

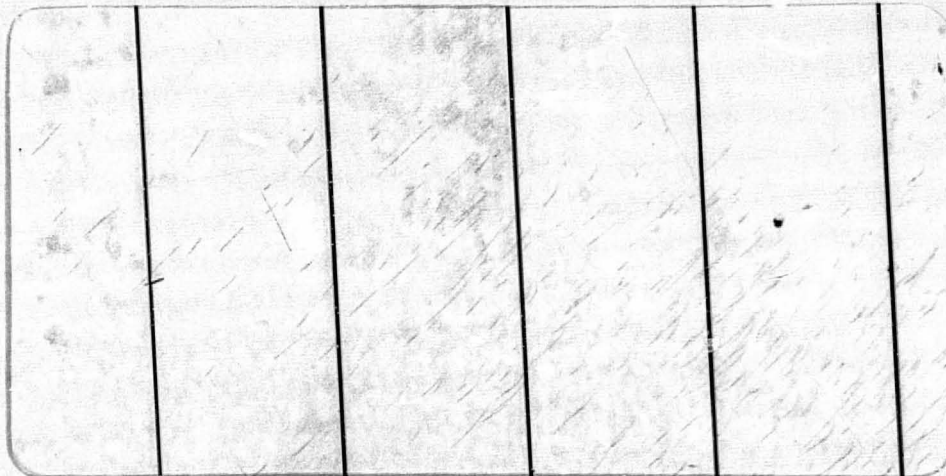
General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

AAE 

AERONAUTICAL AND ASTRONAUTICAL ENGINEERING DEPARTMENT



ENGINEERING EXPERIMENT STATION, COLLEGE OF ENGINEERING, UNIVERSITY OF ILLINOIS, URBANA

Aeronautical and Astronautical Engineering Department
University of Illinois Urbana, Illinois

Technical Report AAE 75-4
UIIU-Eng 75 0504

NASA Grant NGR 14-005-144
Allen I. Ormsbee, Principal Investigator

PROGRAM MANUAL
FOR THE
EPPLER AIRFOIL INVERSION PROGRAM
by
William G. Thomson

University of Illinois
Urbana, Illinois
May 1975

PREFACE

This manual presents a brief discussion of the theory and a detailed description of the computer program for an airfoil inversion technique developed by Eppler. The program represents revision of a deck supplied to us by Dr. Stanley Miley. Contributions by Dr. T. E. Edwards and by Mr. R. H. Awker are also gratefully acknowledged. While the program has been set up specifically for the University of Illinois IBM 360/75 computer, its modification to other computers should be straightforward.

This work was funded by NASA under grant NGR 14-005-144 as part of a low speed airfoil study.

I. THEORY

The method developed by Eppler^{1,2} is an inverse conformal mapping technique that determines the x and y coordinates from a given velocity distribution. The two planes involved are shown in figure 1. The ζ plane shows the flow about a circular cylinder, while the z plane represents the flow about the airfoil. The velocity in the z plane is given in terms of coordinates determined in the ζ plane. z and ζ are defined as:

$$z = x + iy \quad (1)$$

$$\zeta = \xi + i\eta = re^{i\phi} \quad (2)$$

The flow in the ζ plane is such that the rear stagnation point falls on the real axis at $\zeta = 1$.

There exists a transformation of the ζ plane to the z plane such that the z plane represents parallel flow about a closed airfoil at an angle of attack α . Since $\zeta = 1$ represents a stagnation point, the Kutta condition requires that this must transform to the trailing edge of the airfoil. As this is to be an infinite parallel flow, $z(\infty) = \infty$ and $(\frac{dz}{d\zeta})_{\infty}$ must be real. The general function that satisfies these requirements is

$$z(\zeta) = \beta_1 \zeta + \sum_{v=0}^{\infty} \beta_{-v} \zeta^{-v} \quad (3)$$

where β_1 is real, but not equal to zero.

The complex potential in the ζ plane can be represented as

$$F(\zeta) = \Phi + i\Psi = C(e^{-i\alpha}\zeta + \frac{i\alpha}{\zeta}) - \frac{\Gamma}{2\pi i} \ln \zeta \quad (4)$$

where Γ is given by

$$\Gamma = 4\pi C \sin \alpha \quad (5)$$

The complex velocity in the z plane is given by

$$y = \left| \frac{dF}{dz} \right| = \left| \frac{dF/d\zeta}{dz/d\zeta} \right| \quad (6)$$

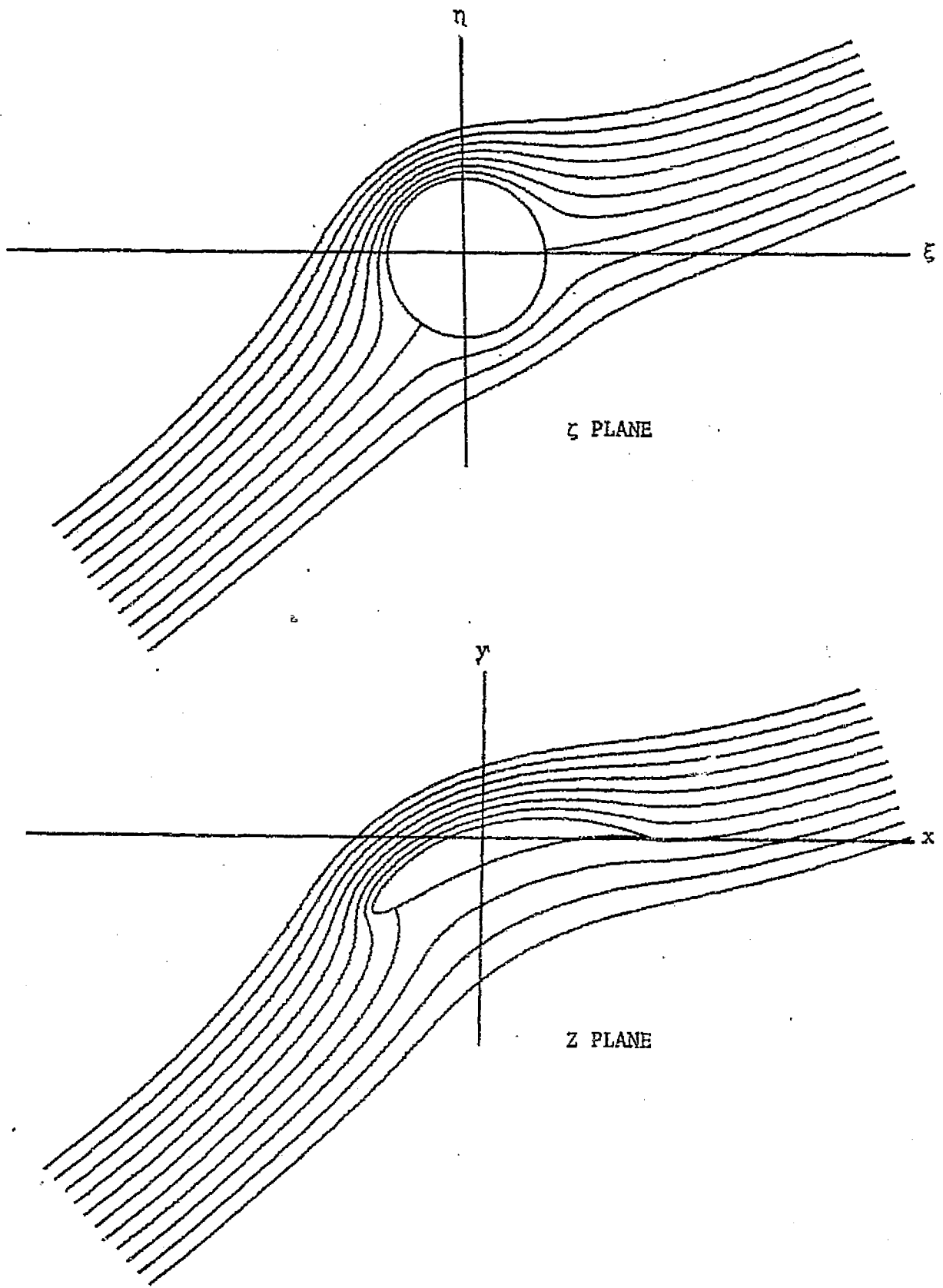


Figure 1. The complex mapping planes

The inverse of this will be used, or

$$\frac{dz}{dF} = \frac{dz/d\zeta}{dF/d\zeta} \quad (7)$$

In order to prevent an undesirable root, this can be written as

$$\ln \frac{dz}{dF} = \ln \frac{dz}{d\zeta} - \ln \frac{dF}{d\zeta} \quad (8)$$

The velocity vector in the z plane can be introduced as $V = Ve^{i\theta}$. Then

$$\frac{dF}{dz} = Ve^{-i\theta} \quad (9)$$

Therefore,

$$\ln \frac{dz}{dF} = - \ln V + i\theta. \quad (10)$$

The real part of $\ln \frac{dz}{dF}$ is then $-\ln V$. Outside the boundary of the unit circle, $\ln \frac{dz}{dF}$ is regular, and can be calculated when the real part is known on the boundary. Since $F(\zeta)$ is also known (eq (4)), equation (8) can be solved for $\frac{dz}{d\zeta}$. $z(\zeta)$ then can be found by integration. $z(\zeta)$ must be of the form of equation (3) to match the boundary conditions. Therefore, the problem is to find an equation for $\ln \frac{dz}{dF}$ which results in a satisfactory $\frac{dz}{d\zeta}$.

From equations (4) and (5)

$$\frac{dF}{d\zeta} = -e^{i\alpha} C \left(\frac{1}{\zeta} - 1 \right) \left(\frac{1}{\zeta} + e^{-2i\alpha} \right). \quad (11)$$

In light of the singularities involved at the stagnation points, $\ln \frac{dz}{dF}$ can be represented by

$$\ln \frac{dz}{dF} = - \ln C - \ln \left[e^{i\alpha} \left(\frac{1}{\zeta} + e^{-2i\alpha} \right) \right] + \sum_{m=0}^{\infty} (a_m + ib_m) \zeta^{-m} \quad (12)$$

Using this equation and equations (8) and (11),

$$\ln \frac{dz}{d\zeta} = \ln \left(1 - \frac{1}{\zeta} \right) + \sum_{m=0}^{\infty} (a_m + ib_m) \zeta^{-m} \quad (13)$$

This results in

$$\frac{dz}{d\zeta} = \left(1 - \frac{1}{\zeta}\right) e^{\sum_{m=0}^{\infty} (a_m + ib_m) \zeta^{-m}} \quad (14)$$

Expanding this yields

$$\frac{dz}{d\zeta} = \left(1 - \frac{1}{\zeta}\right) e^{a_0 + ib_0} e^{(a_1 + ib_1) \zeta^{-1}} e^{(a_2 + ib_2) \zeta^{-2}} \dots \quad (15)$$

$z(\zeta)$ must be of the form of equation (3), so $\frac{dz}{d\zeta}$ must be of the form

$$\frac{dz}{d\zeta} = \beta_1 + \sum_{v=1}^{\infty} -v\beta_{-v} \zeta^{-v-1} \quad (16)$$

If we let $A_n = a_n + ib_n$, equation (15) can be further expanded to

$$\begin{aligned} \frac{dz}{d\zeta} = & \left(1 - \frac{1}{\zeta}\right) \left(1 + A_0 + \frac{A_0^2}{2!} + \frac{A_0^3}{3!} + \dots\right) \left(1 + \frac{A_1}{\zeta} + \frac{A_1^2}{2!\zeta^2} + \frac{A_1^3}{3!\zeta^3} + \dots\right) \left(1 + \frac{A_2}{\zeta^2} + \right. \\ & \left. \frac{A_2^2}{2!\zeta^4} + \dots\right) \dots \end{aligned} \quad (17)$$

Comparing equations (16) and (17) yields

$$\beta_1 = 1 + A_0 + \frac{A_0^2}{2!} + \frac{A_0^3}{3!} + \dots = e^{A_0} \quad (18)$$

Since β_1 is real, A_0 is also real. At infinity, we want $\frac{dz}{d\zeta} = 1$, indicating that the flow at infinity in the circle plane is in the same direction as the flow at infinity in the airfoil plane. From equation (16), $\frac{dz}{d\zeta} = \beta_1$ at infinity. Therefore,

$$\beta_1 = e^{A_0} = 1 \quad (19)$$

Because of this, $A_0 = 0$.

