

N80-29254

A Computer Program for the Design and Analysis of Low-Speed Airfoils

Richard Eppler, et al

NASA Langley Research Center
Hampton, Virginia

August 1980

(NASA-TM-80210) A COMPUTER PROGRAM FOR THE
DESIGN AND ANALYSIS OF LOW-SPEED AIRFOILS
(NASA) 145 p HC A07/MF A01 CSCL 01A

N80-29254

Unclas

G3/02 28437

NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM THE BEST COPY FURNISHED US BY THE SPONSORING AGENCY. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE.

NASA Technical Memorandum 80210

A Computer Program for the Design
and Analysis of Low-Speed Airfoils

Richard Eppler
University of Stuttgart
Stuttgart, West Germany

Dan M. Somers
Langley Research Center
Hampton, Virginia

NASA

National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1980

1-a

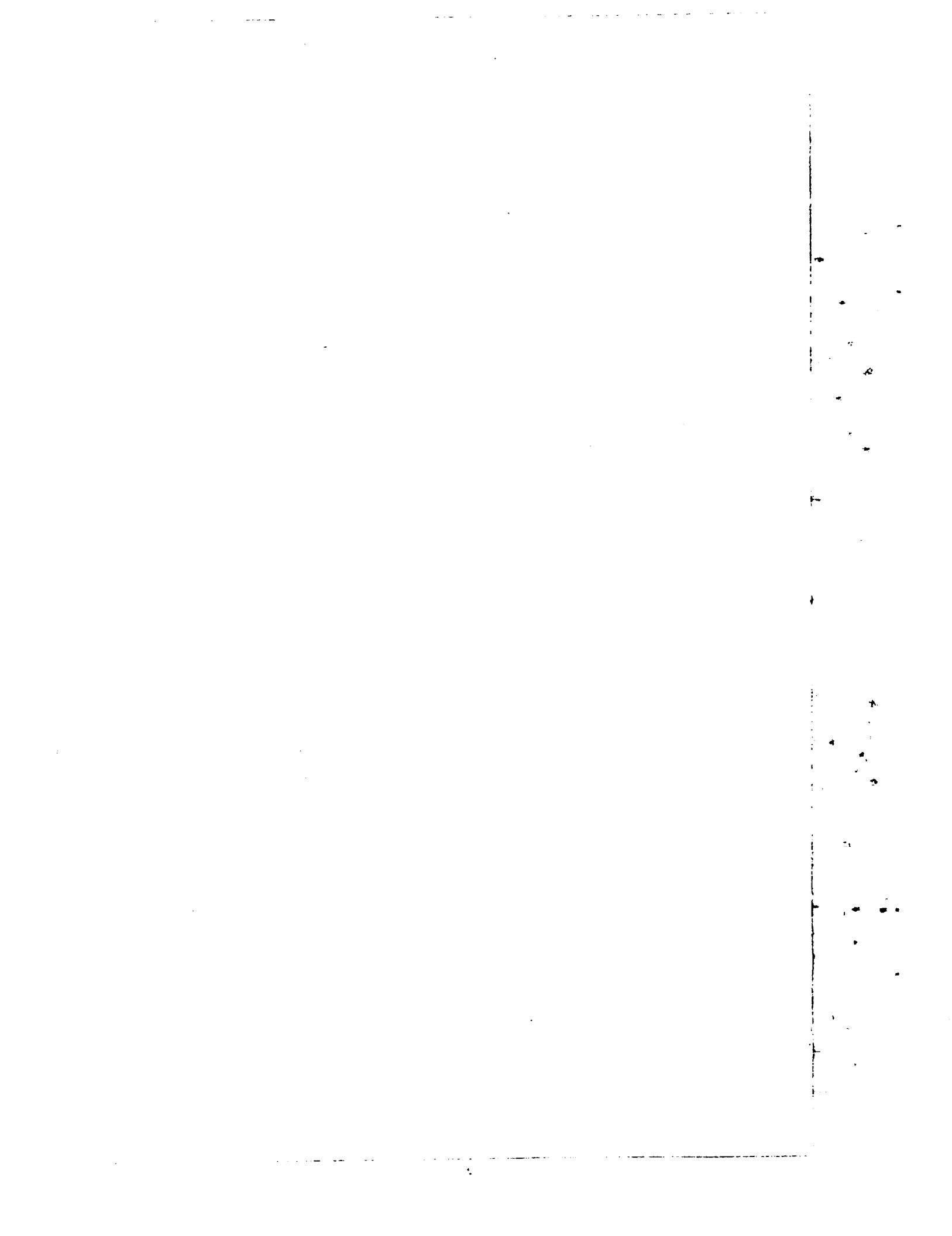


TABLE OF CONTENTS

	Page
<u>SUMMARY</u>	1
<u>INTRODUCTION</u>	1
<u>SYMBOLS</u>	2
<u>THEORY</u>	8
POTENTIAL-FLOW AIRFOIL DESIGN METHOD	8
Conformal-Mapping Algorithm	8
Specification of Velocity Distribution	14
Symmetric Airfoils	17
Nonsymmetric Airfoils	19
POTENTIAL-FLOW AIRFOIL ANALYSIS METHOD	20
General Description	20
Numerical Procedure	20
Trailing-Edge Singularity	23
Spline Routine	24
BOUNDARY-LAYER METHOD	24
General Boundary-Layer Equations	24
Laminar Boundary Layers	26
Turbulent Boundary Layers	28
Transition	28
Laminar Separation Bubbles	29
Boundary-Layer Development	30
Section Characteristics	30
<u>IMPLEMENTATION OF THE THEORY</u>	32
AIRFOIL DESIGN SUBROUTINE	33
AIRFOIL ANALYSIS SUBROUTINES	33
BOUNDARY-LAYER SUBROUTINES	33
ADDITIONAL RELATED SUBROUTINES	36
UTILITY SUBROUTINES	37
<u>INPUT AND OUTPUT (USER'S GUIDE)</u>	37
PRINCIPLES OF INPUT	37
FORMAT OF DATA CARDS	38
OVERVIEW OF INPUT CARDS	39
INPUT FOR AIRFOIL DESIGN	40
TRAL Card	41
TRA2 Card	42
ABSZ Card	44
INPUT FOR AIRFOIL ANALYSIS	45
FXPR Card	45
PAN Card	47
FLAP Card	48

Preceding page blank

	Page
OPTIONS FOR BOTH DESIGN AND ANALYSIS MODES	48
ALFA Card	48
PUXY Card	51
DIAG Card	52
STRD Card	54
STRK Card	54
OPTIONS FOR BOUNDARY-LAYER ANALYSIS	56
RE Card	56
FLZW Card	61
PLW Card	62
PLWA Card	64
CDCL Card	66
FLOW CHART AND INPUT-CARD SUMMARY	69
<u>SYSTEM-DEPENDENT FEATURES OF THE PROGRAM</u>	74
LANGUAGE	74
CORE-STORAGE REQUIREMENTS	74
INPUT AND OUTPUT	75
PLOTTING SOFTWARE	75
<u>CONCLUDING REMARKS</u>	77
<u>APPENDIX - PROGRAM LISTING</u>	78
<u>REFERENCES</u>	126
<u>FIGURES</u>	127

SUMMARY

A conformal-mapping method for the design of airfoils with prescribed velocity-distribution characteristics, a panel method for the analysis of the potential flow about given airfoils, and a boundary-layer method have been combined. With this combined method, airfoils with prescribed boundary-layer characteristics can be designed and airfoils with prescribed shapes can be analyzed. All three methods are described briefly.

A FORTRAN IV computer program for the numerical evaluation of these methods is available through COSMIC. The program and its input options are described in detail. A complete listing is given as an appendix.

INTRODUCTION

The flow about an airfoil in free air can be described approximately by a boundary-layer flow near the surface of the airfoil and by a potential flow everywhere else. Boundary-layer theory can be applied to the flow about an airfoil in two ways. First, the boundary-layer development can be determined for a given potential-flow velocity distribution. This is the direct or analysis problem. Second, the potential-flow field, or at least some of its properties, can be determined for a given boundary-layer development. This is the inverse or design problem. This second application of boundary-layer theory requires the solution of the inverse potential-flow problem where the potential-flow velocity distribution is specified and the airfoil shape is computed. Thus, the viscous airfoil design (inverse) problem can be described as the computation of a shape from a potential-flow velocity distribution which is consistent with a desired boundary-layer development. The first application of this inverse procedure has been described in references 1 and 2. Because Tollmien, Schlichting, Ulrich, Pretsch, and others (see ref. 3) had shown that favorable pressure gradients delay the transition from laminar to turbulent flow, airfoils were designed with aft pressure recoveries. The experimental results for these airfoils confirmed the theoretical predictions. This breakthrough led to the laminar-flow airfoil series.

