K-1)
      DEM = ABS(TMP-E(2*K-1))
      IF(DEM.GT.DEMX) DEMX = DEM
      E(2*K-1) = OM*E(2*K-1) + OMM*TMP
      TMP = E(2*K)
      E(2*K) = AIMAG(ZA(K)) + PHI(2*K)     + B(1)
      THT(2*K) = E(2*K)
      DEM = ABS(TMP-E(2*K))
      IF(DEM.GT.DEMX) DEMX = DEM
      E(2*K) = OM*E(2*K) + OMM*TMP
  306 CONTINUE
      IF(IT.NE.ITP(IP)) GO TO 840
      IP = IP + 1
      WRITE(6,841) IT
  841 FORMAT(/3X,
     *          'THETA BEFORE AND AFTER RELAXATION AT IT = ',I4)
      WRITE(6,832)(THT(I),I=1,128)
      WRITE(6,832)(E(I),I=1,128)
  832 FORMAT(1P10E13.5)
  840 CONTINUE
      IF(IT.GE.ITP(IPMX)) STOP
C
C  NOW USE THESE THETAS TO GET THE NEXT APPROXIMATION TO LN R(K)
C
      CALL CISPLN(Y,X,E,F,KJMX,2,128,1)
C
C  NOW FIND THE AJ AND BJ COEFFICIENTS CORRESPONDING TO THE LN R(K) DATA
C
      DO 310 JP = 1,N
  310 ZA(JP) = CMPLX(F(2*JP-1),F(2*JP))
      DO 307 NP = 1,N
  307 ZA(NP) = CONJG(ZA(NP))
      CALL FFT(ZA,6,IWK)
      DO 308 JP = 1,N
  308 ZA(JP) = CONJG(ZA(JP))/ZNN
      CALL SHUFL(N,ZA,ZCC)
      ZA(65) = ZA(1)
      DO 311 NP = 1,NB2P1
      ZA1(NP) = (CONJG(ZA(NP2-NP)) + ZA(NP))/2.0
  311 ZA2(NP) = ZI*(CONJG(ZA(NP2-NP)) - ZA(NP))/2.0
      ZW = ZW2N
      DO 312 NP = 1,NB2P1
      ZW = ZW/ZW2N
```

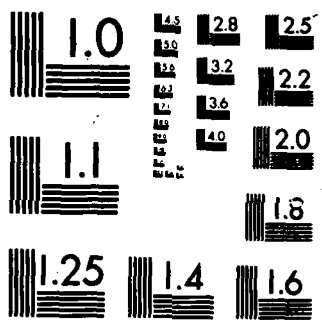
```
  312 ZCC(NP) = (ZA1(NP)+ZA2(NP)*ZW)/2.0
      ZW = Z1/ZW
      DO 313 I = 2,NB2
      ZW = ZW*ZW2N
      NP = NB2 + I
      ZBB = (ZA1(NP2-NP)+ZA2(NP2-NP)*ZW)/2.0
  313 ZCC(NP) = CONJG(ZBB)
      A(1) =   REAL(ZCC(1))
      ZCC(NP1)=0.5 *CONJG(ZA1(1)-ZA2(1))
      A(65) = REAL(ZCC(65))
      DO 314 NP = 2,N
      A(NP) = 2.0*REAL(ZCC(NP))
  314 B(NP) = 2.0*AIMAG(ZCC(NP))
C
C  CHECK FOR CONVERGENCE
C
      IF(IT.EQ.1) DEMX = 1.0
      NRV = 0
      DO 809 LL = 2,128
  809 IF(E(LL).LE.E(LL-1)) NRV = NRV + 1
      WRITE(6,400) IT,DEMX,NRV
  400 FORMAT(I5,1PE12.3,I8)
      IF(DEMX.LT.ANGERR ) GO TO 316
      IT = IT + 1
      IF(IT.LE.IMX) GO TO 305
      WRITE(6,213)
  213 FORMAT(//5X,'ITERATIONS FOR A AND B DID NOT CONVERGE')
      STOP
C
  316 WRITE(6,203) IT,OM,DEMX
  203 FORMAT(//10X,'A AND B ITERATIONS CONVERGED AT IT = ',I3,
     *    '   USING OM = ',F6.3,'  .  MAXIMUM ANGULAR ERROR =',
     *   1PE12.3,'  RADIANS')
      WRITE(6,220)
  220 FORMAT(// 4X,'K',11X,'THETA',15X,'PHI',//)
      DO 69 K = 1,128
   69 WRITE(6,221) K,E(K),PHI(K)
  221 FORMAT(I5,1P2E20.6)
C
      RETURN
      END



      SUBROUTINE BAUGRK(RMAX,RMIN)
C
C   THIS SUBROUTINE MAPS AN OVAL TO A UNIT CIRCLE, USING THE
C   HALSEY-JAMES AND FAST FOURIER TRANSFORM TECHNIQUES.
C
      IMPLICIT REAL(A-H,O-Y),COMPLEX(Z)
      COMMON/TGINTG/N,NP1,NP2,N2,NB2,NB2P1,IP,IPMX,ITP,IWK,IMX,KJMX
      COMMON/TGCMPX/Z1,ZW2N,ZI,ZNN,ZA,ZCC,ZA1,ZA2
      COMMON/TGDBLE/PBN,OM,OMM,ANGERR,A,B,E,F,THT,PHI,X,Y,BETA,S,DR,DI
      DIMENSION X(160),Y(160),E(160),F(160),THT(160),
     * BETA(160),S(160)
```