Comparing the $\frac{1}{\zeta}$ term yields

$$-\frac{\beta_1}{\zeta} + \frac{A_1 \beta_1}{\zeta} = 0 \quad (20)$$

So $a_1 = 1$, $b_1 = 0$.

At infinity, we can arbitrarily set the velocity equal to unity. From equation (12), in the limit as $\zeta \rightarrow \infty$, $\frac{dF}{dz} = C e^{-i\alpha}$. Therefore, $C=1$ and $\ln C=0$.

The problem that remains is to define the a_m and b_m that have not yet been defined such that, along the surface of the airfoil, the velocity assumes the prescribed values.

Along the surface of the circle plane, $\zeta = e^{i\phi}$. Using this, equation (12) results in

$$\ln \frac{dz}{dF} = -\ln [e^{i\alpha} (e^{i\phi} + e^{2i\alpha})] + \sum_{m=0}^{\infty} (a_m + ib_m) e^{-im\phi} \quad (21)$$

If we use the substitutions

$$P(\phi) = \sum_{m=0}^{\infty} (a_m \cos m\phi + b_m \sin m\phi) \quad (22)$$

$$Q(\phi) = \sum_{m=0}^{\infty} (-a_m \sin m\phi + b_m \cos m\phi) \quad (23)$$

equation (21) can be written as

$$\begin{aligned} \ln \frac{dz}{dF} = \ln [e^{i\phi/2} (e^{i(\alpha + \phi/2)} + e^{-i(\alpha + \phi/2)})] + P(\phi) \\ + iQ(\phi) \end{aligned} \quad (24)$$

This can be written as

$$\begin{aligned} -\ln V(\phi) + i\theta(\phi) = -\ln 2 |\cos(\phi/2 - \alpha)| + P(\phi) + i\left[\frac{\phi}{2} + Q(\phi)\right] \\ - \{\{\pi\}\} \end{aligned} \quad (25)$$

where $\{\{\pi\}\}$ is given as

$$\{\{\pi\}\} = \begin{cases} 0 & (0 \leq \phi < \pi + 2\alpha) \\ \pi & (\pi + 2\alpha < \phi \leq 2\pi) \end{cases} \quad (26)$$

The $\{\pi\}$ term is necessary due to the shift in the direction of the velocity at the stagnation point.

The real part of equation (25) can be rearranged into the form

$$P(\phi) = \ln 2 \left| \cos \left(\frac{\phi}{2} - \alpha \right) \right| - \ln V(\phi). \quad (27)$$

Through harmonic analysis, the a_m 's and b_m 's can be determined from equation (27). However, we must have $a_0 = 0$, $a_1 = 1$, and $b_1 = 0$, due to equations (19) and (20). Therefore,

$$\oint_0^{2\pi} P(\phi) d\phi = 0 \quad (28)$$

$$\oint_0^{2\pi} P(\phi) \cos \phi d\phi = \pi \quad (29)$$

$$\oint_0^{2\pi} P(\phi) \sin \phi d\phi = 0 \quad (30)$$

The velocity distribution we specify must meet these requirements.

By integrating equation (14), the transformation

$$z(\zeta) = \int (1 - 1/\zeta) e^{\sum_{m=0}^{\infty} (a_m + ib_m) \zeta^{-m}} d\zeta \quad (31)$$

can be derived. This yields the flow for the entire z plane. If $\zeta = e^{i\phi}$ is entered into this transformation, the resulting $z = x + iy$ will yield the profile of the airfoil. The results are

$$x(\phi) = \int -4 \sin \frac{\phi}{2} \left| \cos \left(\frac{\phi}{2} - \alpha \right) \right| \frac{1}{V(\phi)} \cos \left(\frac{\phi}{2} + Q(\phi) \right) d\phi \quad (32)$$

$$y(\phi) = \int -4 \sin \frac{\phi}{2} \left| \cos \left(\frac{\phi}{2} - \alpha \right) \right| \frac{1}{V(\phi)} \sin \left(\frac{\phi}{2} + Q(\phi) \right) d\phi. \quad (33)$$

The only quantity remaining to be defined, then, is $Q(\phi)$. However $Q(\phi)$ is a conjugate harmonic function of $P(\phi)$, and can be derived from the formula

$$Q(\phi) = \frac{1}{2\pi} \int_0^{2\pi} P(\sigma) \cot \left(\frac{\sigma - \phi}{2} \right) d\sigma \quad (34)$$

Given a velocity distribution that yields a $P(\phi)$ such that equations (28) through (30) are satisfied, and an angle of attack, the x and y coordinates of the desired airfoil can be generated using equations (32) and (33). The angle of attack, α , need not be held constant, but can be a function of ϕ . Thus, the upper surface can be designed at a different (higher for reasonable airfoils) angle of attack than the lower surface, or even different portions of the upper (or lower) surface can be designed at different angles of attack.

The profile of the airfoil is determined by a_m and b_m . Therefore, for a fixed profile, a_m and b_m are fixed. Altering the angle of attack will not alter the airfoil profile, and, therefore, will not alter a_m or b_m . This means that $P(\phi)$ is independent of the angle of attack. Equation (27) can, then, be written in the form of

$$P(\phi) = \ln 2 \left| \cos \left(\frac{\phi}{2} - \alpha^*(\phi) \right) \right| - \ln V^*(\phi) \quad (35)$$

where $V^*(\phi)$ is the specified velocity at the point on the airfoil corresponding to ϕ , and $\alpha^*(\phi)$ is the corresponding angle of attack. At any angle of attack, α , then, the velocity can be given as

$$V(\phi, \alpha) = \frac{\cos(\phi/2 - \alpha)}{\cos(\phi/2 - \alpha^*(\phi))} V^*(\phi) \quad (36)$$

The circle plane can be divided into I_a segments, as in figure (2), where $\phi_0 = 0 < \phi_1 < \phi_2 < \dots < \phi_{I_a} < \phi_{I_L}$. ϕ_{I_a} indicates the stagnation point. The angle of attack specification takes the form

$$\alpha^*(\phi) = \alpha_i = \text{constant for } \phi_{i-1} < \phi \leq \phi_i \quad (37)$$

and the velocity takes the form

$$V^*(\phi) = V_i W_i(\phi) \quad (38)$$

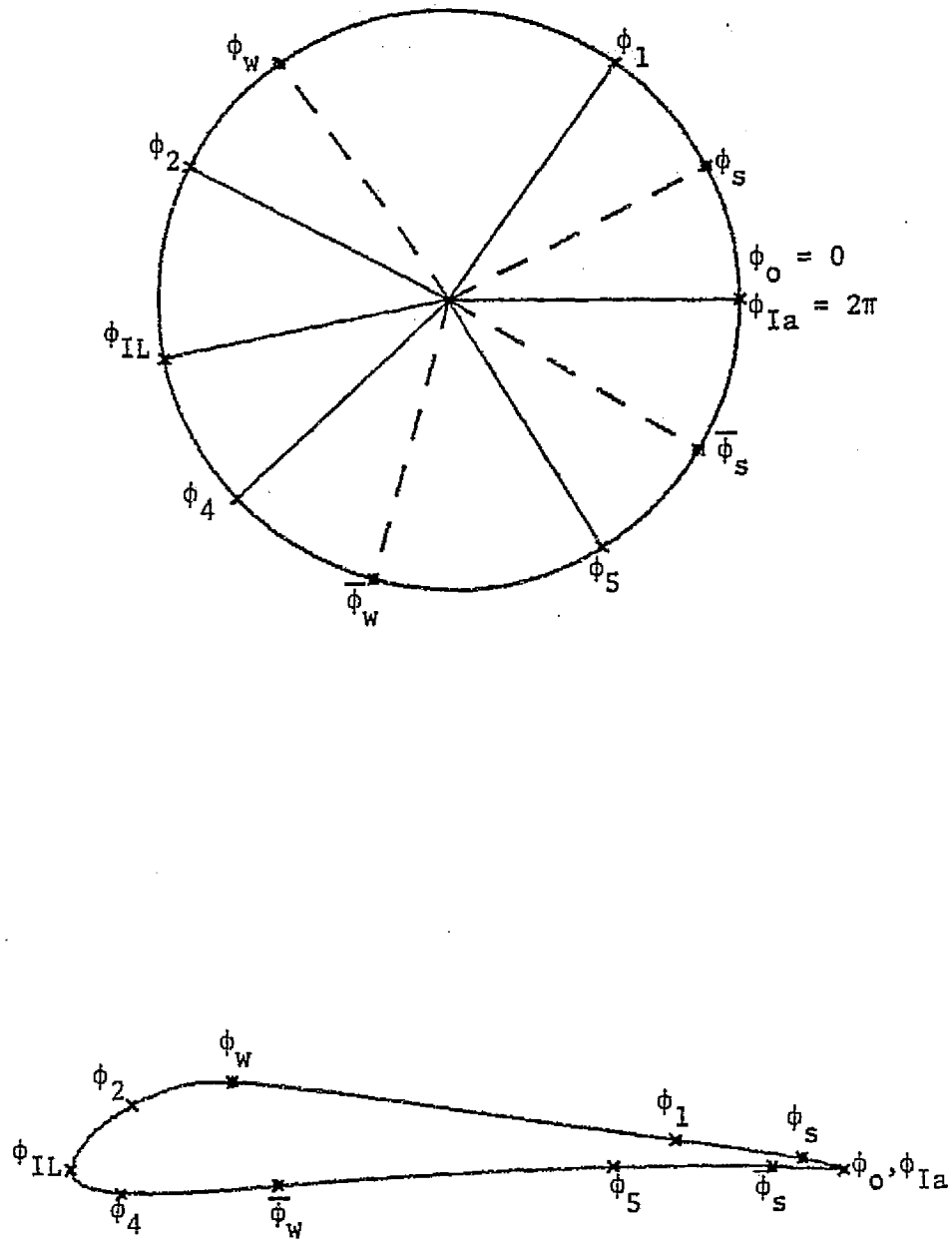


Figure 2. Segementing the profile

where V_i is a constant for $\phi_{i-1} < \phi \leq \phi_i$ and $W(\phi)$ is given as

$$W(\phi) = [1 + K \left\{ \left\{ \frac{\cos \phi - \cos \phi_\omega}{1 + \cos \phi_\omega} \right\} \right\}^{-\mu} [1 - 0.36 \left\{ \left\{ \frac{\cos \phi - \cos \phi_s}{1 - \cos \phi_s} \right\} \right\}^2]^{K_H} \quad (39)$$

on the upper surface. On the lower surface, the velocity distribution is similar, but different values of K_H , μ , ϕ_w , and ϕ_s are used, indicated by \bar{K}_H , $\bar{\mu}$, $\bar{\phi}_w$, and $\bar{\phi}_s$, respectively. The terms in the double brackets of equation (38) are defined by

$$\{\{f(\phi)\}\} = \begin{cases} f(\phi) & f(\phi) > 0 \\ 0 & f(\phi) \leq 0 \end{cases} \quad (40)$$

Equation (39) can be considered to be of the form

$$W(\phi) = W_w(\phi) W_s(\phi)^{K_H} \quad (41)$$

or, on the lower surface

$$W(\phi) = \bar{W}_w(\phi) \bar{W}_s(\phi)^{\bar{K}_H} \quad (42)$$

The W_w (or \bar{W}_w) term produces the major pressure recovery. This is in the form specified by Wortmann,⁹ in which the shape parameter is held constant, which delays separation. The W_s (or \bar{W}_s) term develops the cusp distribution. It is generally applied to the last 3-5% of the airfoil length. Outside of the range of the specified region, $\phi > \phi_w$ for W_w or $\phi > \phi_s$ for W_s (or $\phi < \bar{\phi}_w$ or $\bar{\phi}_s$ on the lower surface), the value of W_w or W_s is one, as dictated by equation (40). Thus, the variation of W_w and W_s is as is given by figure (3).