Since that time, boundary-layer and potential-flow theories have been steadily improved. These theories have been used increasingly to supplement wind-tunnel tests. Today, computing costs are so low that a complete potential-flow and boundary-layer analysis of an airfoil costs considerably less than 1 percent of the equivalent wind-tunnel test.

Different computer programs have been developed for low-speed (incompressible) airfoils. (For example, see ref. 4.) The present paper describes one of these programs. The potential-flow inverse problem still plays a major role in airfoil design. This problem has been solved exactly by means of conformal mapping as shown in reference 5. The method is similar to that of

Lighthill (ref. 6), is direct, and solves most multipoint design problems in a very simple manner. A potential-flow analysis method is also required for comparison with wind-tunnel tests of given airfoils and for analyses of airfoils generated by the design method and then modified by a flap deflection. The airfoil analysis problem is solved using a distributed surface-singularity method similar to those described in references 4 and 7. Some of the details of this method are new and previously unpublished. The boundary-layer method, which uses integral momentum and energy equations, is described in reference 8. The present method does not contain a boundary-layer displacement iteration. The program is very efficient and has been successfully applied at Reynolds numbers from 2.0×10^4 to 1.0×10^8 . (See ref. 9 for example.)

The second part of this paper is a user's guide for the computer program. The input is described in detail, and an overview of the program structure and subroutines is given. A complete listing is contained in the appendix.

Use of trade names or names of manufacturers in this report does not constitute an official endorsement of such products or manufacturers, either expressed or implied, by the National Aeronautics and Space Administration.

SYMBOLS

Values are given in SI Units.

A_p	parasite-drag area, m^2
AR	aspect ratio
a_m, b_m	power-series coefficients
$C_{D,i}$	induced-drag coefficient
$C_{D,p}$	parasite-drag coefficient
C_D	boundary-layer dissipation coefficient
C_f	boundary-layer skin-friction coefficient
C_L	lift coefficient
C_p	pressure coefficient
c	airfoil chord, m
c^*	reference airfoil chord, m
c_d	section profile-drag coefficient
c_f	flap chord, m

c_h	section hinge-moment coefficient about flap-hinge point
c_l	section lift coefficient
$c_{l,ns}$	section lift coefficient without correction due to separation
c_m	section pitching-moment coefficient about quarter-chord point
D^*	dissipation function for laminar boundary layer
F	complex potential function
g	acceleration due to gravity, m/s^2
g_1	slope of panel at beginning of panel
g_2	slope of panel at end of panel
H_{12}	boundary-layer shape factor, δ_1/δ_2
H_{12}^*	modified boundary-layer shape factor for calculation of section profile-drag coefficient
H_{32}	boundary-layer shape factor, δ_3/δ_2
I_a	total number of airfoil segments
i	$= \sqrt{-1}$
K	constant in upper-surface pressure-recovery function
\bar{K}	constant in lower-surface pressure-recovery function
K_H	exponent of upper-surface closure contribution
\bar{K}_H	exponent of lower-surface closure contribution
K_R	desired value of K_S
K_S	sum of exponents of upper- and lower-surface closure contributions, $K_H + \bar{K}_H$
K_{tol}	tolerance on achievement of K_R
l	length, m
p	variable number
p_c	number of points on ζ unit circle
p_p	number of subpanels in one panel

n_q number of points on airfoil, $n_c + 1$
 P real part of $\ln \frac{dz}{d\zeta} - \ln \left(1 - \frac{1}{\zeta}\right)$ on circle $e^{i\phi}$; point
 Q conjugate harmonic function of $P(\phi)$
 R Reynolds number based on free-stream conditions and airfoil chord
 R_{δ_2} Reynolds number based on local conditions and boundary-layer momentum thickness
 r roughness factor
 S wing area, m^2
 S^* reference wing area, m^2
 s source distribution; arc length along airfoil surface, m
 s_{sep} arc length along which boundary layer is separated, m
 s_T arc length for transition from forward portion of airfoil to flap, m
 s_{turb} arc length along which boundary layer is turbulent including s_{sep} , m
 t airfoil thickness, m
 t^* reference airfoil thickness, m
 U potential-flow velocity, m/s
 $U' = \frac{dU}{dx}, s^{-1}$
 U_∞ free-stream velocity, m/s
 u x-component of boundary-layer velocity, m/s
 V aircraft speed, m/s
 V_{max} maximum aircraft speed, m/s
 V_1 aircraft speed for $C_L = 1$, m/s
 v absolute value of complex velocity; local velocity on airfoil, m/s
 v^* local velocity on airfoil for velocity specification in design method, m/s
 v_0 normal velocity at surface in boundary-layer method, m/s

W	aircraft mass, kg
W*	reference aircraft mass, kg
w	pressure-recovery function
\bar{w}	complex velocity, $ve^{-i\theta}$
\vec{w}	vector of velocity induced by panel at point (ξ, η)
w_t	tangential component of velocity induced by panel at end of panel
w_η	η -component of \vec{w}
w_ξ	ξ -component of \vec{w}
x	airfoil abscissa, m; axis in streamwise direction, tangential to surface in boundary-layer method
x_s	chord location of beginning of closure-contribution region, m
x_T	chord location of transition, m
x_w	chord location of beginning of main pressure-recovery region, m
y	airfoil ordinate, m; axis normal to surface in boundary-layer method
z	complex variable, $x + iy$
α	angle of attack relative to zero-lift line, deg
α^*	angle of attack relative to zero-lift line for velocity specification in design method, deg
α_c	angle of attack relative to chord line, deg
α_i^*	α^* for i th segment of ζ circle or airfoil, deg
$\alpha_{i,le}^*$	α^* for last segment of upper surface, deg
α_0	angle of zero lift, deg
β	inner normal angle, rad
Γ	circulation
γ	vorticity
γ_i	vorticity at beginning of panel
γ_j	vorticity at end of panel

γ_p	parabolic-vorticity factor
Δ	incremental change in quantity
Δc_l	section lift-coefficient correction due to separation
$\Delta \alpha$	angle-of-attack correction due to separation, deg
$\Delta \phi$	angle between points on ζ unit circle, $\frac{2\pi}{n_c}$
δ	slope of airfoil surface near trailing edge, rad
δ_f	flap deflection, positive downward, deg
δ_1	boundary-layer displacement thickness
δ_2	boundary-layer momentum thickness
δ_2'	$= \frac{d\delta_2}{dx}$
δ_3	boundary-layer energy thickness
δ_3'	$= \frac{d\delta_3}{dx}$
ε^*	skin-friction function for laminar boundary layer
ζ	complex variable
η	panel ordinate
θ	argument of complex velocity; twist angle, deg
λ	$= \frac{\phi_w}{\Delta\phi}$
$\bar{\lambda}$	$= \frac{\bar{\phi}_w}{\Delta\phi}$
λ^*	$= \frac{\phi_s}{\Delta\phi}$
$\bar{\lambda}^*$	$= \frac{\bar{\phi}_s}{\Delta\phi}$
μ	exponent of upper-surface main pressure recovery
$\bar{\mu}$	exponent of lower-surface main pressure recovery
ν	kinematic viscosity, m^2/s

v_i	$= \frac{\phi_i}{\Delta\phi}$
$v_{i,le}$	$= \frac{\phi_{i,le}}{\Delta\phi}$
ξ	panel abscissa
ρ	air density, kg/m^3
σ	panel length
τ	shear stress, $\text{kg/m}\cdot\text{s}^2$
τ_0	skin friction, $\text{kg/m}\cdot\text{s}^2$
τ_1	angle between ξ -axis and line from beginning of panel to point (ξ, η) , rad
τ_2	angle between ξ -axis and line from end of panel to point (ξ, η) , rad
ϕ	argument of ζ
ϕ_{I_a}	arc limit of lower-surface trailing edge
ϕ_i	arc limit on ζ unit circle
$\phi_{i,le}$	arc limit of leading edge
ϕ_s	arc limit of beginning of upper-surface closure-contribution region
$\bar{\phi}_s$	arc limit of beginning of lower-surface closure-contribution region
ϕ_w	arc limit of beginning of upper-surface main pressure-recovery region
$\bar{\phi}_w$	arc limit of beginning of lower-surface main pressure-recovery region
ψ	integration variable
ω	total amount of upper-surface main pressure recovery
$\bar{\omega}$	total amount of lower-surface main pressure recovery
ω'	initial slope of upper-surface main pressure recovery
$\bar{\omega}'$	initial slope of lower-surface main pressure recovery
∞	infinity