```
      DIMENSION PHI(130),A(65),B(65),ZCC(65),DR(65),DI(65),
     * ZA(165),ZA1(165),ZA2(165),IWK(7)
      DIMENSION ITP(100)
C
      DO 8 I=1,N2+1
    8 PHI(I)=PBN*FLOAT(I-1)
C
      DO 9 I=1,NP1
      ZCC(I)=CMPLX(0.,0.)
      DI(I)=0.
      DR(I)=0.
    9 CONTINUE
      DO 10 I=1,160
   10 E(I)=0.
      IT=0
C
C    FIRST GUESS IS ABS(DOMEGA/DZETA) = (RMIN+RMAX)/2.
C
      RAV=(RMAX+RMIN)/2.
      DO 11 J=1,N2+1
   11 F(J)=RAV
      HPBN=PBN/2.
      WRITE(6,401)
  401 FORMAT(//10X,'PROGRESS OF PHI/S ITERATIONS IS AS FOLLOWS:'//
     *3X,'IT',4X,'DSMX',6X,'NO. OF ARG(DOMEGA/DZETA) REVERSALS')
  999 CONTINUE
      IT=IT+1
      IF (IT.LE.IMX) GO TO 801
      WRITE(6,800)
  800 FORMAT(//5X,'ITERATIONS FOR DI AND DR DID NOT CONVERGE')
      STOP
  801 CONTINUE
      DSMX=0.
C
C    USE ABS(DOMEGA/DZETA) TO APPROXIMATE THE NEXT S
C
      E(1)=0.
      DO 12 I=2,N2+1
      THT(I)=E(I)
12    E(I)=E(I-1)+HPBN*(F(I)+F(I-1))
      DO 13 I=2,N2+1
      E(I)=S(KJMX)*E(I)/E(N2+1)
      DS=ABS(THT(I)-E(I))
      IF (DS.LE.DSMX) GO TO 15
      ESMAX=E(I)
      DSMX=DS
      IDKMAX=I
   15 CONTINUE
      TMP=THT(I)
      THT(I)=E(I)
      E(I)=OM*E(I)+OMM*TMP
   13 CONTINUE
      IF (IT.NE.ITP(IP)) GO TO 840
      THT(1)=E(1)
      IP=IP+1
```