$P(\phi)$ must be continuous over the airfoil, so $P(\phi_i^-) = P(\phi_i^+)$, and $P(0) = P(2\pi)$.

Substituting our values for V^* and α^* into equation (27) yields

$$P(\phi) = \ln 2 \left| \cos \left(\frac{\phi}{2} - \alpha(\phi) \right) \right| - \ln [V_1 W_w(\phi) W_s(\phi)^{K_H}] \quad (43)$$

Since, at the trailing edge, $P(0) = P(2\pi)$,

$$\ln 2 |\cos \alpha_1| - \ln [V_1 W_w(0) W_s(0)^{K_H}] = \ln 2 |\cos \alpha_{I_a}| - \ln [V_{I_a} \bar{W}_w(2\pi)] \quad (44)$$

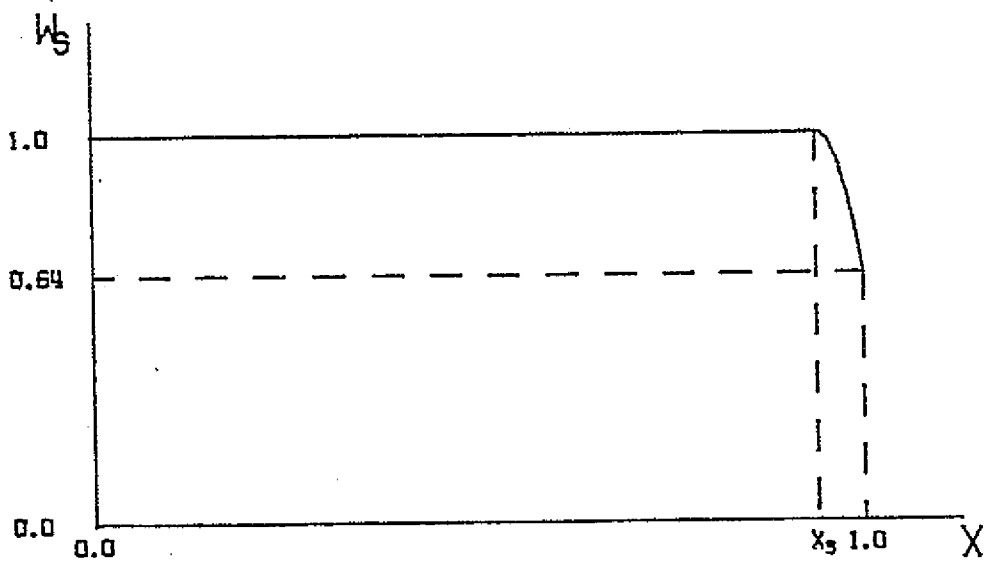
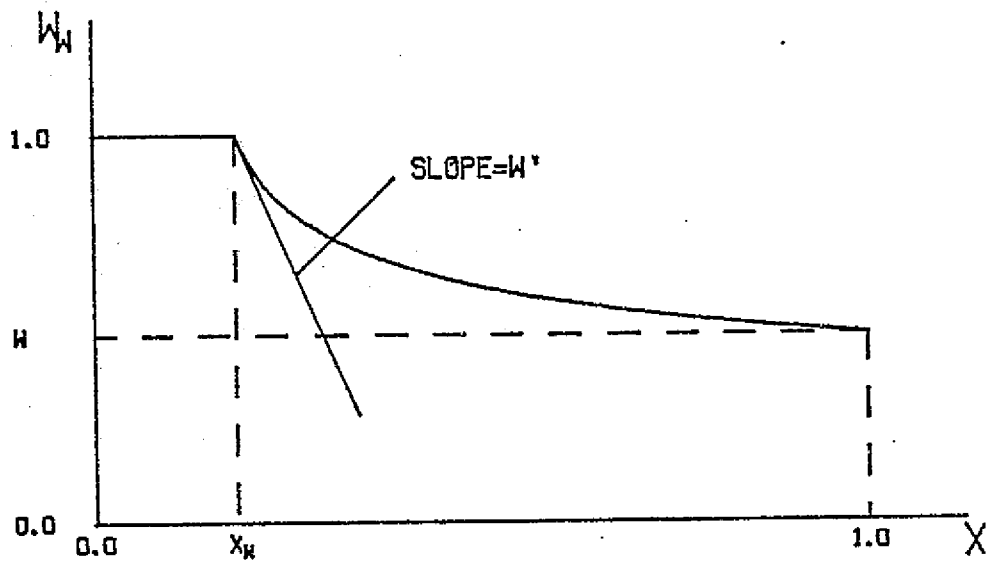


Figure 3. The form of the velocity components.

At all other segment boundaries, which are generally outside of the cusp region, $W_s = \bar{W}_s = 1, P(\phi_i^-) = P(\phi_i^+)$, and therefore

$$\ln 2 \left| \cos\left(\frac{\phi_i}{2} - \alpha_i\right) \right| - \ln[V_i W_w(\phi_i)] = \ln 2 \left| \cos\left(\frac{\phi_i}{2} - \alpha_{i+1}\right) \right| - \ln[V_{i+1} W_w(\phi_i)] \quad (45)$$

On the lower surface, W_w is replaced by \bar{W}_w in equation (45).

If, in any segment, $\phi = \pi + 2\alpha_i$, $P(\phi)$ becomes infinite. This is most likely to occur in the segments to either side of the stagnation point, $\phi_{I_L-1} < \phi < \phi_{I_L+1}$, where ϕ_{I_L} indicates the stagnation point. In order to prevent this, we require

$$\alpha_{I_L} > \alpha_{I_L+1} \quad (46)$$

and

$$\pi + 2\alpha_{I_L} > \phi_{I_L} > \pi + 2\alpha_{I_L+1} \quad (47)$$

If all the ϕ_i 's and α_i 's along with μ , $\bar{\mu}$, ϕ_w , $\bar{\phi}_w$, ϕ_s , and $\bar{\phi}_s$ are given in the problem specification, this leaves only K_H and \bar{K}_H left to be determined. However, with equations (28)-(30) there are three conditions that must be met. Therefore, we need to free one of the quantities listed above. The quantity best suited for this is ϕ_{I_L} , the location of the stagnation point.

If we use equation (43) to define $P(\phi)$, equation (29) becomes

$$\int_{\sigma}^{2\pi} \left[\ln \left| \cos\left(\frac{\phi}{2} - \alpha(\phi)\right) \right| - \ln V(\phi) - \ln W_w(\phi) - K_H \ln W_s(\phi) + \ln 2 \right] \cos \phi d\phi = \pi \quad (48)$$

This can be expanded to

$$\begin{aligned} & \sum_{i=1}^{I_L} \int_{\phi_{i-1}}^{\phi_i} \left[\ln \left| \cos\left(\frac{\phi}{2} - \alpha_i\right) \right| - \ln V_i - \ln W_w - K_H \ln W_s + \ln 2 \right] \cos \phi d\phi \\ & + \sum_{i=I_L}^{I_a} \int_{\phi_{i-1}}^{\phi_i} \left[\ln \left| \cos\left(\frac{\phi}{2} - \alpha_i\right) \right| - \ln V_i - \ln W_w - K_H \ln W_s + \ln 2 \right] \cos \phi d\phi = \pi \end{aligned} \quad (49)$$

The integral $\cos\phi \ln|\cos \frac{\phi}{2} - \alpha_i| d\phi$ can be evaluated as

$$\int \cos\phi \ln|\cos \frac{\phi}{2} - \alpha_i| d\phi = (\sin\phi + \sin 2\alpha_i) \ln|\cos \frac{\phi}{2} - \alpha_i| + 1/2 (\phi \cos 2\alpha_i - \sin\phi) + \text{constant} \quad (50)$$

If we denote W_{ci} as

$$W_{ci} = - \int_{\phi_{i-1}}^{\phi_i} \cos\phi \ln W_s(\phi) d\phi \quad (51)$$

and introduce the notation

$$\ln(i,j) = \ln|\cos(\frac{\phi_i}{2} - \alpha_j)| \quad (52)$$

equation (49) can be written as

$$\begin{aligned} & K_H W_{c1} + \overline{K}_H W_{cI_a} + \sum_{i=1}^{I_a} \{ \sin 2\alpha_i (\ln(i,i) - \ln(i-1, i)) \\ & + \frac{1}{2} (\phi_i - \phi_{i-1}) \cos 2\alpha_i + \frac{1}{2} (\sin\phi_i - \sin\phi_{i-1}) + \sin\phi_i [\ln(i,i) - \ln V_i] \\ & - \sin\phi [\ln(i-1, i) - \ln V_i] \} - \int_0^{\phi_W} \cos\phi \ln W_W d\phi \\ & - \int_{\phi_W}^{2\pi} \cos\phi \ln \overline{W}_W(\phi) d\phi = \pi \end{aligned} \quad (53)$$

Due to equation (45), the next to the last term, with $i=n$, and the last term, with $i=n+1$ ($n=1, I_a-1$), cancel each other out. Out of these terms, only the last with $i=1$ and the next to the last with $i=I_a$ remain. However, since $\phi_0=0$ and $\phi_{I_a} = 2\pi$, $\sin\phi_0 = \sin\phi_{I_a} = 0$, these terms also drop out. The third term of the summation also drops out. Therefore, equation (53) can be written as

$$\begin{aligned} & K_H W_{c1} + \overline{K}_H W_{cI_a} + \sum_{i=1}^{I_a} \{ \sin 2\alpha_i (\ln(i,i) - \ln(i-1, i)) \\ & + \frac{1}{2} (\phi_i - \phi_{i-1}) \cos 2\alpha_i \} - \int_0^{\phi_W} \cos\phi \ln W_W d\phi - \int_{\phi_W}^{2\pi} \cos\phi \ln \overline{W}_W(\phi) d\phi = \pi \end{aligned} \quad (54)$$

We now introduce J_c such that

$$J_c = \sum_{i=1}^{I_a} \left\{ \sin 2\alpha_i ((\ln(i,i) - \ln(i-1,i)) + \frac{1}{2} (\phi_i - \phi_{i-1}) \cos 2\alpha_i) \right. \\ \left. - \int_0^{\phi_w} \cos \phi \ln W_w d\phi - \int_{\bar{\phi}_w}^{2\pi} \cos \phi \ln \bar{W}_w d\phi \right\} \quad (55)$$

If we define $\alpha_0 = \alpha_{I_a+1} = 0$, we can alter the indices to get

$$J_c - \pi = \sum_{i=1}^{I_a} \left\{ \sin 2\alpha_i \ln(i,i) - \sin 2\alpha_{i+1} \ln(i, i+1) \right. \\ \left. + \frac{1}{2} \phi_i (\cos 2\alpha_i - \cos 2\alpha_{i+1}) \right\} - \int_0^{\phi_w} \cos \phi \ln W_w d\phi \\ - \int_{\bar{\phi}_w}^{2\pi} \cos \phi \ln \bar{W}_w d\phi \quad (56)$$

Now equation (54) can be written as

$$K_H W_{c1} + \bar{K}_H W_{CI_a} + J_c - \pi = 0 \quad (57)$$

In a similar manner, if we define W_{si} as

$$W_{si} = \int_{\phi_{i-1}}^{\phi_i} \sin \phi \ln W_s(\phi) d\phi \quad (58)$$

and we define J_s as

$$J_s = \sum_{i=0}^{I_a} \left\{ -(1 + \cos 2\alpha_i) \ln(i,i) + (1 + \cos 2\alpha_{i+1}) \ln(i, i+1) \right. \\ \left. + \frac{1}{2} \phi_i (\sin 2\alpha_i - \sin 2\alpha_{i+1}) \right\} - \int_0^{\phi_w} \sin \phi \ln W_w(\phi) d\phi \\ - \int_{\bar{\phi}_w}^{2\pi} \sin \phi \ln \bar{W}_w(\phi) d\phi \quad (59)$$