Subscripts:

h	hinge
i, j	index; 1, 2, 3, ..., ∞
le	leading edge
lim	limit
ls	lower surface
max	maximum
min	minimum
sep	separation
te	trailing edge
us	upper surface

THEORY

POTENTIAL-FLOW AIRFOIL DESIGN METHOD

Conformal-Mapping Algorithm

The airfoil design method is based on conformal mapping as are several other methods. (For example, see ref. 6.) In the complex ζ -plane, the unit circle is in an infinite flow of velocity 1 at an angle of attack α . (See fig. 1.) If the rear stagnation point is at $\zeta = 1$, the complex potential of the flow is

$$\left. \begin{aligned} F(\zeta) &= e^{-i\alpha}\zeta + e^{i\alpha}\zeta^{-1} - \frac{\Gamma}{2\pi i} \ln \zeta \\ \frac{dF}{d\zeta} &= e^{-i\alpha} - e^{i\alpha}\zeta^{-2} - \frac{\Gamma}{2\pi i} \zeta^{-1} \end{aligned} \right\} \quad (1)$$

By substituting $\Gamma = 4\pi \sin \alpha$ where Γ is the circulation, $dF/d\zeta$ can be written

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

This flow is conformally mapped into the z -plane by a complex function $z(\zeta)$ such that

$$z(\infty) = \infty \quad \text{and} \quad \left(\frac{dz}{d\zeta}\right)_{\infty} = 1 \quad (3)$$

This means that the ζ unit circle, as mapped into the z -plane, is also in an infinite flow of velocity 1 at an angle of attack α , just as it was in the ζ -plane. The conjugated complex velocity vector \bar{w} in the z -plane is

$$\bar{w} = ve^{-i\theta} = \frac{dF}{dz} = \frac{dF/d\zeta}{dz/d\zeta} \quad (4)$$

where v is the absolute value of the complex velocity and θ is its argument. The function $dF/d\zeta$ is known from equation (1).

The mathematical problem is to determine the function $z(\zeta)$ such that the velocity v is a prescribed function on the airfoil. This function is not specified in x but rather on the circle $\zeta = e^{i\phi}$ as $v(\phi)$. The variables x and ϕ are related by

$$x \approx \frac{c}{2}(1 + \cos \phi) \quad (5)$$

where c is the airfoil chord. This approximation is only exact for a flat plate, and it becomes less accurate as thickness and camber increase. If the resulting function $x(\phi)$ differs too much from equation (5), an iteration can be performed after specifying a new function $v(\phi(x))$. This iteration, however, has not been required in the practical application of the method.

If $v(\phi)$ is given, the mathematical design problem can be solved by taking the logarithm of equation (4)

$$\ln \bar{w} = \ln v - i\theta = \ln \frac{dF}{d\zeta} - \ln \frac{dz}{d\zeta} \quad (6)$$

At the boundary $\zeta = e^{i\phi}$, the real part of $\ln \frac{dz}{d\zeta}$ is known because $v(\phi)$ and $\frac{dF}{d\zeta}$ are known. Thus, the function $\frac{dz}{d\zeta}$ which maps the circle into the airfoil can be determined from fundamental complex-variable theory. The problems which remain are to derive an algorithm for numerical evaluation and to introduce $v(\phi)$ in such a manner that practical design problems can be solved easily.

The algorithm is based upon the introduction of $\ln \frac{dz}{d\zeta}$ in the form

$$\ln \frac{dz}{d\zeta} = \ln \left(1 - \frac{1}{\zeta} \right) + \sum_{m=0}^{\infty} (a_m + ib_m) \zeta^{-m} \quad (7)$$

This form has the advantage that the power series does not have an infinite value at any point on the boundary $\zeta = e^{i\phi}$. Use of equation (6) allows the function $\ln \frac{dz}{d\zeta} - \ln \left(1 - \frac{1}{\zeta} \right)$ along the boundary $\zeta = e^{i\phi}$ to be written as

$$\begin{aligned} \left[\ln \frac{dz}{d\zeta} - \ln \left(1 - \frac{1}{\zeta} \right) \right]_{\zeta=e^{i\phi}} &= P(\phi) + iQ(\phi) \\ &= -\ln v(\phi) + i\theta + \ln \left(\frac{dF}{d\zeta} \right)_{\zeta=e^{i\phi}} - \ln(1 - e^{-i\phi}) \end{aligned} \quad (8)$$

Use of equations (2) and (7) enables the real part of equation (8) to be evaluated as

$$\sum_{m=0}^{\infty} (a_m \cos m\phi + b_m \sin m\phi) = P(\phi) = -\ln \left[\frac{v(\phi)}{2 \left| \cos \left(\frac{\phi}{2} - \alpha \right) \right|} \right] \quad (9)$$

Thus, the coefficients a_m and b_m of equation (7) can be determined in the same manner as coefficients of a real Fourier series. For the final procedure, these coefficients need not be computed. Only the conjugate harmonic function $Q(\phi)$ is required. It is determined by

$$\sum_{m=0}^{\infty} (b_m \cos m\phi - a_m \sin m\phi) = Q(\phi) = \frac{1}{2\pi} \int_0^{2\pi} P(\psi) \cot \frac{\psi - \phi}{2} d\psi \quad (10)$$

By inserting equation (9) into equation (8), it follows that

$$\begin{aligned} \ln \left(\frac{dz}{d\zeta} \right)_{e^{i\phi}} &= -\ln \frac{v(\phi)}{2 \left| \cos \left(\frac{\phi}{2} - \alpha \right) \right|} + iQ + \ln(1 - e^{-i\phi}) \\ &= \ln \frac{2 \left| \cos \left(\frac{\phi}{2} - \alpha \right) \right|}{v(\phi)} + iQ + \ln \left(2ie^{-i\phi/2} \sin \frac{\phi}{2} \right) \end{aligned}$$

or

$$\left(\frac{dz}{d\zeta}\right)_{e^{i\phi}} = 4i \sin \frac{\phi}{2} \left| \cos \left(\frac{\phi}{2} - \alpha \right) \right| \frac{1}{v(\phi)} e^{i\left(Q - \frac{\phi}{2}\right)}$$

For $d\zeta = ie^{i\phi} d\phi$, this yields

$$\frac{dz}{d\phi} = ie^{i\phi} \left(\frac{dz}{d\zeta}\right)_{e^{i\phi}} = -4 \sin \frac{\phi}{2} \left| \cos \left(\frac{\phi}{2} - \alpha \right) \right| \frac{1}{v(\phi)} e^{i\left(Q + \frac{\phi}{2}\right)}$$

Splitting this equation into real and imaginary parts results directly in

$$\frac{dx}{d\phi} = -4 \sin \frac{\phi}{2} \left| \cos \left(\frac{\phi}{2} - \alpha \right) \right| \frac{1}{v(\phi)} \cos \left[\frac{\phi}{2} + Q(\phi) \right] \quad (11)$$

and

$$\frac{dy}{d\phi} = -4 \sin \frac{\phi}{2} \left| \cos \left(\frac{\phi}{2} - \alpha \right) \right| \frac{1}{v(\phi)} \sin \left[\frac{\phi}{2} + Q(\phi) \right] \quad (12)$$

If $Q(\phi)$ is determined, the airfoil coordinates $x(\phi)$ and $y(\phi)$ can be computed by simple quadrature.