```
      WRITE(6,841) IT
  841 FORMAT(13X,'S BEFORE AND AFTER RELAXATION AT IT = ',I4)
      WRITE(6,832) (THT(I),I=1,N2+1)
      WRITE(6,832) (E(I),I=1,N2+1)
  832 FORMAT(1P10E13.5)
  840 CONTINUE
      IF (IT.GE.ITP(IPMX)) STOP
C
C    CHECK FOR CONVERGENCE
C
      IF (IT.EQ.1) DSMX=1.
      NRV=0
      DO 809 LL=2,N2
  809 IF (E(LL).LE.E(LL-1)) NRV=NRV+1
      WRITE(6,400) IT,DSMX,NRV
  400 FORMAT(I5,1PE12.3,I8)
      IF (DSMX.LT.ANGERR) GO TO 900
C
C    NOW USE THESE SS TO APPROXIMATE THE NEXT BETAS
C
      CALL CISPLN(BETA,S,E,THT,KJMX,2,128,1)
C
C    CALCULATING THE ARG(DOMEGA/DZETA)
C
      PB2=HPBN*FLOAT(N)
      DO 14 I=1,N2
   14 THT(I)=THT(I)-PHI(I)-PB2
C
C    NOW FIND THE DIJ AND DRJ COEFFICIENTS CORRESPONDING TO THE
C    ARG(DOMEGA/DZETA) DATA
C
      DO 310 JP=1,N
  310 ZA(JP)=CMPLX(THT(2*JP-1),THT(2*JP))
      DO 307 NP=1,N
  307 ZA(NP)=CONJG(ZA(NP))
      CALL FFT(ZA,6,IWK)
      DO 308 JP=1,N
  308 ZA(JP)=CONJG(ZA(JP))/ZNN
      CALL SHUFL(N,ZA,ZCC)
      ZA(NP1)=ZA(1)
      DO 311 NP=1,NB2P1
      ZA1(NP)=(CONJG(ZA(NP2-NP))+ZA(NP))/2.
  311 ZA2(NP)=ZI*(CONJG(ZA(NP2-NP))-ZA(NP))/2.
      ZW=ZW2N
      DO 312 NP=1,NB2P1
      ZW=ZW/ZW2N
  312 ZCC(NP)=(ZA1(NP)+ZA2(NP)*ZW)/2.
      ZW=Z1/ZW
      DO 313 I=2,NB2
      ZW=ZW*ZW2N
      NP=NB2+I
      ZBB=(ZA1(NP2-NP)+ZA2(NP2-NP)*ZW)/2.
  313 ZCC(NP)=CONJG(ZBB)
      DI(1)=REAL(ZCC(1))
      ZCC(NP1)=.5*CONJG(ZA1(1)-ZA2(1))
      DI(NP1)=REAL(ZCC(NP1))
      DO 314 NP=2,N
      DI(NP)=2.*REAL(ZCC(NP))
  314 DR(NP)=-2.*AIMAG(ZCC(NP))
```

AD-A164 070 RESEARCH ON TURBINE FLOWFIELD ANALYSIS METHODS(U) 2/2
CALSPAN ADVANCED TECHNOLOGY CENTER BUFFALO NY W J RAE
JAN 85 CALSPAN-7177-A-3 AFOSR-TR-85-1220
UNCLASSIFIED F49620-83-C-0096 F/G 20/4 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
C
C   NOW USE DIJ AND DRJ TO GET THE NEXT APPROXIMATION OF
C   ABS(DOMEGA/DZETA)
C
      DODZTO=ALOG(E(2)/PBN)
      ZCC(1)=CMPLX(0.,0.)
      DO 302 JP=2,N
  302 ZCC(JP)=CMPLX(DR(JP)/2.,DI(JP)/2.)
      ZCC(NP1)=CMPLX(0.,0.)
      ZW=Z1/ZW2N
      DO 303 NP=1,NB2P1
      ZA1(NP)=ZCC(NP)+CONJG(ZCC(NP2-NP))
      ZW=ZW*ZW2N
  303 ZA2(NP)=ZW*(ZCC(NP)-CONJG(ZCC(NP2-NP)))
      DO 304 NP=1,NB2P1
  304 ZA(NP)=ZA1(NP)+ZI*ZA2(NP)
      DO 309 NP=2,NB2
  309 ZA(NP2-NP)=CONJG(ZA1(NP))+ZI*CONJG(ZA2(NP))
      CALL FFT(ZA,6,IWK)
      CALL SHUFL(N,ZA,ZCC)
      DR(1)=DODZTO-REAL(ZA(1))
      DO 20 K=1,N
      F(2*K-1)=EXP(DR(1)+REAL(ZA(K)))
      F(2*K)=EXP(DR(1)+AIMAG(ZA(K)))
   20 CONTINUE
      F(N2+1)=F(1)
      GO TO 999
C
  900 WRITE(6,203) IT,OM,DSMX
  203 FORMAT(//10X,'DI AND DR ITERATIONS CONVERGED AT IT= ',I3,
     *    '   USING OM = ',F6.3,' .  MAXIMUM ARC LENGTH ERROR = ',
     *    1PE12.3)
      CALL CISPLN(BETA,S,E,THT,KJMX,2,128,1)
      WRITE(6,220)
  220 FORMAT(//4X,'K',11X,'S',19X,'BETA',15X,'PHI',//)
      DO 69 K=1,N2
   69 WRITE(6,221) K,E(K),THT(K),PHI(K)
  221 FORMAT(I5,1P3E20.6)
C
      RETURN
      END



      SUBROUTINE SIMPSON(Z0,ZN,A,B,M,N,ZI)
      IMPLICIT REAL(A-H,O-Y),COMPLEX(Z)
      DIMENSION A(65),B(65)
      IF (Z0.NE.ZN) GO TO 11
      ZI=0.
      RETURN
   11 CONTINUE
      ZM=CMPLX(2.*FLOAT(M),0.)
      ZH=(ZN-Z0)/ZM
      NM=N-1
      ZSZ0=CMPLX(A(N),B(N))
      ZSZN=CMPLX(A(N),B(N))
```

```
      DO 10 J=1,NM
      ZSZO=ZSZO*ZO+CMPLX(A(N-J),B(N-J))
   10 ZSZN=ZSZN*ZN+CMPLX(A(N-J),B(N-J))
      ZIO=CEXP(ZSZO)+CEXP(ZSZN)
      ZI1=CMPLX(0.,0.)
      ZI2=CMPLX(0.,0.)
      MM=2*M-1
      DO 20 I=1,MM
      ZZ=ZO+CMPLX(FLOAT(I),0.)*ZH
      ZSZZ=CMPLX(A(N),B(N))
      DO 30 J=1,NM
   30 ZSZZ=ZSZZ*ZZ+CMPLX(A(N-J),B(N-J))
      IF (MOD(I,2).NE.0) GO TO 31
      ZI2=ZI2+CEXP(ZSZZ)
      GO TO 32
   31 ZI1=ZI1+CEXP(ZSZZ)
   32 CONTINUE
   20 CONTINUE
      ZI=ZH*(ZIO+2.*ZI2+4.*ZI1)/3.
      RETURN
      END


      SUBROUTINE OMDZETA(ZOMST,ZTSTRT,A,B,ZMGA,ZTAGS,ZTANSR,N,EPS,M,NS)
C
C   THIS SUBROUTINE PERFORMS THE SAME FUNCTION FOR ICIRC = 1 AS DOES
C   OMETA FOR ICIRC = 0, NAMELY:
C   - FOR M NOT EQUAL TO 1 OR 2, IT RETURNS ZETA (IN ZTANSR) FOR A
C     GIVEN VALUE OF SMALL OMEGA (IN ZMGA), USING NEWTON-RAPHSON
C   - FOR M = 1, IT RETURNS SMALL OMEGA (IN ZMGA) FOR A GIVEN VALUE
C     OF ZETA (IN ZTAGS)
C   - FOR M = 2, IT RETURNS D(SMALL OMEGA)/D(ZETA) (IN ZTANSR) FOR
C   GIVEN VALUES OF ZETA (IN ZTAGS).
C   THE INTEGRATIONS OF D(SMALL OMEGA)/D(ZETA) ARE DONE IN SUBROUTINE
C   SIMPSON.
C   ZOMST IS THE VALUE OF SMALL OMEGA AT THE START OF THE INTEGRATION.
C   ZTSTRT IS THE VALUE OF ZETA AT THE START OF THE INTEGRATION.
C   NS IS THE NUMBER OF STEPS TO BE USED BY SUBROUTINE SIMPSON.
C   N IS THE NUMBER OF A AND B COEFFICIENTS.
      IMPLICIT REAL(A-H,O-Y),COMPLEX(Z)
      DIMENSION A(65),B(65)
      ZI=CMPLX(1.,0.)
      ZETA=ZTAGS
      NM=N-1
      IT=1
    5 CONTINUE