Equation (30) can now be written as

$$K_H W_{SI} + \bar{K}_H W_{SI_a} + J_S = 0 \quad (60)$$

From equation (45),

$$\ln V_1 = \ln (i,i) - \ln (i,i+1) + \ln V_2 \quad (61)$$

Substituting this value of V_1 into equation (44) yields

$$\begin{aligned} \ln |\cos \alpha_{I_1}| - \ln (i,i) + \ln (i, i+1) + \ln V_2 - K_H \ln W_w(0) \\ = \ln |\cos \alpha_{I_a}| - \ln V_{I_a} - \bar{K}_H \ln \bar{W}_w (2\pi) \end{aligned} \quad (62)$$

From equation (45), V_2 can be determined as a function of V_3 , and so forth until V_{I_a} is reached. In this manner, V_1 and V_{I_a} are eliminated from (44), yielding

$$-K_H \ln W_w(0) + \bar{K}_H \ln \bar{W}_w (2\pi) + \sum_{i=0}^{\infty} \{-\ln (i,i) + \ln (i,i+1)\} = 0 \quad (63)$$

Defining J_T as being the summation term of this equation, equation (63) can be written as $-K_H \ln W_w(0) + \bar{K}_H \ln \bar{W}_w (2\pi) + J_T = 0$ (64)

We now have three equations (equations (57), (60), and (64)) for the three unknowns (K_H , \bar{K}_H , and ϕ_{i_L}). Eliminating K_H and \bar{K}_H yields

$$(J_c - \pi) D_1 - J_S D_2 + J_T D_3 = 0 \quad (65)$$

where

$$D_1 = W_{s1} \ln \bar{W}_w(2\pi) + W_{SI_a} \ln W_w(0) \quad (66)$$

$$D_2 = W_{c1} \ln \bar{W}_w(2\pi) + W_{CI_a} \ln W_w(0) \quad (67)$$

$$D_3 = W_{c1} W_{SI_a} - W_{CI_a} W_{s1} \quad (68)$$

Equation (65) is a transcendental equation for ϕ_{I_L} . Once the value of ϕ_{I_L} is determined, K_H and \bar{K}_H can be determined from equations (57) and (60).

With equation (45), we now have $I_a - 1$ equations for the I_a values of V_i . The last equation comes from the previously unused equation (29), which guarantees uniform flow at infinity.

$$\int_0^{2\pi} P(\phi) d\phi = \sum_{i=1}^{I_a} \left\{ \int_{\phi_{i-1}}^{\phi_i} [\ln |\cos(\frac{\phi}{2} - \alpha_i)| - \ln \frac{V_i}{V_1} - K_H \ln W_S - \ln W_W(\phi)] d\phi \right\} + 2\pi(\ln 2 - \ln V_1) = 0 \quad (69)$$

Now that all of the ϕ_i 's, K_H , and \bar{K}_H are known, $P(\phi)$ can be calculated from equation (43), $Q(\phi)$ can be calculated from equation (34), and $x(\phi)$ and $y(\phi)$ can be calculated from equations (32) and (33).

For practical numerical calculations, the circle plane is divided into $2N$ equal parts, with the positions given by

$$\phi = \tau_v = \frac{v\pi}{N} \quad (v = 0, 1, 2, \dots, 2N-1). \quad (70)$$

Next, the ϕ_i 's (except ϕ_{I_L}) α_i 's, K , \bar{K} , μ , and $\bar{\mu}$ are chosen. The values of W_{C1} , $W_{C_{I_a}}$, W_{S1} , and $W_{S_{I_a}}$ can then be calculated (equations (51) and (58)). Using equations (66) through (68), D_1 , D_2 , and D_3 can be calculated, and the transcendental equation can be established. Once ϕ_{I_L} is determined K_H and \bar{K}_H can be calculated from equations (57) and (60). $P(\phi)$ and $Q(\phi)$ can be determined at each point on the circle. $x(\phi)$ and $y(\phi)$ are then determined, so $2N$ points are determined on the airfoil. These points are equally spaced on the circle plane, but they are not equally spaced in the airfoil plane.

The resulting coordinates will yield an airfoil oriented at its zero lift line. However, the angle between the zero lift line and the chord line (β) can be determined and subtracted from α to yield the angle of attack with respect to the chord line.

Since the values of K_H and \bar{K}_H are determined by the closure requirements, there is no direct control of these values in the input specifications. K_H and \bar{K}_H determine the trailing edge angle. In order to maintain some control over this trailing edge angle, a desired value of $K_S = K_H + \bar{K}_H$ can be specified, and by iteration, varying either α_i on the upper or lower surface (or both surfaces), or K or \bar{K} , the desired K_S can be attained.

II. PROGRAM

The calculations required for the solution of the Eppler problem are carried out with the aid of an IBM-360/75 computer at the University of Illinois. The Eppler program not only determines the profile of the airfoil but also determines the boundary layer momentum thickness and the energy form parameter. However, in the present application, the boundary layer capabilities of the program have not been fully utilized.

The required programs are kept in files on the PLORTS system. The file name of the Eppler program is EPPLER, while the file names of the required input data are EDATA through EDATG. A sample input data deck is shown in Figure A1.

The first card in an input data deck for the Eppler program is a card with an Alpha-numeric listing of the titles of the cards that follow. These titles are read in 20A4 format. It is essential that the order of the titles not be changed and all titles must be included on this card, even if the named card is not used in the program. This first card can be thought of as part of the program itself, as it is never changed. The remaining cards, with the exception of the title, are in the format (A4, I6, 14F5.2). Some of the data that is input through the F5.2 format is divided by a factor of 10 in the program, so it is important not to specify the decimal point. All the data should be right justified, and the program will convert the data to the correct multiple of 10. The data that is divided by 10. in the program will be identified in the following discussion as having a psuedo-format of F5.3.

The manner in which the data on each card is treated is determined by the title, which is listed in the first 4 spaces on each card. The data is read into

TRA1TRA2ALFAAGAMABSZ REENDEBETAPLOTTITL

ABSZ 9200 100

AGAM 100 100 100 100 100 100

TITL

U OF I HLE 1657-20-28-19 AIRFOIL

PLOT 3470-83372128815625330711089424000

TRA1000027 4120 4800 4290 0975 0000 4800 9200 3200

TRA2000027 150 4290 100 5000 0542 300 4000 100 8000 0499 20014500 001

BETA -1 100

RE 03 01633

ENDE

/*

Figure A1. A sample input data deck

the program as MARKE, NUPU, and PUFF, where MARKE is the title, NUPU is an integer, and PUFF is a 14 element array. The data is then transferred to the appropriate variable according to the title.

The first title listed on the first card is the TRA1 title. The TRA1 card is the card that inputs the ϕ_i and α_i . The ϕ_i are input in terms of circle divisions, and the α_i are input in degrees. ϕ_{I_L} is determined by the program, so it is input as zero. The ϕ_i and α_i are input as pairs, and up to seven pairs can be input on one card. If it is desired to break the circle plane into more than seven segments, more TRA1 cards need be specified; with a maximum of four cards, as storage is allowed for only 28 segments. The last ϕ_i must be equal to the number of divisions in the circle. The ϕ_i must be listed in increasing order, including the computed value of ϕ_{I_L} .

Spaces 5 through 10 of the TRA1 card (NUPU) are reserved for the profile number. If several different airfoils are developed at the same time, they can be identified by this profile number.

Spaces 5 through 10 of the TRA2 card are also reserved for the profile number, but in this case, the profile number is used only to keep track of the input data, as this number is not used in the program. These spaces can also be left blank on the TRA2 card.

The remainder of the words on the TRA2 card define the input velocity function. Words 1 through 5 define the upper surface and words 6 through 10 define the lower surface. Word 1 is ϕ_s , given in circle divisions, and word 2 is ϕ_w . The meaning of words 4 and 5 depends on word 3. If word 3 is 0.0, word 4 is k and word 5 is μ . If word 3 is 1.0, word 4 is w' and word 5 is w . If word 3 is 2.0, word 4 is μ and word 5 is w . Words 4 and 5 are divided by 10.0 in the program, so the psuedo format is F5.3.

The specification of w and w' (word 3 being equal to 1.0) is recommended only with large values of w' , so the path of W_w is strongly curved. The process

converges slowly when w' is small, and convergence is not guaranteed when μ is negative. For less strongly curved paths, the specification of μ and w is recommended (word 3 equals 2.0).

Words 6 through 10 define the lower surface in the same manner that words 1 through 5 define the upper surface. Thus, for a symmetrical airfoil, words 6 through 10 would repeat words 1 through 5.

Word 11 is referred to as ITMOD, and determines the variable that is changed in the iteration process to set K_s to the specified value. If ITMOD is 0.0, no iteration is carried out. If ITMOD is 1.0, the α_i on the upper surface are altered by a factor $\Delta\alpha_i$ until K_s attains the desired value. If ITMOD is 2.0, the α_i the lower surface will be altered and if ITMOD is 3.0, the α_i will be altered on both the upper and lower surface by an equal amount. If ITMOD is 4.0, K is modified, if ITMOD is 5.0, \bar{K} is modified, and if ITMOD is 6.0, K and \bar{K} are modified by equal amounts. ITMOD = 3.0 or 6.0 is useful for symmetrical airfoils.

Word 12 is K_s , written in the psuedo-format of F5.3. Word 13 is the tolerance acceptable in the K_s computation, also written in the psuedo-format of F5.3. A suggested value for this is .001, the smallest value available in the F5.3 format. Word 14 is not used.

The next card in the list is the ALFA card. This card inputs the various angles of attack that the pressure distribution is developed for and that are used in the boundary layer portion of the program. The first word after the title is NAL, the number of angles of attack listed, in I6 format. NAL can be as large as 14. If NAL is specified as larger than 14, it is reset to 4. The next 14 (or less) words are the angles of attack, in degrees, written in F5.2 Format. If NAL is given as a negative number, the angle of attack will be α_i given on the TRAl card, where i is on the ALFA card in F5.2 format (see the sample data deck in Figure A1 for an example of this). If an ALFA card is given with no angles of attack and NAL=0, the angles of attack of the previous profile are repeated.

The AGAM card controls the output of the Eppler program. The 16 of the AGAM card is ignored, but 14 AGAM(i)'s are read in F5.2 format. In general, the AGAM(i)'s are either zero or not zero. If AGAM(1) is not zero, the x and y coordinates of the airfoil are generated. If AGAM(1) is equal to zero, only the transcendental equation is solved. If AGAM(2) is not equal to zero, the profile list will be printed, along with a velocity distribution for each angle of attack on the ALFA card. If AGAM(3) is not zero, the input data and the solution to the transcendental equation is printed out for the initial input and the final iteration. If AGAM(4) is not zero, the input data and the solution to the transcendental equation will be printed out for all iterations. AGAM(5) and AGAM(6) refer to the boundary layer portion of the program. If AGAM(5) is not zero, the program will print out a listing of the distance along the surface from the stagnation point, the local velocity, the energy thickness form parameter H_{32} (the energy dissipation boundary layer thickness divided by the momentum thickness), and the momentum thickness. If AGAM(5) is equal to 1.0, the local Reynolds number, based on the momentum thickness and the local velocity is printed out instead of the momentum thickness. If AGAM(6) is not equal to zero, the boundary layer transition point, boundary layer separation point, and drag (calculated by the Squire Young Method) are printed out. AGAM(7) through AGAM(14) are not presently used, but are reserved for further use.

At the University of Illinois, most runs are made with AGAM(1) through AGAM(6) equal to 1.0. This results in the most complete output. An attempt to run with AGAM(6) equal to zero resulted in the failure of the program for unknown reasons.

Card ABSZ lists the number of circle divisions, NKR, in spaces 11 through 15. NKR must be divisible by 4, and $NKR + 1$ points result in the profile of

the airfoil. As NKR is increased, the accuracy of the solution increases, as well as the computational time required. The maximum NKR is 120, but 60 is usually a sufficient number unless large slopes in the velocity function are encountered, as with a Stratford distribution. For the airfoils designed at the University of Illinois, an NKR of 92 was chosen.

The ABSZ card also lists ABFA in spaces 16 through 20, which multiplies all values given in circle divisions. ABFA is normally equal to 1.0. It is necessary to change ABFA only if the number of circle divisions is changed, so it is not necessary to change all the input data given in circle divisions. If no ABSZ card is given, NKR is set to 60 and ABFA is set to 1.0.