As in every inverse method, the velocity distribution $v(\phi)$ is not completely arbitrary. At infinity, equations (3) must be satisfied. This requires that the coefficient of ζ^0 in $dz/d\zeta$ from the expansion of equation (7) be equal to 1. Furthermore, equations (11) and (12) must result in a closed airfoil. This requires that the coefficient of ζ^{-1} in $dz/d\zeta$ be equal to 0. (See eq. (7).) From equation (7), it follows that

$$\frac{dz}{d\zeta} = e^{a_0 + ib_0} \left[1 + (-1 + a_1 + ib_1)\zeta^{-1} + \dots \right]$$

The above conditions are only satisfied if

$$a_0 = 0$$

$$b_0 = 0$$

$$a_1 = 1$$

$$b_1 = 0$$

Because b_0 does not appear in the Fourier expansion of $P(\phi)$, the remaining three conditions from equation (9) are:

$$\pi a_0 = \int_0^{2\pi} P(\phi) d\phi = 0 \quad (13)$$

$$\pi a_1 = \int_0^{2\pi} P(\phi) \cos \phi d\phi = \pi \quad (14)$$

$$\pi b_1 = \int_0^{2\pi} P(\phi) \sin \phi d\phi = 0 \quad (15)$$

The algorithm for the numerical evaluation of the airfoil coordinates is defined by equations (9) to (12) subject to the conditions set forth in equations (13) to (15).

The integration of equations (11) and (12) allows the coordinates of the airfoil to be obtained from $v(\phi)$. The entire procedure is very flexible. In other methods, the function $P(\phi)$ in equation (9) is specified by selecting one angle of attack $\alpha = \alpha^*$ for which the velocity distribution $v^*(\phi, \alpha^*)$ is to occur on the airfoil. In the present method, the function $P(\phi)$ in equation (9) is defined as

$$P(\phi) = -\ln \left[\frac{v^*(\phi, \alpha^*)}{2 \left| \cos \left(\frac{\phi}{2} - \alpha^* \right) \right|} \right] \quad (16)$$

This function $P(\phi)$ specifies the airfoil and is independent of α . As a result, the velocity distribution $v(\phi, \alpha)$ for any arbitrary α is obtained from equation (16),

$$\frac{v(\phi, \alpha)}{\left| \cos \left(\frac{\phi}{2} - \alpha \right) \right|} = \frac{v^*(\phi, \alpha^*)}{\left| \cos \left(\frac{\phi}{2} - \alpha^* \right) \right|} = f(\phi) \quad (17a)$$

which can also be written

$$v(\phi, \alpha) = v^*(\phi, \alpha^*) \left| \frac{\cos\left(\frac{\phi}{2} - \alpha\right)}{\cos\left(\frac{\phi}{2} - \alpha^*\right)} \right| \quad (17b)$$

Equation (17a) also follows directly from equations (1) and (4). The velocity distribution $v(\phi, \alpha)$ is therefore defined by $v^*(\phi, \alpha^*)$ and α^* without any knowledge of the airfoil shape.

It should be noted that the right-hand side of equation (17a) is invariant with respect to α . One of the most important consequences of this invariance is that it is not necessary to select only one angle of attack α^* at which the velocity distribution v^* is to occur on the airfoil. It is possible to specify v^* by selecting different values of α^* for different segments of the airfoil. If α^* is a function of ϕ , equations (16) and (17a) still hold. For any given segment, the velocity $v(\phi, \alpha)$ for any α is only identical to the specified $v^*(\phi, \alpha^*(\phi))$ if $\alpha = \alpha^*(\phi)$.

Of course, $v(\phi, \alpha)$ must be continuous over the entire airfoil ($0 \leq \phi \leq 2\pi$). This does not require $\alpha^*(\phi)$ to be continuous. It is only necessary that $v^*(\phi, \alpha^*)$ compensate for any discontinuity in $\alpha^*(\phi)$. If $\alpha^*(\phi)$ is introduced as piecewise constant

$$\alpha^*(\phi) = \alpha_i^* \quad (\phi_{i-1} \leq \phi \leq \phi_i)$$

at each discontinuity in $\alpha^*(\phi)$, the matching condition

$$\lim_{\epsilon \rightarrow 0} \left| \frac{v^*(\phi_{i-1} + \epsilon, \alpha_{i+1}^*)}{\cos\left(\frac{\phi_{i-1}}{2} - \alpha_{i+1}^*\right)} \right| = \lim_{\epsilon \rightarrow 0} \left| \frac{v^*(\phi_i - \epsilon, \alpha_i^*)}{\cos\left(\frac{\phi_i}{2} - \alpha_i^*\right)} \right|$$

must hold.

A problem arises with the velocity distribution at the front stagnation point. If segment i includes the front stagnation point, $v^*(\phi, \alpha_i^*) = 0$ must be specified for $\cos\left(\frac{\phi}{2} - \alpha_i^*\right) = 0$. Thus, $P(\phi)$ is undefined at this point. (See eq. (16).) Because this is only true at an isolated point, the problem could be solved numerically. It is much easier, however, to prevent the problem by introducing an arc limit $\phi_{i,le}$ near the leading edge and specifying v^* ahead of this point while the stagnation point is beyond it and vice versa. This is possible if

$$\left. \begin{array}{l} \phi_{i,le} - 2\alpha_{i,le}^* < \pi \\ \phi_{i,le} - 2\alpha_{i,le+1}^* > \pi \end{array} \right\} \quad (18a)$$

and

which requires

$$\alpha_{i,le+1}^* < \alpha_{i,le}^* \quad (18b)$$

In the present method, $\phi_{i,le}$ must not be specified. It is one of the parameters left undefined in order to satisfy the conditions set forth in equations (13) to (15). The algorithm is written such that those conditions are satisfied automatically if inequality (18b) is satisfied at the leading edge.

Specification of Velocity Distribution

The final specification of $\alpha^*(\phi)$ and $v(\phi, \alpha^*)$ has the following form. The circle is divided into I_a arcs. Each arc, i , extends from ϕ_{i-1} to ϕ_i . Thus, $\phi_0 = 0$ and $\phi_{I_a} = 2\pi$. The leading edge is $\phi_{i,le}$. For each arc,

$$v(\phi, \alpha^*) = v_i w(\phi) \quad (\phi_{i-1} \leq \phi \leq \phi_i) \quad (19)$$

where v_i is a constant to be discussed later and $w(\phi)$ is a function which is independent of i . This function $w(\phi)$ allows a main pressure recovery and a closure contribution to be introduced on each surface (ref. 5). (See fig. 2.) It has the following form:

$$w(\phi) = \left[1 + K \left\{ \frac{\cos \phi - \cos \phi_w}{1 + \cos \phi_w} \right\} \right]^{-\mu} \left[1 - 0.36 \left\{ \frac{\cos \phi - \cos \phi_s}{1 - \cos \phi_s} \right\}^2 \right]^{K_H} \quad (20)$$

over the range $(0 \leq \phi \leq \phi_{i,le})$, and

$$w(\phi) = \left[1 + \bar{K} \left\{ \frac{\cos \phi - \cos \bar{\phi}_w}{1 + \cos \bar{\phi}_w} \right\} \right]^{-\bar{\mu}} \left[1 - 0.36 \left\{ \frac{\cos \phi - \cos \bar{\phi}_s}{1 - \cos \bar{\phi}_s} \right\}^2 \right]^{\bar{K}_H} \quad (21)$$

over the range $(\phi_{i,le} < \phi \leq 2\pi)$. The terms in braces must be considered special functions which are interpreted as follows: if $f(\phi) \leq 0$, $\{f(\phi)\} = 0$ and if

$f(\phi) > 0$, $\{f(\phi)\} = f(\phi)$. Thus, $w(\phi) = 1$ if $\phi > \phi_w$ and $\phi > \phi_s$ in equation (20) and if $\phi < \bar{\phi}_w$ and $\phi < \bar{\phi}_s$ in equation (21). (It should be noted that ϕ_w and $\phi_s < \pi$ and $\bar{\phi}_w$ and $\bar{\phi}_s > \pi$.)

A typical distribution w is sketched in figure 2 where $\frac{x}{c} = \frac{1}{2}(1 + \cos \phi)$ is used to give an impression of $w(x)$ as defined by equations (20) and (21). The leading factors in equations (20) and (21) represent the main pressure recoveries for the upper and lower surfaces, respectively. The lengths of the main pressure recoveries are specified by ϕ_w for the upper surface and by $\bar{\phi}_w$ for the lower surface. The total amount of pressure recovery and the shape of the recovery are determined by K and μ or \bar{K} and $\bar{\mu}$ depending on the surface under consideration. The total amount of main pressure recovery (the total main velocity-reduction factor) for the upper surface is obtained by neglecting the second factor in equation (20)

$$\omega = \left[1 + K \left(\frac{1 - \cos \phi_w}{1 + \cos \phi_w} \right) \right]^{-\mu} \quad (22)$$

The absolute value of the slope of $w(x)$ at the beginning of the recovery ($\phi = \phi_w$) is

$$\omega' = \frac{\mu K}{\frac{1}{2}(1 + \cos \phi_w)} \quad (23)$$

A more detailed analysis of equation (20) shows that a concave pressure recovery is specified by $K > 0$ and $\mu > 0$. An approximately linearly decreasing velocity is achieved by $K < 0$ and $\mu = -1$. A pressure recovery having a very steep initial slope is specified by a large value for K and a small positive value for μ . Thus, it is possible to specify almost exactly a Stratford pressure recovery by a careful choice of K and μ in equation (20).