      ZSMA=CMPLX(A(N),B(N))
      IF (M.EQ.1) GO TO 22
      IF (M.EQ.2) GO TO 40
      CALL SIMPSON(ZTSTRT,ZETA,A,B,NS,N,ZSM)
      ZG=ZSM+ZOMST-ZMGA
      DO 10 J=1,NM
   10 ZSMA=ZSMA*ZETA+CMPLX(A(N-J),B(N-J))
      ZDG=CEXP(ZSMA)
```

```
      ZETOLD=ZETA
      ZETA=ZETOLD-ZG/ZDG
      IF (CABS(ZETA-ZETOLD).LT.EPS) GO TO 20
      IT=IT+1
      IF (CABS(ZETA).GT.1.) ZETA=.9*ZETA/CABS(ZETA)
      IF (IT.LE.100) GO TO 5
      M=5
      RETURN
   20 ZTANSR=ZETA
      RETURN
   22 CALL SIMPSON(ZTSTRT,ZETA,A,B,NS,N,ZMGA)
      ZMGA=ZMGA+ZOMST
      RETURN
   40 DO 41 J=1,NM
   41 ZSMA=ZSMA*ZETA+CMPLX(A(N-J),B(N-J))
      ZTANSR=CEXP(ZSMA)
      RETURN
      END


      SUBROUTINE ZISPLN(Y,X,E,F,NP,IRTN,NRTN,NPD)
C
C  THIS IS A COMPLEX-VALUED VERSION OF CISPLN.
C  THIS SUBROUTINE FITS A CUBIC SPLINE TO A FUNCTION Y(X), DEFINED BY
C  NP PAIRS OF POINTS.  THE SECOND DERIVATIVE OF THE FUNCTION
C  IS PERIODIC, WITH PERIOD 2*PI.  IF NPD = 1, THE FUNCTION
C  ITSELF IS ALSO PERIODIC, WHILE IF NPD = 2, THE FUNCTION INCREASES
C  BY 2.*PI EVERY PERIOD: Y(X+2.*PI) = Y(X) + 2.*PI.  SOLUTIONS FOLLOW
C  PAGES 9 - 15 OF
C  THE THEORY OF SPLINES AND THEIR APPLICATIONS, BY J. H. AHLBERG,
C  E. N. NILSON, AND J. L. WALSH, ACADEMIC PRESS, 1967
C
C
C  NOTE THAT Y,X,E,AND F ARE, RESPECTIVELY, ORDINATE, ABSCISSA,ABSCISSA,
C  AND ORDINATE.
C
      IMPLICIT COMPLEX(A-H,O-Z)
      REAL SIZE,DF,DYDX,DA,DB,DC,DL
      DIMENSION Y(160),X(160),E(160),F(160),BDA(160),EM(160),H(160)
      DIMENSION S(160),T(160),V(160),D(160)
      TPI = CMPLX(8.*ATAN(1.),0.)
C
C  THIS SECTION (ENTERED WHEN IRTN = 1) USES THE NP PAIRS OF INPUT
C  COORDINATES X AND Y TO FIND THE COEFFICIENTS OF THE SPLINE FIT.
C
C  THESE COEFFICIENTS - HERE CALLED EM(KJ) - ARE THE SECOND DERIVATIVES
C  OF THE FUNCTION.
C
C
      NPM = NP - 1
      N = NP + 1
      IF(IRTN.EQ.2) GO TO 20
      IF(IRTN.EQ.3) GO TO 30
      IF(IRTN.EQ.4) GO TO 40
```

```
      DO 1 KJ = 2,NP
    1 H(KJ) = X(KJ) - X(KJ-1)
      H(N) = H(2)
      DO 2 KJ = 2,NP
    2 BDA(KJ) = H(KJ+1)/(H(KJ)+H(KJ+1))
      E(1) = CMPLX(0.,0.)
      F(1) = CMPLX(0.,0.)
      S(1) = CMPLX(1.,0.)
      DO 3 KJ = 2,NPM
      DN = 2.0 + (1.0 - BDA(KJ))*E(KJ-1)
      E(KJ) = -BDA(KJ)/DN
      D(KJ) =
     * 6.0*((Y(KJ+1)-Y(KJ))/H(KJ+1)
     *    -(Y(KJ)-Y(KJ-1))/H(KJ))/(H(KJ)+H(KJ+1))
      S(KJ) = -(1.0-BDA(KJ))*S(KJ-1)/DN
    3 F(KJ) = (D(KJ)-(1.0-BDA(KJ))*F(KJ-1))/DN
      Y(N) = Y(2)
      IF(NPD.EQ.2) Y(N) = Y(N) + TPI
      D(NP) = 6.0*((Y(N)-Y(NP))/H(2)
     *    -(Y(NP)-Y(NPM))/H(NP))/(H(NP)+H(2))
      NPMM = NP - 2
      T(NP) = CMPLX(1.,0.)
      V(NP) = CMPLX(0.,0.)
      DO 6 I = 1,NPMM
      KJ = NP - I
      T(KJ) = E(KJ)*T(KJ+1) + S(%J)
      V(KJ) = E(KJ)*V(KJ+1) + F(KJ)
    6 CONTINUE
      EM(NP) = (D(NP)-BDA(NP)*V(2)-(1.0-BDA(NP))*V(NPM))/
     *    (BDA(NP)*T(2)+(1.0-BDA(NP))*T(NPM) + 2.0)
      DO 4 I = 1,NPM
      KJ = NP - I
    4 EM(KJ) = E(KJ)*EM(KJ+1) + F(KJ)    + S(KJ)*EM(NP)
      SIZE = ABS(X(NP)-X(1))/10.0
      RETURN
C
C  THIS SECTION (ENTERED WHEN IRTN = 2) RETURNS NRTN INTERPOLATED
C  VALUES OF THE ORDINATE F AT ASSIGNED VALUES OF THE ABSCISSA E.
C
   20 KJ = 2
      DO 21 J = 1,NRTN
      A = E(J)
   24 DA = REAL(A) - REAL(X(KJ))
      DL = AIMAG(A) - AIMAG(X(KJ))
      DA = DA*DA + DL*DL
      DB = REAL(A) - REAL(X(KJ-1))
      DL = AIMAG(A) - AIMAG(X(KJ-1))
      DB = DB*DB + DL*DL
      DC = REAL(X(KJ)) - REAL(X(KJ-1))
      DL = AIMAG(X(KJ)) - AIMAG(X(KJ-1))
      DC = DC*DC + DL*DL
      IF(DA.LE.DC.AND.DB.LE.DC) GO TO 23
      KJ = KJ + 1
      IF(KJ.LE.NP ) GO TO 24
      DF = ABS(A - X(NP))
```

```
      IF(DF.GT.SIZE) WRITE(6,200) J,E(J),NP,X(NP)
  200 FORMAT(//10X,'WARNING - ENTRY IN CISPLN EXCEEDS END OF BASE ',
     *   'ARRAY',/5X,'E(',I3,') = ',1P2E16.8,'  EXCEEDS X(',I3,') = ',
     *   2E16.8)
      DF = ABS(A - X(1))
      IF(DF.LT.(-SIZE)) WRITE(6,201)J,E(J),X(1)
  201 FORMAT(//10X,'WARNING - ENTRY IN CISPLN IS LESS THAN THE  IRST',
     *  ' BASE POINT',
     *   /5X,'E(',I3,') = ',1P2E16.8,'  IS LESS THAN X(1) =     16.8)
      KJ = NP
   23 DXA = X(KJ) - A
      DXB = A - X(KJ-1)
      CBA = DXA*DXA*DXA
      CBB = DXB*DXB*DXB
      F(J) = (EM(KJ-1)*CBA+EM(KJ)*CBB)/6.0/H(KJ)
     *   + DXA*(Y(KJ-1)-EM(KJ-1)*H(KJ)*H(KJ)/6.0)/H(KJ)
     *   + DXB*(Y(KJ)-EM(KJ)*H(KJ)*H(KJ)/6.0)/H(KJ)
   2: CONTINUE
      RETURN
C
C   THIS SECTION (ENTERED WHEN IRTN=3) CONVERTS THE NP PAIRS
C   OF INPUT COORDINATES X AND Y TO INTRINSIC COORDINATES
C   BETA AND S, WHICH ARE STORED AS:
C       BETA = REAL(E), S = REAL(F)
C  NOTE THAT THE SECOND DERIVATIVES OF THIS SPLINE FIT WERE
C  ESTABLISHED IN A PREVIOUS CALL, IN WHICH THE FUNCTION SMALL OMEGA
C  WAS PASSED IN Y, AND THE APPROXIMATE ARCLENGTH IN X
C