The RE card is used to input the Reynolds number into the program. The psuedo-format of the RE card is (A4, 6X, 5(211, 3X, F5.3)). The first of the 11 words represents MA, which at one time was used to determine the suction mode. Since the capability of boundary layer suction has been removed from the program, this word is no longer used. The second 11 word is MU, the mode for boundary layer transition. When MU is equal to 1, transition is by laminar separation. If MU is equal to 2, transition occurs at the first decrease in velocity. If MU is equal to 3, transition occurs when the velocity remains constant throughout a step distance or decreases. If MU is 4, transition occurs when the natural logarithm of the local Reynolds number based on δ_2 and the local velocity exceeds or equals $18.43 H_{32} - 21.74$. MU = 5 is similar to MU = 4, except the value that $\ln (RE)$ is compared to is $18.43 H_{32} - 22.10$. Therefore, MU = 5 is a more conservative estimate for transition. The F5.3 word is the free stream Reynolds number, based on the chord length and free stream velocity. All lengths in the program are non-dimensionalized with respect to this chord length, and all velocities are non-dimensionalized with respect to this velocity. There can be up to 5 Reynolds numbers, each with its own MA and MU. The program will continue to read in Reynolds numbers (up to 5)

until a zero value is read as a Reynolds number.

The ENDE card is necessary for proper termination of the program. It is the final data card, and indicates all data has been read in.

The next three titles on the list are cards that have been added to the program at the University of Illinois. The first of these cards is the BETA card, which replaces the ALFA card. If a BETA card is used instead of an ALFA card, either a punched output is generated or data is filed into the PLOTS system that is used by the Stratford program. This data consists of four parts, written in 6F12.9 format. The first part is DS, the increment of the surface distance for each x increment. There are NKR DS's generated. The other three parts are a velocity function (VF), and x and y coordinates of the airfoil. There are NKR + 1 of each of these values. The velocity function is equal to the local velocity divided by $(1 + \cos \alpha_1)$. The program was originally designed to give a punched output, but was modified to file the data directly into PLOTS. However, as the PLOTS system is due to be removed from the IBM-360 at the University of Illinois, it will be necessary to change back to a punched output deck.

The next card that has been added to the program is the PLOT card. This card reads data into the system that is then either punched out or filed into PLOTS. Nothing is done with this data by the program, as this is only a convenient method of getting data into the input deck for the Stratford programs.

The last card to be described is the TITL card. No data is on the TITL card, but this card signals that the next card is in 20A4 format, and is the title of the airfoil. This title will be printed in the output and inserted into the Stratford input deck.

There are some restrictions on the order the cards are read in. The ABSZ (if one is used), AGAM, TITL, and PLOT cards should be read into the computer first,

although not necessarily in that order. The data on these cards remains valid until another similar card is read into the computer. Thus, for example, if several profiles are to be developed with the same number of circle divisions, it is not necessary to repeat the ABSZ card. The next cards to be read in are the TRA1 and TRA2 cards, in that order. Once the TRA2 card is read in, the profile is generated. The ALFA or BETA card is then read in, followed by the RE card. The RE card initiates the calculation of the boundary layer. If other profiles are desired, new TRA1 and TRA2 cards can now be read in, preceded by new ABSZ, AGAM, PLOT, and TITL cards, as necessary. These cards can be followed by ALFA or BETA and RE cards if boundary layer information is desired. The ENDE card terminates the program after all the profiles and boundary layer calculations are complete.

The descriptions of the output which follows assumes AGAM(1) through AGAM(6) are not equal to zero. If any of these words are equal to zero, the corresponding portion of the output will be deleted.

The first data listed in the output are the input data and the solution to the transcendental equation. This data is preceded by the title, profile number, iteration number, and iteration mode (0 through 6). The headings of the table of data do not agree with the nomenclature presented in this paper. NUE represents the same quantity as ϕ_i , ALPHA is α_i , WS is w and \bar{w} , WHK is w' and \bar{w}' , DRAK is K and \bar{K} , DRAM is μ and $\bar{\mu}$, HK is K_H and \bar{K}_H , FLA is ϕ_w and $\bar{\phi}_w$, and LAS is ϕ_s and $\bar{\phi}_s$.

The next data listed are the profile of the airfoil in x and y coordinates and the velocity distribution for each angle of attack on the ALFA or BETA card. At the end of this listing, the values of CM, BETA, ETA, SX, and SY are printed out. CM is the moment coefficient at zero lift and BETA is the angle between the zero lift line and the chord line. Since all angles of attack are given in reference to the zero lift line, this angle is necessary to compute the geometric angle of

attack. ETA, SX, and SY are apparently remnants of trouble shooting the program, as they are not particularly useful. ETA is the number of points in the circle plane divided by the chord and π . This term is used in non-dimensionalizing the chord. SY and SX are summation of the x and y coordinates of the airfoil profile.

The last section of data is derived from the boundary layer portion of the program. First there are 2 tables, one for the upper surface and one for the lower surface. These tables list the surface coordinate, local velocity, H_{32} , and δ_2 . If AGAM(5) is equal to 1.0, the local Reynolds number based on δ_2 and the local velocity is printed in place of δ_2 . However, nothing in the output indicates that this has been done, so it is important that it be noted that AGAM(5) is equal to 1.0 if this data is to be used. If H_{32} is a negative number, the flow in the boundary layer is turbulent.

Following these two tables are listings for the upper and lower surface transition points, separation points, and drag coefficients. Once again, there is a problem of nomenclature, as the transition points are under the heading INS., the separation points are under the heading TRANS., and the drag coefficient are under the heading SEP.. The transition and separation points are given in terms of surface coordinates.

```

DIMENSION ALV(14),V(14),MARKEN(16)
DIMENSION XXX(3),YYY(3)
DIMENSION RE(5),PA(5),MU(5)
DIMENSION TST(5),BANT(5),CW(4,2,14),SU(4,2,14),SA(4,2,14)
COMMON PI(121),XP(121),YP(121),ARG(121),X(121),Y(121),P(121),
1PUFF(14),AGAM(14),CS(120),VF(121),ANI(90),ALFA(90),FKERN(30),ABSZ,
2ABGR,HABGR,IB,MC,NKR,NC,NUPRC,JAB,JST,CN,ETA,ABFA,PI,BGGEN, SX,
3DARG,SY,IZZ,V1,TITLE(19)
COMMON /GRZK/CDK,AA(7),BB(7)
IIII=1
DO 4 I = 1,16
4 MARKEN(I) = C.
MARKE = C.
CDK = .01
PI = 3.141592654
BGGEN = .0174532925199
10 READ(5,1)(MARKEN(I),I=1,16)
1 FORMAT(16 A4)
ABSZ = 6C.
ABFA = 1.
9 ABGR = 360./ABSZ
HABGR = .5*ABGR
IB = .25*ABSZ +.1
MC = 2*IB
NKR = 2*MC
ABSZ = NKR
NC = NKR +1
DO 8 M=1,IR
ARI = MC + 1 - 2*M
8 FKERN(M) = ABGR*CCSG(ARI*HABGR)/(SING(ARI*HABGR)*PI)
11 READ(5,2)MARKE,NLPU,PUFF
2 FORMAT(A4,I6,14F5.2)
DO 12 I=1,16
IF (MARKE.EC.MARKEN(I)) GO TO 13
12 CONTINUE
14 WRITE(6,3) MARKE
3 FORMAT(1X,'INCORRECT DATA CARD WITH CODE.....',A4)
GO TO 11
13 GO TO (15,22,333,140,142,25,150, 331,375,376,14,14,14,14,14,14),I
15 NUPRC = NUPL
J=1
16 I=1
17 ANI(J) = RUND(PUFF(I)*ABFA,1000.)
IF(ANI(J).EC.C.)JST=J
I=I+1
ALFA(J) =PUFF(I)
IF(ANI(J)-ABSZ +.1) 18,21,21
18 J=J+1
IF(I-14) 19,20,20
19 I=I+1
GO TO 17
20 READ(5,2)MARKE,NUPU,PUFF
GO TO 16
21 JAB=J
ALFA(J+1)=0
GOTO 11
22 CALL TRAPRO
GO TO 11
25 IF(PLFF(2).EC.O.) GO TO 28
DO 27 J=1,5
RERX=PUFF(2+J)

```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```

      IF(REPX)26,28,26
26 RE(J) = 1.E5*RERX
      IPU=INT(PUFF(2*J-1))
      MA(J) = IPU/100
      MU(J) = IPU/10 - 10*MA(J)
27 JR = J
28 CALL GRP(ALV,NAL,RE,MU,JR,AAA,EBB,DDD,AZM,AZ,AKK,PEO)
      GOTO 11
140 DO 141 I= 1,14
141 AGAM(I)= PLFF(I)
      GC TC 11
142 ABSZ = PUFF(1)
      ABFA = PUFF(2)
      GC TC 9
331 IPUNCH=1
      GC TC 332
333 IPUNCH=0
332 IF(NUPU)3331,335,3333
3331 NAL=-NLPL
      DO 3332 I=1,NAL
      N=IFIX(PLFF(I))
3332 ALV(I)=ALFA(N)
      GC TC 335
3333 NAL=NUPU
      IF(NLPU.GT.14)NAL=4
      DO 334 I=1,NAL
334 ALV(I)=PLFF(I)
335 IF(AGAM( 2))336,11,336
336 N=0
3361 NZ=2
      IF(IZZ.GT. 0) NZ = 3
      CALL ZEZ(I2Z,NZ,NZT)
      WRITE(6,379) NZT, TITLE
      WRITE(6,337)NUPFC, (ALV(M),M=1,NAL)
337 FCRMAT(1H ,7HPRCFIL ,15,7H ALFA,14F7.2)
379 FCRMAT(A1,19A4)
      IZZ = IZZ+1
      WRITE(6,338)
338 FORMAT(52H N X Y V-DISTRIBUTION FOR GIVEN ALPHA)
      IF(N.NE.0) GC TC 339
      DO341 N=1,NC
      IF(IZZ.GT.62) GC TC 3361
339 ND = N-1
      ZN= ND
      PHIH= ZN*HABGR
      XDR = 1CC.*X(N)
      YDR = 1CC.*Y(N)
      DO 340M=1,NAL
340 V(M) = ABS(VF(N)*COSG(PHIH - ALV(M)))
      IZZ = IZZ+1
341 WRITE(6,342)ND,XDR ,YDR ,(V(M),M=1,NAL)
342 FORMAT(I4,2F8.3,14F7.3)
      IZZ = IZZ+1
      WRITE(6,344)CM,DARG,ETA,SX,SY
344 FORMAT(4H CM=,F7.4,6H BEFA=,F5.2,9HDEG. ETA=,F5.3,4H SX=,F6.3,4H S
      IY=,F6.3)
      IF(IPUNCH.EQ.0) GC TO 11
      NC1=NC-1
      IIII=IIII+1
      WRITE(7,330) (DS(M),M=1,NJ1)
      WRITE(7,330) (VF(M),M=1,NJ)

```

```
WRITE(7,330) (X(M),M=1,NQ)
WRITE(7,330) (Y(M),M=1,NQ)
GC TC 11
330 FORMAT(6F12.9)
375 AAA=PLFF(1)/100.
    BBB=PUFF(2)/100.
    DDD=PUFF(3)/100.
    AZM=PUFF(4)/100.
    AZ=PLFF(5)/100.
    AKK=PUFF(6)/100.
    REO=PUFF(7)*10000.
GC TC 11
376 READ(5,377) TITLE
    WRITE(7,378) TITLE
377 FORMAT(15A4)
378 FORMAT(' ',15A4)
GC TC 11
150 STOP
    CEBLG SUBCHK
    END
```



```
SUBROUTINE ZEZ(IZ,ID,IZT)
DIMENSION IT(4)
DATA IT/1H+,1H ,1H0,1H1/
IF(IC-2)2,2,1
1 ID=3
  IZ = 1
  GO TO 3
2 IZ = IZ+ID
3 IZT = IT(IC+1)
RETURN
END
```