It can be seen that the first denominator in equation (20) is $(1 + \cos \phi_w)$ instead of $(1 - \cos \phi_w)$ and that ω contains ϕ_w in equation (22). These terms cause a smaller total amount of pressure recovery (higher ω) with decreasing recovery length (decreasing ϕ_w). This is insignificant in that the main pressure recovery can be specified in three different modes:

- (1) K and μ are specified
- (2) ω' and ω are specified (only concave pressure recoveries (K and $\mu > 0$) are specified in this mode)
- (3) μ and ω are specified

The length of the recovery is always specified by ϕ_w . Of course, the same three modes apply to the lower surface as well for \bar{K} , $\bar{\mu}$, and $\bar{\phi}_w$.

The second factor in both equations (20) and (21) represents the closure contribution. It is normally restricted to a small region surrounding the trailing edge ($\phi_s \approx 20^\circ-30^\circ$ and $\bar{\phi}_s \approx 330^\circ-340^\circ$). If $K_H > 0$, this term results in a convex velocity decrease as shown in figure 2. This type of velocity decrease is frequently present near the trailing edge.

If the trailing-edge angle is nonzero, the velocity goes to zero at the trailing edge (rear stagnation point). This condition cannot be achieved with equations (20) and (21) as it would require K_H and \bar{K}_H to become infinite. Hence, these equations mathematically permit only a zero trailing-edge angle. However, the region influenced by the requirement for a trailing-edge stagnation point is very small so that, in a practical sense, the trailing edge as defined by the method can be considered as having a nonzero angle. The exponents K_H and \bar{K}_H cannot be specified. They are determined such that the airfoil will be closed and, therefore, they are a consequence of all of the specifications.

In summary, the velocity specification, equation (19), along with equations (20) and (21) allow several different arcs on the unit circle in the ζ -plane to be specified as shown in figure 3. The limits of these arcs are shown as solid radial lines. The angles ϕ_w , ϕ_s , $\bar{\phi}_w$, and $\bar{\phi}_s$ are indicated by broken radial lines. For all arcs between ϕ_w and $\bar{\phi}_w$, the essential part of the velocity specification is the value α_i^* . Thus, for any given arc i , v is specified such that

$$v(\phi, \alpha_i^*) = v_i = \text{Constant}$$

which means that for this arc i , $v = v_i = \text{Constant}$ if $\alpha = \alpha_i^*$. The absolute value of v_i cannot be specified, but is determined by the previously mentioned matching conditions in the form of equation (17a).

For the arcs over which $w(\phi) \neq 1$, the value of α^* must also be specified. For these arcs, α^* is that angle of attack for which $v(\phi)$ varies with $w(\phi)$. (See eq. (19).) Normally there is no arc limit specified within the pressure-recovery region. This is, however, not a restriction within the algorithm. One could imagine that there will be design problems in which α^* should change within the pressure-recovery region. This is not prohibited.

If two consecutive arcs have the same α^* , no dividing arc limit is necessary. This is also true if α^* is the same for the pressure recovery and the arc immediately ahead of it. Of course, specifying two or more consecutive arcs having the same α^* is not prohibited, except around the leading edge. The values of ϕ_w and $\bar{\phi}_w$ do not necessarily need to be specified at arc limits, although the pressure recovery frequently is not specified for the same angle of attack at which constant v occurs over the arc ahead of the pressure-recovery region itself. This is illustrated in figure 3 where ϕ_w is an arc limit and $\bar{\phi}_w$ is not. This means that α^* can change at ϕ_w , but not at $\bar{\phi}_w$.

Thus the total specification of $v(\phi)$ requires only

- (1) The arc limits ϕ_i (except $\phi_{i,le}$)
- (2) The α_i^* values for all arcs
- (3) The pressure-recovery specifications - ϕ_w , μ , K , $\bar{\phi}_w$, $\bar{\mu}$, and \bar{K}
- (4) The lengths of the closure contributions as specified by ϕ_s and $\bar{\phi}_s$

This specification leaves the I_a constants v_i free, along with the arc limit $\phi_{i,le}$ and the closure-contribution exponents K_H and \bar{K}_H . There are I_a matching conditions and equations (13) to (15) must be satisfied. All these conditions can be satisfied in closed form which leads to a transcendental equation for $\phi_{i,le}$. The solution of this equation can be obtained by normal numerical methods. The values of K_H , \bar{K}_H , and v_i can also easily be determined from the solution of the transcendental equation.

A more extensive description of the method is given in reference 5, which contains the formula for the splitting of a term containing a "corner" (discontinuity in slope) at $\phi = \phi_{i,le}$. (See $P(\phi)$ in eq. (9).) This results in a better convergence of the remaining power series and reduces the numerical errors in the method. Some minor differences exist between reference 5 and the present paper. In reference 5,

- (1) Only $I_a = 6$ arcs are available with the leading edge occurring at ϕ_3
- (2) The closure-contribution exponents K_H and \bar{K}_H also include the main pressure recovery

Symmetric Airfoils

A simple example of an airfoil design is that of the symmetric NACA 6-series airfoils (refs. 1 and 2). These airfoils are intended to exhibit, at a certain lift coefficient $c_{l,1}$ or positive angle of attack α_1 , constant velocity from $x = 0$ to $x = l_1$. Over this length, the pressure gradient is favorable for $\alpha < \alpha_1$. The same is true for the corresponding length on the lower surface for $\alpha > -\alpha_1$. For all α where $|\alpha| < \alpha_1$, the pressure gradients on both surfaces are favorable up to $x = l_1$. It was predicted that, at these angles of attack over a certain Reynolds number range, the boundary layers on both surfaces would remain laminar to $x = l_1$. This was, of course, experimentally realized. The c_d - c_l polars of these airfoils exhibit what is now commonly referred to as a "laminar bucket." For simplicity, the range of α for which both surfaces exhibit favorable pressure gradients will be called the laminar bucket. Thus, for the symmetric airfoils described above, the laminar bucket is determined by the range $-\alpha_1 \leq \alpha \leq \alpha_1$. The length l_1/c is called the extent of laminar flow. This length might be different on the upper and lower surfaces of non-symmetric airfoils. In this situation, the lower-surface length would be called l_2/c .

For $\alpha = 0^\circ$, the velocity distribution exhibits increasing velocity up to $x = l_1$. This requires a certain airfoil thickness and, correspondingly, a certain pressure recovery towards the trailing edge. The total amount of recovery depends on the width of the laminar bucket as defined by α_1 and on the extent of laminar flow l_1/c .

Using the present method, this NACA 6-series example can be designed very simply. For the upper surface, $\alpha^* = \alpha_1$ and, for the lower surface, $\alpha^* = -\alpha_1$. Unfortunately, the specification of the recovery function for a given α_1 and l_1/c , such that a reasonable trailing edge results, is not so simple. For example, $\phi_w = \frac{\pi}{2}$ for $l_1/c = 0.5$. There is no such rule, however, for the specification of K and μ . If their values are too large, K_H and \bar{K}_H will be negative and a negative thickness will occur ahead of the trailing edge. If their values are too small, K_H and \bar{K}_H will be large and the closure contributions will close the airfoil very abruptly, resulting in a very large trailing-edge angle. Thus, it is usually necessary to select K and μ by trial and error such that the desired trailing-edge shape is obtained. This shape is judged by the value of $K_H + \bar{K}_H$.