   30 D(1)=-H(2)*(EM(1)+EM(2)/2.)/3.+(Y(2)-Y(1))/H(2)
      DO 31 KJ=2,NP
   31 D(KJ)=H(KJ)*(EM(KJ-1)/2.+EM(KJ))/3.+(Y(KJ)-Y(KJ-1))/H(KJ)
      DO 32 KJ=1,NP
      DL = ATAN2(AIMAG(D(KJ)),REAL(D(KJ)))
      E(KJ) = CMPLX(DL,0.)
   32 IF (REAL(E(KJ)).LT.0.) E(KJ) = E(KJ)+TPI
      DO 322 KK=2,NP
      IF((REAL(E(KK))-REAL(E(KK-1))).LT.(-3.14159265))E(KK)=E(KK)+TPI
  322 CONTINUE
      E(NP) = E(1) + TPI
      F(1) = CMPLX(0.,0.)
      DYDX = AIMAG(D(1))/REAL(D(1))
      DSDTK = SQRT(1.+DYDX*DYDX)*ABS(REAL(D(1)))
      DO 33 KJ=2,NP
      DSDTO=DSDTK
      DYDX = AIMAG(D(KJ))/REAL(D(KJ))
19927



             DYDX = AIMAG(D(KJ))/REAL(D(KJ))
      DSDTK = SQRT(1. + DYDX*DYDX)*ABS(REAL(D(KJ)))
      DL = ABS(X(KJ)-X(KJ-1))
   33 F(KJ) = F(KJ-1) + DL*(DSDTO+DSDTK)/2.
      RETURN
```

```
C
C  THIS SECTION (ENTERED WHEN IRTN = 4) CONVERTS THE NP PAIRS
C  OF INPUT COORDINATES X AND Y TO INTRINSIC COORDINATES
C  BETA AND S, WHICH ARE STORED AS:
C     BETA = REAL(E), S = REAL(F)
C
   40 CONTINUE
      DO 41 KJ = 2,NP
   41 H(KJ) = X(KJ) - X(KJ-1)
      H(N) = H(2)
      DO 42 KJ = 2,NP
   42 BDA(KJ) = H(KJ+1)/(H(KJ)+H(KJ+1))
      E(1) = CMPLX(0.,0.)
      F(1) = CMPLX(0.,0.)
      S(1) = CMPLX(1.,0.)
      DO 43 KJ = 2,NPM
      DN = 2. + BDA(KJ)*E(KJ-1)
      E(KJ) = (BDA(KJ) - CMPLX(1.,0.))/DN
      D(KJ) = 3.*(BDA(KJ)*(Y(KJ)-Y(KJ-1))/H(KJ)
     *    + (CMPLX(1.,0.) - BDA(KJ))*(Y(KJ+1)-Y(KJ))/H(KJ+1))
      S(KJ) = - BDA(KJ)*S(KJ-1)/DN
   43 F(KJ) = (D(KJ) - BDA(KJ)*F(KJ-1))/DN
      Y(N) = Y(2)
      IF(NPD.EQ.2) Y(N) = Y(N) + TPI
      D(NP) = 3.*(BDA(NP)*(Y(NP)-Y(NPM))/H(NP)
     *    + (CMPLX(1.,0.)-BDA(NP))*(Y(2)-Y(NP))/H(2))
      NPMM = NP - 2
      T(NP) = CMPLX(1.,0.)
      V(NP) = CMPLX(0.,0.)
      DO 46 I = 1,NPMM
      KJ = NP - I
      T(KJ) = E(KJ)*T(KJ+1) + S(KJ)
      V(KJ) = E(KJ)*V(KJ+1) + F(KJ)
   45 CONTINUE
      EM(NP) = (D(NP)+(BDA(NP)-CMPLX(1.,0.))*V(2)-BDA(NP)
     *    *V(NPM))/((CMPLX(1.,0.)-BDA(NP))*T(2)+BDA(NP)*T(NPM)
     *    + CMPLX(2.,0.))
      DO 44 I = 1,NPM
      KJ = NP - I
   44 EM(KJ) = E(KJ)*EM(KJ+1) + F(KJ) + S(KJ)*EM(NP)
      SIZE = ABS(X(NP)-X(1))/10.
      DO 47 KJ = 1,NP
   47 D(KJ) = EM(KJ)
      DO 48 KJ = 1,NP
      DL = ATAN2(AIMAG(D(KJ)),REAL(D(KJ)))
      E(KJ) = CMPLX(DL,0.)
   48 IF(REAL(E(KJ)).LT.0.) E(KJ) = E(KJ) + TPI
      DO 422 KK = 2,NP
      IF((REAL(E(KK))-REAL(E(KK-1))).LT.-3.14159265)E(KK)=E(KK)+TPI
  422 CONTINUE
      E(NP) = E(1) + TPI
      F(1) = CMPLX(0.,0.)
```

```
      DYDX = AIMAG(D(1))/REAL(D(1))
      DSDTK = SQRT(1.+DYDX*DYDX)*ABS(REAL(D(1)))
      DO 433 KJ = 2,NP
      DSDTO = DSDTK
      DYDX = AIMAG(D(KJ))/REAL(D(KJ))
      DSDTK = SQRT(1.+DYDX*DYDX)*ABS(REAL(D(KJ)))
      DL = ABS(X(KJ)-X(KJ-1))
  433 F(KJ) = F(KJ-1) + DL*(DSDTO+DSDTK)/2.
      RETURN
      END




      SUBROUTINE OMZTDR(DR,DI,ZETAB,ZTA,ZOMA,KMX,LMX,IOE)
      IMPLICIT REAL(A-H,O-Y),COMPLEX(Z)
C
C  THIS SUBROUTINE INTEGRATES D(SMALL OMEGA)/D(ZETA) BY THE
C  TRAPEZOIDAL RULE TO FIND THE GRID IN THE SMALL-OMEGA PLANE.
C  IT MAKES SOME SMALL ADJUSTMENTS IN THE VALUES OF THE
C  DERIVATIVES, SO AS TO MAKE THE RESULTS PERIODIC.
C
      DIMENSION ZTA(10,40),ZDV(10,40),ZSTR(10),ZOMA(10,40)
      DIMENSION DR(65),DI(65)
      Z1 = CMPLX(1.,0.)
      DO 10 K = 1,KMX
      DO 20 L = 1,LMX
      ZTAGS = ZTA(L,K)
      CALL OMDZETA(Z1,Z1,DR,DI,Z1,ZTAGS,ZTANSR,65,1.,2,1)
      ZDV(L,K) = ZTANSR
   20 CONTINUE
   10 CONTINUE
      KMXH = (KMX+1)/2
      KM = KMXH - 1
      KMXP = KMX + 1
      KP = KMXH + 1
C
C  INTEGRATE FIRST ALONG THE PERIODIC BOUNDARY:
C
      KASE = 1
   27 IF(IOE.EQ.1) GO TO 11
C
C  IF IOE EQUALS 0/1, SMALL OMEGA = -1. IS NOT/IS A GRID POINT
C
      CALL OMDZETA(Z1,Z1,DR,DI,Z1,ZETAB,ZTANSR,65,1.,2,1)
      IF(KASE.EQ.2) ZTANSR = ZR*ZTANSR
      ZOMA(LMX,KMXH) = -Z1+0.5*(ZTANSR+ZDV(LMX,KMXH))*(ZTA(LMX,KMXH)
     *    - ZETAB)
      GO TO 12
   11 ZOMA(LMX,KMXH) = -Z1
   12 CONTINUE
      DO 25 J = 1,KM
      K = KMXH - J
```

```
25 ZOMA(LMX,K) = ZOMA(LMX,K+1)+0.5*(ZDV(LMX,K)+ZDV(LMX,K+1))
  *    *(ZTA(LMX,K)-ZTA(LMX,K+1))
      IF(KASE.EQ.2) GO TO 28
      ZR = 2.*Z1/(ZOMA(LMX,1)+Z1)
      DO 26 K = 1,KMXH
26 ZDV(LMX,K) = ZR*ZDV(LMX,K)
      KASE = 2
      GO TO 27
28 CONTINUE
      DO 29 I = 1,KMXH
29 ZOMA(LMX,KMXP-I) = ZOMA(LMX,I)
C
C   NOW INTEGRATE FROM THE BLADE SURFACE TO THE PERIODIC BOUNDARY:
C
      DO 40 K = 1,KMX
      ZSTR(1) = ZOMA(1,K)
      DO 41 L = 2,LMX
41 ZSTR(L) = ZSTR(L-1)+0.5*(ZDV(L,K)+ZDV(L-1,K))*
  *    (ZTA(L,K)-ZTA(L-1,K))
      ZR = (ZOMA(LMX,K) - ZOMA(1,K))/(ZSTR(LMX) - ZOMA(1,K))
      DO 42 L = 1,LMX
42 ZDV(L,K) = ZR*ZDV(L,K)
      DO 43 L = 2,LMX
43 ZOMA(L,K) = ZOMA(L-1,K)+0.5*(ZDV(L,K)+ZDV(L-1,K))*
  *    (ZTA(L,K)-ZTA(L-1,K))
40 CONTINUE
      RETURN
      END
```