```

SUBROUTINE CECF(RET,U,US,H,D2,CD,CF,H12)
CALL F12B(H,F12,EPST)
IF(H)2,15,1
1 CC=((H*6.8377961-20.521103)*F+15.707952)/RET
CF=EPST/RET
GO TO 15
2 H32=-H
ALPHA=0.029*(0.93-1.95*ALJG10(F12))*1.7C5
CF=ALPHA/RET**C.268
T=(1.C-1.0/F12)/SQRT(CF)
PI=-F12*C2*LS/(CF*U)
IF(T-10.C)3,3,4
3 P1=-0.77380-T*(0.07212-T*(0.03263-0.000942*T))
GO TO 5
4 P1=-(107.7651-T*(20.4916-T*(1.1407-T*(0.0281-0.0001*T))))/
1 (20.5403-T*(1.1341-T*(0.0312-0.0002*T)))
5 IF(PI-P1)9,6,6
6 C3=0.0
IF(T-10.C)8,8,7
7 C1=106.2651-T*(20.4916-T*(1.1407-T*(0.0281-0.0001*T)))
C2=18.0403-T*(1.1341-T*(0.0312-0.0002*T))
GO TO 14
8 C1=7.93440+T*(0.87300-T*(0.39501-0.01139*T))
C2=9.605
GO TO 14
9 P5=-3.63819+T*(0.93059-T*(0.09422-T*(0.004641-T*0.731E-04)))
IF(PI-P5)10,11,11
10 C1=-76.C+T*(63.7-7.C*T)
C2=-30.5-(C1+33.5)/P5
C3=0.0
GO TO 14
11 P3=-1.07221+T*(0.06537+T*(0.01046-T*(0.000174-T*0.6E-05)))
IF(PI-P3)13,12,12
12 P2=-0.42195-T*(0.17378-T*(0.04022-T*(0.001428-T*0.25E-04)))
C3=((6.5*P2-10.9*P3-2.9)*(P2-P1)+(6.5*P2-2.5*P1+1.8)*(P3-P2))/
1 ((P3-P1)*(P3-P2)*(P2-P1))
C2=(6.5*P2-2.5*P1+1.8)/(P1-P2)-C3*(P1+P2)
C1=-1.5-F1*(C2+2.5)-C3*P1*P1
GO TO 14
13 P4=-2.24228+T*(0.45376+T*(0.03652+T*(0.001926-T*0.277E-04)))
C3=((19.4*P4-30.5*P5-16.7)*(P4-P3)+(19.4*P4-10.9*P3+10.6)*(P5-P4))/
1 ((P5-P3)*(P5-P4)*(P4-P3))
C2=(19.4*P4-10.9*P3+10.6)/(P3-P4)-C3*(P3+P4)
C1=-6.2-P3*(C2+10.9)-C3*P3*P3
14 TP=C1+C2*PI+C3*PI*PI*PI
TSP=(H12-(H12-1.0)*(1.0-0.72155/(0.93-1.95*ALJG10(H12))))*
1 (0.55216-C.38750*H32+0.04855*SQRT(0.77500*H32-1.10667))/
2 (SQRT(0.77500*H32-1.10667)*SQRT(ALPHA)*(H12*(H32-1.43100))*2)
CC=CF*((TP-C.134*T)*(1.0+PI*(1.0+1.0/H12)))/(TSP*RET**0.134)+H32*
1 (1.0+PI*(1.0-1.0/H12))
15 RETURN
END

```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```

SUBROUTINE GRUP(F,C2,U,WRE,LS,DL,Z,DZ,DO2,V,MA)
COMMON /GRZK/CDK,AA(7),BB(7)
Z = C2*ABS(F)
RET = WRE*WRE*U*C2
CALL CDCF(RET,U,LS,F,DZ,CD,CF,F12)
IF(MA)1,1,2
1 V = 0.
GC TC 13
2 GC TC(13,13,13,6,6,5,5,3),MA
3 IF(F)4,1,1
4 V = 25.*(H+1.98)*D2*LS
GC TC 12
5 IF(F)6,1,1
6 B = BB(MA)
PSI = AA(MA)
IF(BV.NE.0.)PSI=BV*ALOG(RET)+PSI
IF(PSI-1.52)8,7,7
7 IF(PSI-1.99)11,11,9
8 PSI=1.52
GC TC 10
9 PSI=1.99
10 B = C.
11 FZ = SIGN(PSI,H)
CALL CDCF(RET,U,US,HZ,DZ,CDS,CFS,H12S)
V=(U*(CDS-(PSI+B)*CFS)+D2*US*(E-PSI+F12*(B+PSI)))/(B+ABS(H)-1.)
12 IF(V)13,1,1
13 VDU =V/U
LSU =LS/L
17 CD2 = (CF-(2.+H12)*LSU*D2+VDU)*DL
DZ = (CD - 3.*Z*USU + VDU)*DL
RETURN
END

```

```

SUBROUTINE F12B(H,H12,EPST)
  IF(H)4,5,1
1 IF(H-1.57258)2,3,3
2 H12 = (SQRT(H-1.515090)) * ((-227.18220 * H + 724.55916) * H - 583.60182)
  1 + 4.0292200
  EPST = ((-0.03172850655 * H12 + 0.3915405523) * H12 - 1.686094798) * H12
  1 + 2.512588652
  GO TO 5
3 EPST = (H * 2.221687229 - 4.226252829) * H + 1.372390703
  H12 = (25.71578574 * H - 89.53214201) * H + 79.87084472
  GO TO 5
4 H12 = (0.04855 - SQRT(-0.77500 * H - 1.106668)) / (H + 1.43100)
5 RETURN
  END

```

```

SUBROUTINE GRP(ALV,NAL,RE,MU,JR,AAA,BBB,DDD,AZM,AZ,AKK,REO)
COMMON P1(121),XP(121),YP(121),ARG(121),X(121),Y(121),P(121),
1PUFF(14),AGAM(14),ES(120),VF(121),ANI(90),ALFA(90),FKERN(30),ABSZ,
2ABGR,FABGR,IB,MC,NKR,NC,NUPRC,JAB,JST,CM,ETA,ABFA,PI,BOGEN,SX,
3DARG,SY,IZZ,V1,TITLE(19)
DIMENSION ALV(14),RE(5),MU(5),WRE(5),H(5),D2(5),CH(5,2,14),
1SA(5,2,14),SU(5,2,14),SAR(5),SUR(5),CWD(5),DD(5),RR(5),REI(5)
HABLC=2.49
UKK=C.O
DC 2 J=1,JR
H(J) = J
RR(J) = RE(J)*1.E-6
2 WRE(J)= SQRT(PE(J))
NALA = 1
NALE = IABS(NAL)
IF(NAL.LT.O) NALA = NALE
DO 38 IA = NALA, NALE
IF(AGAM(5).NE.O.) IZZ = 99
DC 38 JU =1,2
ND = 2*JU -3
NDSD = JU - 1
FND = ND
NEND = 1 + NDSD*NKR
AP = ALV(IA)/HABGR + .5*ABSZ + 1.
NU = INT(AP+.01*FND) + NDSD
NLNDSD = NU - NDSD
DSR = DS(NUNDSD)*ABS(FLJAT(NU)-AP)
S=C.
UK=C.
IF(AGAM(5))4,26,4
4 IZP = IZZ + IABS(NU-NEND) -64
IF(IZP)10,10,6
6 CALL ZEZ(IZZ,3,IZT)
WRITE(6,7) IZT,TITLE
WRITE(6,8)NUPRC,ALV(IA)
7 FORMAT(A1,1SA4)
8 FORMAT(1H , 'BCUNDARY LAYER      PROFILE',I6,10X, 'ALPHA=',F6.2)
10 CALL ZEZ(IZZ,1,IZT)
IF(JU-1)16,12,16
12 WRITE(6,14) IZT,(RR(J),MU(J),_=1,JR)
14 FORMAT(A1,13HUPPER SURFACE,2X,5(3X,3HRN=,F6.2,4X,3HMU=,I2))
GOTO 20
16 WRITE(6,18) IZT,(RR(J),MU(J),_=1,JR)
18 FORMAT(A1,13HLOWER SURFACE,2X,5(3X,3HRN=,F6.2,4X,3HMU=,I2))
IF(IZP)26,26,20
20 CALL ZEZ(IZZ,1,IZT)
WRITE(6,22) IZT,(J,J=1,JR)
22 FORMAT(A1,7H  S      ,8H  U      ,5(11,9H  H32      ,11H  D2      ))
GC TC 26
24 NUNDSD = NU - NDSD
DSR = DS(NUNDSD)
26 ZN = NU
CALV=CCSG((ZN-1.O)*HABGR-ALV(IA))
UK1=ABS(VF(NL)*CALV)
S = S + DSR
DC 33 J=1,JR
DSZ = DSR
UR = UK
CALL GRS(H(J),D2(J),UR,JK1,DSZ,WRE(J),MU(J),0,0,0,0,0,0,H(J),D2(J),
1XA,XL,CCG)
DC(J) = C2(J)

```

```

      IF(AGAM(5).EQ.1.)DD(J)=UK1*RR(J)*D2(J)
      IF(UK)30,28,30
28   SAR(J)=C.0
      SUP(J)=G.0
30   SAR(J)= SAR(J)+XA
33   SLR(J)= SLR(J)+XU
333  IF(AGAM(5))31,35,31
31   CALL ZEZ(IZZ,1,ITZ)
      WRITE(6,32)ITZ,S,UK1,(H(KK),CC(KK),KK=1,JR)
32   FORMAT(A1,F7.4,F8.4,5(F10.5,F11.6))
35   IF(NU-NEND)34,36,34
34   UK = UK1
      IF (UK1.GT.LKK) LKK=UK1
      NU = NU +ND
      GC TC 24
36   IF (J.EQ.1) WRITE(7,376) AAA,EBB,DDD,UKK,AZM,AZ,S,AKK
      IF (J.EC.1) WRITE(7,377) REQ
377  FORMAT(F15.4)
      DC38 J=1,JR
      CALL H12R(H(J),H12R,CTS)
      IF(H12R.GT.HABLC)H12R = HABLC
      CW(J,JU,IA)=(UK1**((2.5+.5*H12R))*D2(J)*2.
      SA(J,JU,IA)= SAR(J)
38   SU(J,JU,IA)= SLR(J)
376  FORMAT(8F10.4)
      I = 0
      IF(AGAM(6))39,60,39
39   IF(AGAM(5))44,40,44
40   IF (IZZ-55)42,44,44
42   IF(NAL)48,44,44
44   CALL ZEZ(IZZ,3,ITZ)
      WRITE(6,7) TITLE
      WRITE(6,46)NLPRC
46   FORMAT(1F, 'BCUNDARY LAYER RESULTS      PROFILE',I6)
48   CALL ZEZ(IZZ,2,ITZ)
      WRITE(6,4)ITZ,(RE(J),MU(J),L=1,JR)
49   FORMAT(A1,15X,5(3FRE=,E9.2,1X,3HMU=,I2,3X))
      IF(I.NE.C) GOTO 51
      DC 60 I=NALA,MALE
      IF(IZZ.GE.57) GOTO 44
51   CALL ZEZ(IZZ,1,ITZ)
      WRITE(6,50)ITZ,ALV(I),(J,J=1,JR)
      WRITE(7,55) ALV(I)
55   FORMAT('  ENAM1 ALPHA=',F5.2,' ,N=92,END')
50   FORMAT(A1,6HALPHA=,F6.2,3X,5(11,7H INS. ,7H TRANS.,7H SEP. ))
      CALL ZEZ(IZZ,1,ITZ)
      WRITE(6,52)ITZ,(SL(J,1,I),SA(J,1,I),CW(J,1,I),J=1,JR)
52   FORMAT(A1,13HUPPER SURFACE,2X,5(2F7.3,F7.4))
      CALL ZEZ(IZZ,1,ITZ)
      WRITE(6,54)ITZ,(SL(J,2,I),SA(J,2,I),CW(J,2,I),J=1,JR)
54   FORMAT(A1,13HLOWER SURFACE,2X,5(2F7.3,F7.4))
      DC 56 J = 1,JF
56   CWC(J) = CW(J,1,I) +CW(J,2,I)
      CALL ZEZ(IZZ,1,ITZ)
      WRITE(6,58)ITZ,(CWC(J),J=1,JR)
58   FORMAT(A1,8HTOTAL CC,7X,5(13X,F8.4))
60   CONTINUE
      RETLPA
      END