This trial-and-error procedure has been included in the program in the form of iteration modes as follows:

- (1) The values of K and/or \bar{K} are iterated such that $K_S = K_H + \bar{K}_H$ where K_S is specified
- (2) The α^* values for the upper and/or lower surface are iterated such that $K_H + \bar{K}_H$ becomes the specified value K_S ; the main pressure recovery is not altered
- (3) The α^* values for arcs i_{1e} and/or $i_{1e} + 1$ are iterated such that $K_H + \bar{K}_H$ becomes the specified value K_S ; the main pressure recovery is not altered

Iteration mode 1 addresses questions like: What amount of pressure recovery is required for the desired laminar bucket width and extent of laminar flow? Iteration mode 2 concerns questions such as: What laminar bucket width and extent of laminar flow are possible given a certain amount of pressure recovery? Iteration mode 3 is normally used in the design of hydrofoils where the α^* values for the segments covering most of the upper and lower surfaces are responsible for the range of lift coefficients over which no suction peak occurs and, therefore, the danger of cavitation is minimized. If this range is not to be influenced by the iteration, only the leading-edge region can be used to modify the airfoil thickness.

Two airfoils which exhibit velocity distributions typical of the symmetric NACA 6-series airfoils are shown in figures 4 and 5. The first airfoil (fig. 4), designated 1095, has the same laminar bucket width and extent of laminar flow as the NACA 65₃-018 airfoil. It is specified by $\alpha^* = 2.7^\circ$ along the upper surface and $\alpha^* = -2.7^\circ$ along the lower surface while the amount of pressure recovery is iterated under mode 1. The second airfoil (fig. 5), 1096, also has

only two segments for which α^* is specified. The difference between the two designs is that, while airfoil 1095 uses iteration mode 1, airfoil 1096 uses iteration mode 2 where the α^* values are iterated leaving the pressure recovery function unchanged. The pressure recovery for airfoil 1096 is about the same as that for the NACA 65₃-018 airfoil. In fact, airfoil 1096 had to be modified slightly to reach an airfoil thickness of 18.0 percent. The result is $\alpha_1 = 3.69^\circ$. This means that the present design method allows an increase in the laminar bucket width of 37 percent.

Nonsymmetric Airfoils

Of course, no more effort is required for the specification and computation of nonsymmetric airfoils than for symmetric ones. The nonsymmetric NACA 6-series airfoils contain an approximation error due to the introduction of camber by vorticity distributions along the mean line.

As an example, a nonsymmetric airfoil, 1098, defined by four segments is shown in figure 6. This airfoil already contains a generalization of the previous NACA 6-series examples. The lower surface again requires only one α^* , $\alpha_4^* = 2^\circ$, which determines the lower limit of the laminar bucket. The upper surface, however, is divided into three segments. For the segment from $x/c \approx 0.15$ to the beginning of the pressure-recovery region at $x_1/c \approx 0.52$, $\alpha_1^* = 8^\circ$. For the shorter segment from $x/c \approx 0.04$ to 0.15, $\alpha_2^* = 10^\circ$ and, for the even shorter segment from the leading edge to $x/c \approx 0.04$, $\alpha_3^* = 12^\circ$. This airfoil, then, should not have a sharp suction peak on the upper surface up to $\alpha = 12^\circ$. This feature is achieved by the third upper-surface segment.

For comparison, airfoil 1097 is shown in figure 7. This airfoil is specified by only one α^* for the upper surface ahead of the pressure-recovery region. All the other specifications, including α^* for the lower surface, the pressure recoveries for both surfaces, and the α^* values for the recoveries, are the same as those for airfoil 1098. The results are $\alpha_2^* = 9.98^\circ$ and an airfoil thickness of 18.45 percent. It should be mentioned that this airfoil was easily specified using iteration mode 3 which iterates only α_2^* .

A comparison of the two airfoils shows that, at $\alpha = 12^\circ$, airfoil 1097 has already developed a sharp suction peak at the leading edge. For many applications, it is beneficial to shift the formation of such peaks to higher angles of attack. This is achieved by specifying higher α^* values for the segments near the leading edge. Of course, this results in a thicker airfoil and, accordingly, a greater amount of pressure recovery. If the amount of pressure recovery is to remain unchanged, however, some other changes have to be made to reduce the thickness. The previous examples show that this can be achieved by introducing lower α^* values for the segments away from the leading edge. This is usually not a disadvantage in that, for moderate Reynolds numbers, the boundary layer will remain laminar even through a moderately adverse pressure gradient.

Airfoils 1097 and 1098 also demonstrate that the pressure gradient at a certain angle of attack α over a certain segment of the airfoil can be precisely controlled by changing the value of α^* for that segment. For the upper

surface, higher values of α^* result in more favorable pressure gradients. Higher values of α^* were used near the leading edge to modify airfoil 1097 to produce airfoil 1098. Lower values of α^* result in more adverse pressure gradients for the upper surface. This feature is the difference between airfoils 1097 and 1098 between $x/c \approx 0.15$ and 0.52 . Obviously, for the lower surface, these trends are reversed.

In summary, the specification of α^* is equivalent to the specification of the pressure gradient. The only difference is that a given change $\Delta\alpha^*$ does not have an equal influence on the pressure gradient everywhere along the airfoil surface; the influence decreases with increasing distance from the leading edge. This is not, however, necessarily a disadvantage. The boundary layer is thinner near the leading edge and, therefore, requires a greater change in pressure gradient to affect a given change in the shape factor. Thus, it is very easy to solve multipoint airfoil design problems by specifying α^* such that the boundary layer has desired properties at various angles of attack.

POTENTIAL-FLOW AIRFOIL ANALYSIS METHOD

General Description

The potential-flow airfoil analysis method employs panels with distributed surface singularities. The geometry of the panels is determined by a spline fit of the airfoil coordinates, with the end points of the panels being the input airfoil coordinates themselves (fig. 8). The singularities used are vorticities distributed parabolically along each panel. The flow condition, which requires the inner tangential velocity to be zero, is satisfied at each airfoil coordinate (i.e., at the end points of the panels, not the midpoints) as indicated by the double arrows in figure 8. Two angles of attack, 0° and 90° , are analyzed. The flow for an arbitrary angle of attack can be derived from these two solutions by superposition. The entire procedure does not require any restrictions on the input point distribution, smoothing, or rearranging of the coordinates; only the original airfoil coordinates are used. An option is included, however, by which additional points can be splined in between the original coordinates. This option allows more precise results to be obtained should a portion of the airfoil have a sparse number of points.

A flap deflection can be introduced by geometrically rotating part of the airfoil about a flap-hinge point. The connection between the forward portion of the airfoil and the flap is defined by an arc consisting of additional points which are generated automatically according to an input arc length.

Numerical Procedure

The numerical procedure is based upon determination of the velocity vector \vec{w} induced at a point (ξ, η) by the vorticity distribution $\gamma(\xi)$ along a straight panel which extends from $\xi = 0$ to $\xi = \sigma$ on the ξ -axis as shown in figure 9. For

$$\gamma(\xi) = \gamma_i \left(1 - \frac{\xi}{\sigma}\right) \quad (24)$$

the velocity components are

$$\frac{2\pi}{\gamma_i} w_\xi = \left(1 - \frac{\xi}{\sigma}\right) T + \frac{\eta}{\sigma} \Lambda \quad (25)$$

and

$$\frac{2\pi}{\gamma_i} w_\eta = -\left(1 - \frac{\xi}{\sigma}\right) \Lambda + \frac{\eta}{\sigma} T - 1 \quad (26)$$

where

$$\Lambda = \ln \frac{\sin \tau_2}{\sin \tau_1}$$

and

$$T = \tau_2 - \tau_1 = \arctan \frac{\eta}{\xi - \sigma} - \arctan \frac{\eta}{\xi}$$

It follows that for

$$\gamma(\xi) = \frac{\xi}{\sigma} \gamma_j \quad (27)$$

the velocity components are

$$\frac{2\pi}{\gamma_j} w_\xi = \frac{\xi}{\sigma} T - \frac{\eta}{\sigma} \Lambda \quad (28)$$

and

$$\frac{2\pi}{\gamma_j} w_\eta = -\frac{\xi}{\sigma} \Lambda - \frac{\eta}{\sigma} T + 1 \quad (29)$$

and for

$$\gamma(\xi) = \frac{\xi}{\sigma} \left(1 - \frac{\xi}{\sigma}\right) \gamma_p \quad (30)$$

the components are

$$\frac{2\pi}{\gamma_p} w_\xi = \left(\frac{\xi}{\sigma} - \frac{\xi^2}{\sigma^2} + \frac{\eta^2}{\sigma^2} \right) \Gamma + \frac{\eta}{\sigma} \left(2 \frac{\xi}{\sigma} - 1 \right) \Lambda - \frac{\eta}{\sigma} \quad (31)$$

and

$$\frac{2\pi}{\gamma_p} w_\eta = \left(\frac{\xi^2}{\sigma^2} - \frac{\eta^2}{\sigma^2} - \frac{\xi}{\sigma} \right) \Lambda + \frac{\eta}{\sigma} \left(2 \frac{\xi}{\sigma} - 1 \right) \Gamma - \frac{\xi}{\sigma} + \frac{1}{2} \quad (32)$$

The superposition of the vorticities in equations (24) and (27) results in a linear vorticity distribution with $\gamma(\xi = 0) = \gamma_i$ and $\gamma(\xi = \sigma) = \gamma_j$. Its induced velocity is given by equations (25), (26), (28), and (29). For the parabolic vorticity distribution, equation (30), the factor γ_p cannot be derived from the vorticities γ_i and γ_j on the panel under consideration but, instead, depends on the vorticities on both the preceding and succeeding panels.