Appendix C
DICTIONARY OF VARIABLES

FORTRAN SYMBOL	ALGEBRAIC EQUIVALENT	DEFINITION, USE, COMMENTS
A(J), B(J)	A_j , B_j	Eq. 2-20, 2-21
AK	$k = S^2$	Eq. 2-23
AKP	$k' = \sqrt{1-k^2}$	Eq. 2-30
AKQ	k^2	-
ALM	λ	Eq. 2-31
CAPK	$K(\bar{k})$	Eq. 2-34
CAPKPM	$K'(k')$	Eq. 2-34
DLT	δ	Eq. 2-28
DXIM, DXIR	$Re(\Delta\hat{\xi}), Im(\Delta\hat{\xi})$	Eq. 2-38, 4-4
EX	$1/(2 - \frac{\tau}{\pi})$	See Figure 2
EXINV	$2 - \frac{\tau}{\pi}$	-
G, H	G , H	See Figure 1
HCPKPM	$\frac{1}{2} K'(k')$	-
OM	—	Relaxation factor used in ϕ , θ mapping
PBN	$\frac{\pi}{N}$	-
PHI	ϕ	-
SGA	σ	Eq. 2-31
SS	S	Eqs. 2-23
TAU	τ	Eq. 2-28

FORTRAN SYMBOL	ALGEBRAIC EQUIVALENT	DEFINITION, USE, COMMENTS
TPI	2π	-
XIR, XIM	$Re(\hat{\xi}), Im(\hat{\xi})$	-
ZA(N)	$A(n)$	Eq. A-9; used elsewhere for temporary storage
ZB	b	Eq. 2-9
ZC	c	Eq. 2-9
ZD	a	Eq. 2-9
ZI	$\sqrt{-1}$	-
ZN, ZT, ZLE, ZTE	z_N, z_T	See Figure 1
ZAL, ZBT, ZGM	α, β, γ	Eq. 2-22
ZA1(N), ZA2(N), ZCC(N)	$A_1(n), A_2(n), c(n)$	See, for example, Eq. A-9. Also used for temporary storage
ZBOM	Ω	-
ZBOMK	Ω^K	
ZEETA	η	-
ZETA	ζ	-
ZNTRD		centroid of the ω-plane
ZØMSTR		Ω - plane image of ZNTRO
ZOMS(K), ZOMP(K)	ω_s, ω_p	-
ZPLUS, ZMINUS	Ω^+, Ω^-	Eq. 1-6
ZXI	$\hat{\xi}$	-
ZXY		$x+iy$