```

```
SUBROUTINE LMP(US,U,D2,WRE,F,M,FUM)
GO TO(1,2,3,4,8,8,8),M
1 IF(US+1.E-4) 7,8,8
2 IF(US-1.E-4) 7,7,8
3 AK=21.74
GO TO 45
4 AK=22.10
45 FUM=ALOG(WRE*WRE*D2*U)+AK-18.4*M
GO TO 9
7 FUM=22.10
GO TO 9
8 FUM=-22.10
9 RETURN
END
```

```

SLBFCUTINE GRS(H,C2,UK,UK1,DL,WRE,MU,MA,V,V1,HR,D2R,XA,XU,DCC)
DATA EAP,EUM,FE/.000005,.0001,-1.0/
BIT=C.
IF(MA)3,101,3
101 V1 = C.
V = C.
IF(FE)3,3,1
1 IF(ABS(UK-UKV)+ABS(UK1-UK1V)+ABS(DL-DLV)+ABS(H-HV)-.1E-6)2,3,3
2 IF(ABS(C2*WRE-D2V*WREV).GE..1E-6) GO TO 3
BIT = 1.
D2E = D2E*WREV/WRE
3 D2V=C2
HV = H
UKV= UK
DLV = DL
WREV = WRE
DCC = 0.
XA = CL
XU = DL
XI=DL
VV = V
IF(DL)4,4,5
4 FE=F
D2E= C2
GO TO 50
5 LSTR =(LK1-LK)/DL
IF(MA.LT.4)VSTR =(V1-V)/DL
IF(UK)6,7,6
6 IF((LK1-LK)/LK - 1.)8,8,7
7 D2E = .290043/(WRE*SQRT(LSTR))
HE = 1.619977
GO TO 50
8 IF(C2)9,9,10
9 D2E =(1.664108/WRE)*SQRT(2.*DL/(LK1+LK))
HE = 1.572584
GO TO 50
10 X = 0.
ISTAB = C
IF(F.LT.C.)XU = 0.
IF(BIT)12,12,11
11 CALL UMF(LSTR,LK,D2,WRE,H,FU*)
GO TO 40
13 IF(X +DL -DLV)12,12,19
12 XS = X
UK=LKV+LSTR*XS
VG= VV + VSTR*XS
IF(H)14,14,16
14 HAP=1.5162
IF(HAP+EAP+F)20,20,15
15 XA=X
CALL F12E(F,F12,EPST)
D2F = D2*(LK/LK1)*((5.+H12)*.5)
FE = F
GO TO 50
16 HAP = 1.515095
IF(HAP+EAP-F)17,18,18
17 CALL UMF(LSTR,LK,C2,WRE,H,ML,FLM)
IF(FLM+ELM)20,18,18
18 XL = X
H=-H
ISTAB = C

```



```

19 DL = DLV - X
   GO TC 12
20 CALL GRUP(H,D2,UK,WRE,WSTR ,DL,Z,DZ,DD2,VG,MA)
   ESP = .CC2
   IF(MA-3)22,22,21
21 ESP = .CC1
   IF(X .EC.0.)V = VG
22 XS = XS + .5*DL
   D2M = D2 + .5*DD2
   ZM = Z + .5*DZ
   UM=LKV+LSTR*XS
   ISG = 1
   IF(D2M)25,25,23
23 ISG = 2
   HM = ZM/D2M
   IF(HM - 2.)30,25,25
25 ISGV = ISG
   IF(ISTAB-9)26,49,49
26 ISTAB = ISTAB +1
   DL = .5*DL
   GO TC 13
30 IF(HM - HAP+EAP)31,32,32
31 DL = .5*DL*(ABS(H)-HAP)/(ABS(H)-HM)
   GO TC 13
32 IF(H.LT.C.)HM = -HM
   IF(MA.LT.4)VM= VV+ XS*VSTR
   CALL GRUP(HM,D2M,UM,WRE,WSTR ,DL,ZM,DZM,DD2M,VM,MA)
   ISG = 3
   D2E = D2 +CC2M
   IF(D2E)25,25,33
33 HE= (Z+DZM)/D2E
   ISG = 4
   IF(HE-2.)34,25,25
34 HDIFF = HE-APS(H)
   ISG = 5
   IF(ABS(HDIFF)-.01)35,35,25
35 HS = ABS(ABS(H)-2.*ABS(HM)+HE)
   ISG = 6
   IF(HS-ESP)36,36,25
36 IF(HE-HAP+FAF)37,38,38
37 DL = DL*(ABS(H)-HAP)/(ABS(H)-HE)
   GO TC 13
38 IF(H)39,39,40
39 HE = -HE
   GO TC 42
40 UE=LK+LSTR*DL
   CALL LMP(WSTR,UE,D2E,WRE,HE,NL,FUME)
   IF(FUME-EUM)42,42,41
41 DL = DL*FUM/(FLP-FLME)
   GO TC 13
42 DCC = DCC + DL*VM
424 X=X+DL
   IF(X -DLV)47,50,50
47 H = HE
   D2 = D2E
   GO TC(43,13,43,45,13,43),ISGV
43 IF(HS-.1*ESP)44,13,13
44 DL = 2.*DL
   ISTAB =ISTAB-1
   GO TC 13
45 IF(HDIFF)44,13,13

```

```
49 STP=0.0
   IF(ISTAB.NE.9) STP=FLJAT(1/(ISTAB-9))
   WRITE(6,51)
51 FORMAT(1X,'*****WARNING*****  B.L. INFORMATION IS NOT VALID AFTER
   HERE. STEP SIZE IS AT ITS LOWEST BOUND.')
50 HR = HE
   DZR = DZE
   RETURN
   END
```

SUBROUTINE DRAW(WC,WS,WL,DK,DM,FL,AG,MA)

CCSP = CCSG(AG*FL)

CALL DRAW(WC,WS,WL,DRAK(J),DRAK(J),FLA(J),ABGR,MKP)

WEEN MKP=0, WIRD DK ALS KAPPA ALFGEFASST, SONST ALS K.

FK= DK

IF(MA)2,1,2

1 FK= FK*(1.-CCSP)/(1.+CCSP)

2 SINP = SING(AG*FL)

WL = -DM*ALCG(1.+FK)

IF(FL.EQ.0.) FK = 1.

BETA = (CCSP-1.)/FK + CCSP

BQM1 = BETA**2 - 1.

WUBEQ= SQRT(ABS(BQM1))

U= (1.+BETA)*SINP/(1.+CCSP)

IF(BQM1)3,5,4

3 WF = ALCG(ABS((WUBEQ+U)/(WUBEQ-U)))

GO TO 6

4 WF = 2.* ATAN(U/WUBEQ)

GO TO 6

5 WF = C.

6 WC =(WUBEQ*WF - SINP - BETA*FL*AG*1.745329E-2)*DM

WS = (CCSP-1.)*(1.-(1./FK + 1.)*ALOG(1.+FK))*DM

RETURN

END

```

SUBROUTINE TRAPRO
  DIMENSION PLRES(13),FLS(2),FLA(2),DRAK(2),DRAM(2),AC(4,3),D(3),
  1WSI(2),WCI(2),FINT(3),A(4),FK(2),R(3)
  COMMON PL(121),XP(121),YP(121),ARG(121),X(121),Y(121),P(121),
  1PUFF(14),AGAN(14),DS(120),VF(121),ANI(90),ALFA(90),FKERN(30),ABSZ,
  2ABGR,FABGR,IB,NC,AKR,NG,NUPRO,JAB,JST,CM,ETA,ABFA,PI,BDGEN,SX,
  3DARG,SY,IZZ,V1,TITLE(19)
  IZZ = 99
  DO 23 I = 1,13
23  PURES(I)=PLND(PLFF(I),1000.)
    I=1
    J=1
24  FLS(J)= PLRES(I)*ABFA
    CALL DRAW(WC,WL,.6,-1.,FLS(J),ABGR,1)
    CALL DRAW(WCI(J),WSI(J),WLI,-.6,-1.,FLS(J),ABGR,1)
    WCI(J)= WCI(J)+WC
    WSI(J)= WSI(J)+WS
    WLI = WLI+WL
    FLA(J)= PURES(I+1) * ABFA
    IF(FLA(J))25,25,26
25  DRAK(J)= 0
    DRAM(J)= 1.
    GO TO 34
26  WI = COSG(ABGR*FLA(J))
    IF(PURES(I+2)-1.)27,30,29
27  DPAK(J)= .1*PURES(I+3)
28  DRAM(J)= .1*PLFES(I+4)
    GO TO 34
29  DRAK(J)=((.1*PLRES(I+4))*(-10./PURES(I+3))-1.)*(1.+WI)/(1.-WI)
    DRAK(J)= RUNC(DRAK(J),1000.)
    DRAM(J) = .1*PURES(I+3)
    GO TO 34
30  AA = .05*(1.-WI)*PURES(I+3)
    WILN = ALOG(.1*PLRES(I+4))
    FMIT = .5
    MIT = 0
31  FM = -WILN/ALOG(AA/FMIT +1.)
    MIT=MIT+1
    IF(ABS(FM-FMIT)-1.E-6) 33,32,32
32  FMIT = FM
    GO TO 31
33  DRAM(J) = RUNC(FM,1000.)
    DRAS = .05*PLRES(I+3)*(wi+1.)/FM
    DRAK(J) = RUNC(DRAS,1000.)
34  I= I+5
    J= J+1
    IF(J-3)24,38,38
38  MER = 0
    WSI(2)= -WSI(2)
    D(1) = WLI *(WSI(2)+WCI(1))
    D(2) =-WLI *(WCI(2)+WCI(1))
    D(3) = WCI(1)*WSI(2)-WCI(2)*WSI(1)
    ITMCD = PURES(11)
    ITNR=ITMCD
    SHKS = .1*PLFES(12)
    HKST = .1*PLFES(13)
35  DO 36 J=1,3
36  AC(1,J)= 0.
    ALIV = 0.
    SINAI = 0.
    COSAI = 1.

```

```

FNI = 0.
J=1
37 CSAIP = COSG(2.*ALFA(J))
   SNAIP = SING(2.*ALFA(J))
   IF(J-JST-1)40,39,40
39 AC(2,1)= SINAI
   AC(2,2)= -1.-COSAI
   AC(2,3)= -1.
   AC(3,1)= -SNAIP
   AC(3,2)= 1.+CSAIP
   AC(3,3)= 1.
   AC(4,1)= CESAI-CSAIP
   AC(4,2)= SINAI-SNAIP
   AC(4,3)= C.
   ALIS = ALIV
   ALISP = ALFA(J)
   GOTO 41
40 FII = CSLG(HABGR*FNI-90.,ALIV)
   FIIP = CSLG(HABGR*FNI-90.,ALFA(J))
   PB=FNI*HABGR*BOGEN
   AC(1,1)=-FII*F*SNAIP+FII*SINAI+(COSAI-CSAIP)*PB +AC(1,1)
   AC(1,2)=-FII*(1.+COSAI)+FIIP*(1.+CSAIP)+(SINAI-SNAIP)*PB+AC(1,2)
   AC(1,3)= FIIP - FII + AC(1,3)
41 IF(J-JAE-1)42,43,43
42 ALIV =ALFA(J)
   SINAI=SNAIP
   COSAI=CSAIP
   FNI = ANI(J)
   J=J+1
   GC TC 37
43 DC 47 J=1,2
   IF(FLA(J)) 47,47,49
49 CALL DRAW(WC,WS,WL,DRAK(J),DRAK(J),FLA(J),ABGR,0)
   AC(1,1)= WC+ AC(1,1)
   IF(J-2)45,44,45
44 WS = -WS
   WL = -WL
45 AC(1,2) = WS +AC(1,2)
   AC(1,3) =-WL +AC(1,3)
47 CONTINUE
   DC 52 J=1,4
   A(J) = C.
   DC 52 I=1,3
52 A(J) = A(J)+D(I)*AC(J,I)
C LGTRA
53 I=C
   FV = 9.E9
   PHISH = .5 *(ALIS+ALISP)
60 CSLI = CSLG(PHISH,ALIS)
   CSLIP = CSLG(PHISH,ALISP)
   FP=(A(1)+A(4)*90.*BOGEN)+(A(2)*CSLI+A(3)*CSLIP)+A(4)*PHISH*BOGEN
   IF(I-20)61,66,66
61 IF(ABS(FP)-ABS(FV) + .5E-3)62,63,63
63 I = 20
   PHISH = PHISH - PDIF
   GC TC 60
62 PDIF = -FP / (A(2)/(PHISH-ALIS) + A(3)/(PHISH-ALISP))
   I = I+1
65 FV=FP
   PHISH = PHISH + PDIF
   GC TC 60