If a panel is not straight, it is divided into a number of straight subpanels n_p for the numerical computation of its induced velocity at the point (ξ, η) . The number of subpanels n_p is determined by the distances ξ/σ and η/σ . If these distances are great enough, the camber of the panel is neglected. Furthermore, if these distances are even greater, a global development is introduced in which each panel is replaced by a single vortex.

Special higher-order developments are used for the velocity induced at points on the panel itself. At these points, equations (25), (26), (28), (29), (31), and (32) fail. For the vorticity in equation (24), the induced tangential velocity at the beginning of a panel (fig. 10) is

$$\frac{2\pi}{\gamma_i} w_t = \frac{5}{6} g_1 + \frac{1}{3} g_2 + \frac{\pi}{2} \quad (33)$$

where g_1 and g_2 are the slopes shown in figure 10. For the vorticity in equation (27), the induced tangential velocity is

$$\frac{2\pi}{\gamma_j} w_t = \frac{2}{3} g_1 - \frac{1}{6} g_2 \quad (34)$$

and, for the vorticity in equation (30), it is

$$\frac{2\pi}{\gamma_p} w_t = \frac{5}{12} g_1 - \frac{1}{4} g_2 \quad (35)$$

All these equations together allow the computation of the influence of any vorticity distribution on a panel as specified by equations (24), (27), and (30) at any point P . A flow condition at point P results in a linear equation for the unknown γ . Thus, a system of n_q linear equations results, where n_q is the number of points defining the airfoil.

It should be noted that equations (24) to (32) require only very simple modification in order to compute the influence of a source distribution. The induced-velocity vector \vec{w} is rotated 90° in the mathematically positive direction when replacing the vorticity distribution $\gamma(\xi)$ by the same source distribution $s(\xi) = \gamma(\xi)$.

Trailing-Edge Singularity

As in other panel methods, a singularity arises from the circulation Γ around the airfoil, which is unconstrained unless required to satisfy the Kutta condition. Two different shapes are involved.

The first shape is a sharp trailing edge having either a zero or nonzero angle. For this shape, the singularity is caused by a finite circulation around the airfoil which results in an infinite velocity at the sharp trailing edge. Thus, an additional equation is required while one equation already in the equation system can be omitted. This additional equation consists of an extrapolation of the vorticity distribution to the trailing edge and an averaging at the trailing edge itself. The Kutta condition is introduced by requiring equal velocity on both sides of the trailing edge and zero normal velocity relative to the bisector of the trailing-edge angle. This implies that a zero trailing-edge angle is introduced and the shapes of the two panels at the trailing edge are changed using a cubic spline (fig. 11). It should be noted that the bisector of the angle between these new shapes is equal to the bisector of the original nonzero trailing-edge angle.

The omission of one of the equations in the system can sometimes cause errors at the point for which the flow condition is governed by the omitted equation. In other words, the equation system is not exactly singular due to small numerical approximations. In the present method, none of the equations is omitted but rather the entire equation system is multiplied by the transposed coefficient matrix. Thus, all the equations are solved as accurately as possible in a least-squares sense. (See ref. 10, for example.)

The second shape is a blunt trailing edge - one in which the upper- and lower-surface trailing-edge points are not the same (fig. 12). For this shape, not only a vorticity distribution γ but also a source distribution s is introduced at the base of the airfoil. Both are linear distributions over the base length and are determined such that no flow singularities occur at either of the two trailing-edge points. A tangential-flow condition and a normal-flow condition are satisfied at the inner middle of the base. This procedure also results in one more equation than necessary. The equation system is solved in the same least-squares sense as for sharp trailing edges, even though blunt trailing edges are not singular with respect to circulation.

The trailing-edge procedure employed for analysis is determined automatically from the trailing-edge coordinates. Only the airfoil coordinates are required, no other specification is necessary.

Spline Routine

The panel method requires a spline routine. Most software systems contain such routines, but nearly all of these routines do not spline in the x-y plane. Instead, those routines spline $x(s)$ and $y(s)$ separately, where s is the length coordinate along the polygon connecting the points. These routines do not satisfy the condition requiring a minimal integral over the curvature in the x-y plane. Accordingly, a new spline routine was developed which better satisfies this condition. It does contain, however, a certain linearization of the $\xi-\eta$ coordinates shown in figure 8. If the slopes of the spline fit are too large for a given panel, the spline routine might not give results which are precise enough. This could, of course, lead to inaccurate results from the panel method. So, a warning is printed in subroutine SMOOTH if the absolute values of the slopes are greater than 0.4.

BOUNDARY-LAYER METHOD

General Boundary-Layer Equations

An integral method is used for the analysis of the boundary layer (ref. 8). If $u(x,y)$ is the tangential velocity component within the boundary layer, x the length along the surface in the stream direction, and y the length normal to the surface, then the potential-flow velocity is

$$U(x) = \lim_{y \rightarrow \infty} u(x,y) \quad (36)$$

the displacement thickness is

$$\delta_1(x) = \int_0^{\infty} \left(1 - \frac{u}{U}\right) dy \quad (37)$$

the momentum thickness is

$$\delta_2(x) = \int_0^{\infty} \left(1 - \frac{u}{U}\right) \frac{u}{U} dy \quad (38)$$

and the energy thickness is

$$\delta_3(x) = \int_0^{\infty} \left[1 - \left(\frac{u}{U} \right)^2 \right] \frac{u}{U} dy \quad (39)$$

The shape factors H_{12} and H_{32} then are

$$H_{12} = \frac{\delta_1}{\delta_2} \quad (40)$$

and

$$H_{32} = \frac{\delta_3}{\delta_2} \quad (41)$$

If v_0 is the normal velocity at the surface (where blowing is positive and suction is negative), the integral momentum equation is

$$\delta_2' + (2 + H_{12}) \frac{U'}{U} \delta_2 = C_f + \frac{v_0}{U} \quad (42)$$

and the integral energy equation is

$$\delta_3' + 3 \frac{U'}{U} \delta_3 = C_D + \frac{v_0}{U} \quad (43)$$

where the prime denotes differentiation with respect to x , the skin-friction coefficient is

$$C_f = \frac{\tau_0}{\rho U^2} \quad (44)$$

and the dissipation coefficient is

$$C_D = \frac{2}{\rho U^3} \int_0^{\infty} \tau \frac{\partial u}{\partial y} dy \quad (45)$$

Equations (42) and (43) are satisfied by every solution $u(x,y)$ of the full partial differential equation for the boundary-layer flow. Approximate solutions can be determined by only allowing velocity distributions of the

form $\frac{u}{U} = f\left(\frac{y}{\delta(x)}, H(x)\right)$, where δ is a thickness factor and H a shape factor. The present method uses as the thickness factor δ_2 and as the shape factor H_{32} .