Appendix D

LISTING OF METRIC GENERATOR PROGRAM

```
LEVEL 21.7 ( DEC 72 )                                   OS/360  FORTRAN H

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                                    SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,XREF
                     C
                     C  PROGRAM CIMTVL - A PROGRAM TO CALCULATE THE METRICS OF A COORDINATE
                     C  TRANSFORMATION FOR TURBOMACHINERY CASCADES.  FOR DOCUMENTATION,SEE:
                     C
                     C      J. P. NENNI AND W. J. RAE, EXPERIENCE WITH THE DEVELOPMENT OF
                     C      AN EULER CODE FOR ROTOR ROWS, ASME PAPER 83-GT-36, MARCH 1983
                     C      W. J. RAE, A COMPUTER PROGRAM FOR THE IVES TRANSFORMATION IN
                     C      TURBOMACHINERY CASCADES, AFOSR TR-81-0154, ADA096416, NOV 1980
                     C      W. J. RAE, MODIFICATIONS OF THE IVES-LIUTERMOZA CONFORMAL-
                     C      MAPPING PROCEDURE FOR TURBOMACHINERY CASCADES, ASME PAPER
                     C      83-GT-116, MARCH 1983
                     C      W.J. RAE, REVISED COMPUTER PROGRAM FOR EVALUATING THE IVES
                     C      TRANSFORMATION IN TURBOMACHINERY CASCADES, CALSPAN CORPORATION
                     C      REPORT 7177-A-1, JULY 1983
                     C
                     C  THE PROGRAM READS A CARD DECK CONTAINING THE COORDINATES X(K,L) ,
                     C  Y(K,L) ; K = 1,KMX ; L = 1,LMX, AND FINDS BY FINITE DIFFERENCES
                     C  THE METRICS OF A TRANSFORMATION TO A RECTANGLE(KSI,ETA).  IN THIS
                     C  RECTANGLE, THE IMAGE OF THE BLADE SURFACE LIES ALONG ONE SIDE, AND
                     C  THE IMAGES OF THE POINTS AT INFINITY LIE AT CORNERS OF THE RECTANGLE
                     C  (SEE THE REFERENCES ABOVE). THE METRICS ARE WRITTEN ON TAPE 1.
                     C
   ISN 0002                 IMPLICIT REAL*8(A-H,O-Z)
   ISN 0003                 DIMENSION Q(410,6)
   ISN 0004                 READ(5,100) KMX,LMX
   ISN 0005             100 FORMAT(20I4)
   ISN 0006                 READ(5,101) SG
   ISN 0007             101 FORMAT(8F10.4)
                     C
                     C  SG IS THE SLANT GAP BETWEEN BLADES
                     C
   ISN 0008                 DO 10 K = 1,KMX
   ISN 0009                 KM1LMX = (K-1)*LMX
   ISN 0010                 LKA = KM1LMX + 1
   ISN 0011                 LKB = KM1LMX + LMX
   ISN 0012              10 READ(5,200)((Q(LK,5),Q(LK,6)),LK=LKA,LKB)
   ISN 0013             200 FORMAT(1P4E20.13)
                     C
                     C  Q(LK,5) CONTAINS X(K,L), Q(LK,6) CONTAINS Y(K,L)
                     C
   ISN 0014                 KM=KMX-1
   ISN 0015                 LM=LMX-1
                     C
                     C --- SET X AND Y AT K=KMX AND L=1
                     C
   ISN 0016                 LK=KM*LMX+1
   ISN 0017                 Q(LK,5)=Q(1,5)
   ISN 0018                 Q(LK,6)=Q(1,6)
                     C
                     C --- FIND X AND Y AT K=1 AND L=LMX
                     C
   ISN 0019                 Q(LMX,5)=2D0*Q(LM,5)-Q(LMX-2,5)
   ISN 0020                 Q(LMX,6)=2D0*Q(LM,6)-Q(LMX-2,6)
                     C
                     C --- FIND X AND Y AT K=KMX AND L=LMX
                     C
```

```
ISN 0021          LK=KMX*LMX
ISN 0022          Q(LK,5)=2D0*Q(LK-1,5)-Q(LK-2,5)
ISN 0023          Q(LK,6)=2D0*Q(LK-1,5)-Q(LK-2,6)
ISN 0024          TDELXI=4D0/DFLOAT(KM)
ISN 0025          TDELZT=2D0/DFLOAT(LM)
            C
ISN 0026          DO 520 K=1,KMX
ISN 0027          KM1LMX=(K-1)*LMX
            C
ISN 0028          DO 510 L=1,LMX
ISN 0029          LK=KM1LMX+L
ISN 0030          LP1=LK+1
ISN 0031          LM1=LK-1
ISN 0032          KP1=LK+LMX
ISN 0033          KM1=LK-LMX
            C
ISN 0034          IF(K.EQ.1) GO TO 501
ISN 0036          IF(K.EQ.KMX) GO TO 501
ISN 0038          IF(L.EQ.1) GO TO 504
ISN 0040          IF(L.EQ.LMX) GO TO 505
            C
            C --- K=2 TO KM AND L=2 TO LM    (CENTERED DIFFERENCES)
            C
ISN 0042          DXDXI=(Q(KP1,5)-Q(KM1,5))/TDELXI
ISN 0043          DYDXI=(Q(KP1,6)-Q(KM1,6))/TDELXI
ISN 0044          DXDZT=(Q(LP1,5)-Q(LM1,5))/TDELZT
ISN 0045          DYDZT=(Q(LP1,6)-Q(LM1,6))/TDELZT
ISN 0046          GO TO 509
            C
ISN 0047      501 IF(L.EQ.1) GO TO 510
ISN 0049          IF(L.EQ.LMX) GO TO 510
            C
            C --- K=1 OR KMX AND L=2 TO LM
            C
ISN 0051          LK2=LMX+L
ISN 0052          LKKM=(KM-1)*LMX+L
ISN 0053          DXDXI=(Q(LK2,5)-Q(LKKM,5))/TDELXI
ISN 0054          DYDXI=(Q(LK2,6)-Q(LKKM,6))/TDELXI
ISN 0055          DXDZT=(Q(LP1,5)-Q(LM1,5))/TDELZT
ISN 0056          DYDZT=(Q(LP1,5)-Q(LM1,6))/TDELZT
ISN 0057          GO TO 509
            C
            C --- L=1 AND K=2 TO KM
            C
ISN 0058      504 LK1=KM1LMX+1
ISN 0059          LK2=LK1+1
ISN 0060          LK3=LK2+1
ISN 0061          DXDZT=(-3D0*Q(LK1,5)+4D0*Q(LK2,5)-Q(LK3,5))/TDELZT
ISN 0062          DYDZT=(-3D0*Q(LK1,6)+4D0*Q(LK2,6)-Q(LK3,6))/TDELZT
ISN 0063          GO TO 506
            C
            C --- L=LMX AND K=2 TO KM
            C
ISN 0064      505 LKS=(KMX-K)*LMX+LM
ISN 0065          LKB=KM1LMX+LM
ISN 0066          DXDZT=(Q(LKS,5)-Q(LKB,5))/TDELZT
ISN 0067          DQ=SG
ISN 0068          IF(K.GT.KMXH) DQ=-SG
```

```
ISN 0070          DYDZT=(Q(LKS,6)+DQ-Q(LKB,6))/TDELZT
            C
ISN 0071      506 DXDXI=(Q(KP1,5)-Q(KM1,5))/TDELXI
ISN 0072          DYDXI=(Q(KP1,6)-Q(KM1,6))/TDELXI
            C
ISN 0073      509 RDM1=(DXDXI*DYDZT-DXDZT*DYDXI)
            C
            C  STORE DERIVATIVES IN THE ORDER DKSI/DX,DKSI/DY,DETA/DX,DETA/DY
            C
ISN 0074          Q(LK,1)=DYDZT/RDM1
ISN 0075          Q(LK,2)=-DXDZT/RDM1
ISN 0076          Q(LK,3)=-DYDXI/RDM1
ISN 0077          Q(LK,4)= DXDXI/RDM1
ISN 0078      510 CONTINUE
ISN 0079      520 CONTINUE
            C
            C  STORE DERIVATIVES ON TAPE 1
            C
ISN 0080          LKMX=LMX*KMX
            C     WRITE(1) KMX,LMX,LKMX,((Q(I,J),I=1,LKMX),J=1,4)
            C
            C
ISN 0087          STOP
ISN 0088          END
```