```

```

66 ANI(JST) = (PHISH+90.)/HABGR
70 DO 71 I=1,3
71 FINT(I)=AC(1,I)+AC(2,I)*CSLI+AC(3,I)*CSLIP+AC(4,I)*BOGEN*(PHISH+
F90.)
69 HK(1)=(FINT(1)*WLI-FINT(3)*WCI(2))/C(2)
HK(2)=(FINT(1)*WLI+FINT(3)*WCI(1))/D(2)
HKS = HK(1)+HK(2)
IF(ABS(HKS -SHKS) -HKST)74,72,72
72 IF(ITMCD)73,75,73
73 IF(AGAM(4))76,58,76
74 ITMCD = C
75 IF(AGAM(3))76,300,76
76 NZ = 2
IF(IZZ+JAB.GE.59) NZ=3
CALL ZE2(IZZ,NZ,NZT)
WRITE(6,7)NZT,TITLE
89 WRITE(6,77)NUPRC,MER,ITMR
7 FORMAT(A1,19A4)
77 FORMAT(1F,20HRESULTS FOR PROFILE ,I5,5X,10HITERATION ,I3,5X,5HMCDE
1 ,I2)
IZZ = IZZ+1
WRITE(6,78)
78 FORMAT(' NLE ALPHA AS WHK DRAK DRAM HK FLA LAS
1')
JH= 1
JN= 1
79 IZZ = IZZ+1
IF(JN-1)80,81,80
80 IF(JN-JAB)83,81,83
81 X1 = .5*(1.+COSG(FLA(JH)+ABGR))
WHK = (1.+CPAK(JH)*(1.-X1)/X1)**(-DRAM(JH))
WSTR= CPAK(JH)*CPAK(JH)/X1
WRITE(6,82)ANI(JN),ALFA(JN),WSTR,WHK,DRAK(JH),DRAM(JH),HK(JH),FLA(
1JH),FLS(JH)
82 FORMAT(F7.2,F6.2,F7.3,F6.3,2F7.3,F9.6,F5.1,F4.1)
JH= JH+1
IF(JN-JAB)85,86,85
83 WRITE(6,82)ANI(JN),ALFA(JN)
85 JN=JN+1
GOTO 79
86 IF(ITMCD)100,300,100
88 IF(MER)100,99,100
99 IF(AGAM(3))76,100,76
100 IF(ITMCD-4)101,120,120
101 IF(MER)103,102,103
102 DAL = .1
GOTO 104
103 DAL = (SHKS-HKS)*DAL/(HKS-HKSV)
DAL = FUND(DAL,100.)
IF(DAL)104,74,104
104 J = 1
HL=C.
105 IF(HL)107,106,107
106 IF(ITMCD-2)106,109,108
107 IF(ITMCD-1)108,111,108
108 ALFA(J) = ALFA(J)+DAL*(1.-HL)
109 IF(J-JST)111,110,111
110 HL=2.
111 J = J+1
IF(J-JAB-1)105,112,105
112 HKSV=HKS

```

```

MER =MER+1
GCTC 35
120 IF(MER)122,121,122
121 DDK=.1
GCTC 123
122 DDK=(SHKS-HKS)*DDK/(HKS-HKSV)
DDK= RUNC(DEK,1000.)
IF(DEK)123,74,123
123 IF(ITMOD-5)124,125,124
124 DRAK(1)= DRAK(1)+DEK
125 IF(ITMOD-4)126,112,126
126 DRAK(2)= DRAK(2)+DEK
GC TC 112
300 IF(AGAM(1))301,11,301
301 AK=.5*(CCSG(PHISH-ALFA(JST+1))/SING(PHISH-ALFA(JST+1))
1 -CCSG(PHISH-ALFA(JST))/SING(PHISH-ALFA(JST)))
AKP =AK*180./9.8696044
PHIM = C.
NU=1
I= 1
ANU =C.
JH=0
VI= C.
302 JH=JH+1
FF1 = CCSG(ABGR*FLA(JH))
FF2 = DRAK(JH)/(1.+FF1)
FG1 = CCSG(ABGR*FLS(JH))
FG3 = .6/(FG1-1.)
304 VI= VI - CSLG(PHIM-90., ALFA(I))
GC TC 310
306 ARGN = ANU
IF(ANU.GT..5* ABSZ)ARGN= ABSZ - ANU
CSP = CCSG(ARGN*ABGR)
F=0.
IF(ARGN.LT.FLA(JH))F=DRAX(JH)*ALOG((CSP-FF1)*FF2+1.)
G=0.
IF(ARGN.LT.FLS(JH))G=-FK(JH)*ALOG(1.-((CSP-FG1)*FG3)**2)
P(NU)= F+G+CSLG(ANU*HABGR-90.,ALFA(I)) + VI
P1(NL)=F(NU)-AK*ABS(SING((ANU*HABGR - 90.) - PHISH))
NL = NU + 1
ANL= ANL+ 1.
310 IF(ANU-ANI(I))306,306,312
312 IF(ANU- ABSZ)314,320,320
314 PHIM = ANI(I)*HABGR
VI=VI+CSLG(PHIM-90.,ALFA(I))
I = I+1
IF(I-1-JST)304,302,304
320 PS=C.
B2=C.
DC 324 I=1,NKP
PS=PS+P(I)
BI = 2*(I-1)
324 B2 = B2 + SING(BI*ABGR)*P(I)
V1 = 2.*EXP(PS/ABSZ)
SXI = .CCCCCCCC
SY=C.
DC328 N=1,NG
Q=0.
DC326 M=1,IB
MN = N + 1 + MQ - 2*M
MM = 2*M - MN

```

REPRODUCED
ORIGINAL PAGE IS POOR

```

IF(MN.GT.NKR) MN = MN - NKR
IF(MN.LT.1) MN = MN + NKR
326 Q = Q + FKERN(M)*(P1(MN)-P1(MM))
ANU = N-1
ZP = ANU*HABGR - 90.
ZL = COSG(ZP - PHISH)
ZL = ABS((1.-ZL)/(1.+ZL))
IF(ZL.NE.0.)ZL=ALOG(ZL)
ARG(N) = Q - AKP*SING(ZP-PHISH)*ZL + ZP
VF(N) = V1*EXP(-P(N))
WV = COSG(ZP)/VF(N)
XP(N)= WV*SING(ARG(N))
YP(N)=-WV*CCSG(ARG(N))
SXI= SXI+ XP(N)
328 SY = SY + YP(N)
SX = SXI
XPK = SX/(ABSZ - 1.)
329 X(1)=C.
YPK = SY/(ABSZ -1.)
SXI = .CCCCC000
Y(1)=C.
XP(1)= XPK
YP(1)= YPK
XP(NG)=XPK
YP(NG)=YPK
RCV = 0.
DO330 I = 2,NG
SXI= XP(I-1) -XPK - XPK + XP(N) + SXI
X(N) = SXI
Y(N) = Y(N-1) + (YP(N-1) - YPK + YP(N) - YPK)
RQ=X(N)*X(N)+Y(N)*Y(N)
IF(RQ.GT.RQV)L=N
330 RQV = RQ
DO 327 I = 1,3
IEPPL = L-2+I
327 R(I)=SQRT(X(IEPPL)*X(IEPPL)+Y(IEPPL)*Y(IEPPL))
333 TAU = (R(3)-R(1))/(4.*(R(2)+R(2)-R(1)-R(3)))
XNAS = X(L)+TAU*(X(L+1)-X(L-1))+2.*TAU*(X(L+1)+X(L-1)-X(L)-X(L))
YNAS = Y(L)+TAU*(Y(L+1)-Y(L-1))+2.*TAU*(Y(L+1)+Y(L-1)-Y(L)-Y(L))
SQ = XNAS*XNAS + YNAS*YNAS
AT=XNAS/SQ
B= YNAS/SQ
STREF = 1./SQRT(SQ)
ETA = ABSZ*STREF/PI
CM = .5*ETA*STREF*B2
CARG = 19.09859 *(3.*YNAS/XNAS - (YNAS/XNAS)**3)
SX=STREF*SX*200.
SY=STREF*SY*200.
DO 331 N=2,NG
XR=X(N)
X(N)= 1.-B*Y(N)-AT*XR
Y(N)= B*XR -AT*Y(N)
ARG(N) = ARG(N) - CARG
WQ = (XP(N)+XP(N-1)-XPK-XPK)**2 + (YP(N)+YP(N-1)-YPK-YPK)**2
331 DS(N-1) = STREF*SQRT(WQ)*(1.+0.6666667*((XP(N)*YP(N-1)
1-XP(N-1)*YP(N))/WQ)**2)
X(1) = 1.
332 ARG(1) = ARG(1) - CARG
11 RETURN
END

```



```
SUBROUTINE CIP(X,Y,A)
DIMENSION X(3),Y(3),A(3)
C1 = (Y(2)-Y(1))/(X(2)-X(1))
A(3)=(Y(3)-Y(1)-C1*(X(3)-X(1)))/((X(2)-X(1))*(X(2)-X(1)))
A(1)=Y(1)-C1*X(1)+A(3)*X(1)*X(2)
A(2)=C1-A(3)*(X(1)+X(2))
RETURN
END
```

```

SUBROUTINE CIA(X,Y,NP,D)
DIMENSION U(3),V(3),W(3),Z(3),A(3),B(3),X(121),Y(121)
D = 0.
DO 4 N=2,NP
NR = NP-1
1 IF(X(NR)-X(N))3,2,2
2 NR = NR - 1
GO TO 1
3 DNN=Y(N)-Y(NR)-(Y(NR+1)-Y(NR))*(X(N)-X(NR))/(X(NR+1)-X(NR))
IF(DNN-D)5,4,4
4 D = DNN
5 DO 6 I = 1,3
NC = N+1-I
NU = NR-1+I
U(I)=X(NC)
V(I)=Y(NC)
W(I)=X(NU)
6 Z(I)=Y(NU)
7 CALL CIP(U,V,A)
CALL CIP(W,Z,B)
IF(ABS(A(3)-B(3)).LT..0001) GO TO 11
XST = (B(2)-A(2))*5/(A(3)-B(3))
YS = A(2)+2.*A(3)*XST
IF(ABS(YS)-.0001)9,8,8
8 TA = 1./SQRT(1.+YS*YS)
DO 10 I = 1,3
XQ = TA*(U(I)+V(I)*YS)
V(I) = TA*(V(I)-YS*U(I))
U(I) = XQ
XQ = TA*(W(I)+YS*Z(I))
Z(I) = TA*(Z(I)-YS*W(I))
10 W(I) = XQ
GO TO 7
9 D = A(1)-B(1)+(A(2)-B(2)+(A(3)-B(3))*XST)*XST
11 RETURN
END

```

```
FUNCTION SING(A)  
SING =DSIN(A*.0174532925199DC0)  
RETURN  
END
```

```
FUNCTION COSG(A)  
CCSG =DCCS(A*.0174532925159D00)  
RETURN  
END
```

```
FUNCTION RUND(A,B)  
RUND = (AINT(A*B+SIGN(.5,A)))/B  
RETURN  
END
```

```
FUNCTION CSLG(A,B)  
CSLG = ALOG(ABS(SING(A-B)))  
RETURN  
END
```