REFERENCES

1. Rae, W.J., "Revised Computer Program for Evaluating the Ives Transformation in Turbomachinery Cascades" AFOSR TR-83-1284 (July 1983).

2. Viviand, H., "Formes Conservatives des Equations de la Dynamique des Gaz" La Recherche Aerospatiale (1974) No. 1, Jan.-Feb. pp. 65-66.

3. Beam, R.M. and Warming, R.F., "An Implicit Finite - Difference Algorithm for Hyperbolic Systems in Conservation-Law Form" J. Comp. Phys. 22, (1976) 87-110.

4. Briley, W.R. and McDonald, H., "Solution of the Multidimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method" J. Comp. Phys. 24 (1977) 372-392.

5. Thompson, J.F., ed., Numerical Grid Generation, Elsevier Science Publishing Co., New York (1982).

6. Dulikravich, D.S., "Numerical Calculation of Inviscid, Potential Transonic Flows through Rotors and Fans" Ph.D. Thesis, Cornell University (Jan. 1979).

7. Nenni, J.P. and Rae, W.J., "Experience with the Development of an Euler Code for Rotor Rows" ASME Paper 83-GT-36 (Mar. 1983).

8. Ives, D.C. and Liutermoza, J.F., "Analysis of Transonic Cascade Flow Using Conformal Mapping and Relaxation Techniques" AIAA Journal 15 (1977) 647-652.

9. Rae, W.J., "A Computer Program for the Ives Transformation in Turbomachinery Cascades" AFOSR TR-81-0154 (Nov. 1980).

10. Rae, W.J., "Modification of the Ives-Liutermoza Conformal-Mapping Procedure for Turbomachinery Cascades" ASME J. Eng. GT Power 106 (Apr. 1984) 445-448.

11. Bauer, F., et al., "Supercritical Wing Sections II" Vol. 108 of Lecture Notes in Economics and Mathematical Systems, Springer Verlag, New York (1975).

12. Cooley, J.W., Lewis, P.A.W., and Welch, P.D., "The Fast Fourier Transform Algorithm: Programmimg Considerations in the Calculations of Sine, Cosine and Laplace Transforms", Journal of Sound and Vibration 12, (1970) pp. 315-337.

13. Warschawski, S.E., "On Theodorsen's Method of Conformal Mapping of Nearly Circular Regions", Quarterly of Applied Mathematics 3, (1945) pp. 12-28.

14. Henrici, P., "Fast Fourier Methods in Computational Complex Analysis", SIAM Review 21, (1979) pp. 481-527.

15. Mokry, M., "Comment on Analysis of Transonic Cascade Flow Using Conformal Mapping and Relaxation Techniques", AIAA Journal <u>16</u>, No. 1, (January 1978) p. 96.

16. Nielsen, J.N. and Perkins, E.W., "Charts for the Conical Part of the Downwash Field of Swept Wings at Supersonic Speeds", NACA Technical Note 1780, (December 1948), Appendix C.

17. Byrd, P.F., and Friedman, M.D., <u>Handbook of Elliptic Integrals for Engineers and Physicists</u>. Springer Verlag, Berlin (1954).

18. Luke, Y.L., "Approximations for Elliptic Integrals", Mathematics of Computation, <u>22</u> (1968) 627-634.

19. Erdelyi, A., et al. <u>Higher Transcendental Functions</u>, Volume 2, p. 377, McGray-Hill Book Company, New York (1953).

20. Abramowitz, M. and Stegun, I.S., Handbook of Mathematical Functions, National Bureau of Standards, Applied Mathematics Series 55 (1964).

21. Oates, G.C., "Cascade Flows" Chapter 12 of <u>The Aerothermodynamics of Aircraft Gas Turbine Engines</u>, G.C. Oates, ed., AFAPL-TR-78-52, (July 1978).

22. Davis, R.T., "Numerical Methods for Coordinate Generation Based on Schwarz-Christoffel Transformations" AIAA Paper 79-1463 (July 1979).

23. W. Squire, "Computer Implementation of the Schwarz-Christoffel Transformation: Journal of the Franklin Institute, <u>299</u> No. 5 (May 1975)

24. O. L. Anderson et al. "Solution of Viscous Internal Flows on Curvilinear Grids Generated by the Schwarz-Christoffel Transformation" pp. 507-524 of Ref. 5

25. K. P. Sridhar & R. T. Davis, "A Schwarz-Christoffel Method for Generating Internal Flow Grids" AIAA Paper, A82-29005 (1982)

26. C. Farrell & J. J. Adamczyk, "Full Potential Solution of Transonic Quasi-Three-Dimensional Flow Through a Cascade Using Artificial Compressibility" ASME J. Eng. Power <u>104</u> (1982) 143-153

27. Erdelyi, A., et al. <u>Higher Transcendental Functions</u> Vol. 1, Chapter 2, McGraw-Hill Book Company, New York (1953)

28. Kober, H., Dictionary of Conformal Transformations, Dover Publications New York (1957)

29. Ahlberg, J.H., Nilson, E.N., and Walsh, J.L. <u>The Theory of Splines and Their Applications</u> Academic Press, New York (1967)

30. Hartree, D.R. <u>Numerical Analysis</u>, 2nd Edition, Oxford, Clarnedon (Press (1958)

END

FILMED

3-86

DTIC