Laminar Boundary Layers

For laminar boundary layers, some Hartree profiles (ref. 3) have been selected as velocity distributions. If λ is the reference length and U_∞ the reference velocity, this leads to

$$C_f = \frac{\epsilon^*}{R \frac{U}{U_\infty} \frac{\delta_2}{\lambda}} \quad (46)$$

$$C_D = \frac{2D^*}{R \frac{U}{U_\infty} \frac{\delta_2}{\lambda}} \quad (47)$$

$$H_{12} = 4.02922 - (583.60182 - 724.55916H_{32} + 227.18220H_{32}^2) \sqrt{H_{32} - 1.51509}$$

if $1.51509 \leq H_{32} < 1.57258$ (48)

$$H_{12} = 79.870845 - 89.582142H_{32} + 25.715786H_{32}^2$$

if $H_{32} \geq 1.57258$

$$\epsilon^* = 2.512589 - 1.686095H_{12} + 0.391541H_{12}^2 - 0.031729H_{12}^3$$

if $1.51509 \leq H_{32} < 1.57258$ (49)

$$\epsilon^* = 1.372391 - 4.226253H_{32} + 2.221687H_{32}^2$$

if $H_{32} \geq 1.57258$

and

$$D^* = 7.853976 - 10.260551H_{32} + 3.418898H_{32}^2 \quad (50)$$

(See ref. 8.) The Reynolds number is

$$R = \frac{U_{\infty} l}{\nu} \quad (51)$$

The local Reynolds number based on momentum thickness is

$$R_{\delta_2} = \frac{U \delta_2}{\nu} = R \frac{U}{U_{\infty}} \frac{\delta_2}{l} \quad (52)$$

For airfoil boundary layers, the reference length l is the airfoil chord c and the reference velocity U_{∞} is the free-stream velocity.

Equations (46) to (50) specify δ_2' and δ_3' as functions of δ_2 and δ_3 , provided the potential-flow velocity $U(x)$, the suction velocity $v_0(x)$, and the Reynolds number R are given. Because δ_2 and δ_3 are involved, it is logical to use H_{32} as the shape factor. This is not a major deviation from other methods which use H_{12} instead of H_{32} , as H_{12} is a unique function of H_{32} . (See eqs. (48).) Thus, the shape of the boundary-layer velocity distribution $u(y)$ can be determined from H_{32} as well as from H_{12} . The Blasius boundary layer has a shape factor $H_{32} = 1.57258$. If $H_{32} \geq 1.57258$, $u(y)$ has no inflection point. If $H_{32} < 1.57258$, $u(y)$ has an inflection point.

$H_{32} = 1.51509$ is the laminar-separation limit where $\frac{\partial u}{\partial y} = 0$ at $y = 0$. Favorable pressure gradients and $v_0 < 0$ tend to increase H_{32} and vice versa.

Initial values for δ_2 and δ_3 can be derived from closed solutions for the first step of length Δx . If the boundary layer begins at $x = 0$ with $U = 0$ (i.e., the potential flow has a stagnation point at $x = 0$), the solution for the first step is

$$\left. \frac{\delta_2(\Delta x)}{l} = 0.29004 \sqrt{\frac{U_{\infty} \Delta x}{RU(\Delta x)l}} \right\} \quad (53)$$

and

$$\delta_3 = 1.61998 \delta_2$$

For a sharp edge with $U = U_0$ at $x = 0$, the first step can be determined from the Blasius solution, which results in

$$\left. \begin{aligned} \frac{\delta_2(\Delta x)}{l} &= 0.66411 \sqrt{\frac{U_\infty \Delta x}{RU_0 l}} \\ \text{and} \\ \delta_3 &= 1.57258\delta_2 \end{aligned} \right\} \quad (54)$$

(See ref. 8.)

The numerical integration of equations (42) and (43) results in δ_2 and δ_3 and, hence, H_{32} at every point x . Laminar separation is predicted when $H_{32} = 1.51509$. Constant potential-flow velocity yields the constant value $H_{32} = 1.57258$. The approximate solutions obtained from this method usually agree very well with exact solutions (ref. 8).

Turbulent Boundary Layers

The turbulent boundary-layer method does not allow the derivation of the expressions for H_{12} , C_f , and C_D in the same simple manner as did the laminar method which used certain selected velocity distributions. Instead, the empirical expressions of Wieghardt, Ludwig-Tillman, and Rotta are employed in a slightly modified form as derived in reference 8. These expressions are

$$H_{12} = \frac{11H_{32} + 15}{48H_{32} - 59} \quad (55)$$

$$C_f = 0.045716 \left[(H_{12} - 1) \frac{U\delta_2}{v} \right]^{-0.232} e^{-1.260H_{12}} \quad (56)$$

$$C_D = 0.0100 \left[(H_{12} - 1) \frac{U\delta_2}{v} \right]^{-1/6} \quad (57)$$

While, for laminar boundary layers, a single value of H_{32} indicates separation, this is not true for turbulent boundary layers as determined by equations (55) to (57). It can only be stated that for $H_{32} > 1.58$, there will surely be no separation and that for $H_{32} < 1.46$, there will surely be separation. It has been determined, however, that thicker boundary layers tend to separate at lower H_{32} values. Because the present method yields lower values of H_{32} for adverse pressure gradients than other methods, turbulent separation is assumed when $H_{32} = 1.46$.

Transition

The prediction of boundary-layer transition requires a switch from the laminar laws (eqs. (46) to (50)) to the turbulent ones (eqs. (55) to (57)). The

transition criterion by which this switch is determined assumes that transition has occurred if

$$\ln R_{\delta_2} \geq 18.4H_{32} - 21.74 - 0.36r \quad (58)$$

where r is a roughness factor. This criterion, as shown in figure 13 for $r = 0$, predicts that transition occurs later if H_{32} is higher (i.e., the pressure gradient is more favorable). The constants in inequality (58) were empirically derived (ref. 11). A roughness factor of $r = 0$ corresponds to natural transition on a smooth surface with no free-stream turbulence. Increasing r results in earlier transition. A roughness factor of $r = 4$ is representative of the disturbances caused by insects or a rough surface or by a turbulent free stream. It is recommended that the results obtained for different roughness factors be checked when comparing the method to data obtained in a turbulent free stream (ref. 12). Roughness factors of 6 or below can be used. A more detailed description of the transition criterion is given in several references (for example, ref. 12).

Laminar Separation Bubbles

Frequently, laminar separation ($H_{32} = 1.51509$) occurs before the transition criterion, inequality (58), is satisfied. Upon the prediction of laminar separation, the program continues the computation by switching to the turbulent laws. In the real flow, laminar separation is often followed by turbulent reattachment. This phenomenon is known as a laminar separation bubble. The results of the present method exhibit a certain analogy to this bubble. If the local Reynolds number R_{δ_2} is small enough, the turbulent shear stress and dissipation laws yield little more, and sometimes less, skin friction and dissipation than the laminar laws. In this situation, the method indicates either a very slowly increasing shape factor H_{32} or a decreasing H_{32} which rapidly reaches the turbulent separation value of $H_{32} = 1.46$.

For a slowly increasing H_{32} , a laminar separation bubble which extends at least to the point where $H_{32} = 1.58$ must be expected. This possibility occurs not only following laminar separation, but also in a very steep adverse pressure gradient following transition. If the method shows immediate turbulent separation following transition, this indicates that laminar separation without turbulent reattachment must be expected in the real flow.

If a bubble is predicted, there is only one way to alleviate it. The airfoil must be modified such that the pressure gradient at the beginning of the turbulent boundary layer is reduced. This is, of course, much easier in the design mode than in the analysis mode.

In many multipoint designs, a bubble exists at many angles of attack, and an improvement at any one angle of attack may cause difficulties at another. For such multipoint designs, it is important to consider the variation of Reynolds number with C_L which occurs on an aircraft in flight.

Boundary-Layer Development

It is very informative, particularly with respect to laminar separation and laminar separation bubbles, to plot the boundary-layer development $R\delta_2$ versus H_{32} as shown in figure 13. This figure shows the boundary-layer development along the upper surface of airfoil 1098 at various angles of attack for $R = 1 \times 10^6$. For segments of the airfoil having constant velocity ($\alpha = \alpha^*$) (fig. 6), the boundary layer approaches the Blasius solution having a constant shape factor H_{32} of 1.573 and increasing momentum thickness δ_2 . These segments correspond to the vertical lines in figure 13. As the angle of attack is increased ($\alpha > \alpha^*$), the velocity distributions become concave over the forward portion of the airfoil or, in other words, the airfoil "pulls a peak" at the leading edge. These concave distributions are similar to those which produce Hartree boundary layers, but the curves in figure 13 show increasing H_{32} with increasing $R\delta_2$ for $\alpha > \alpha^*$, whereas the Hartree boundary layers result in lower but still constant shape factors. This means that these velocity distributions are more concave than the Hartree (power law) distributions. Thus, as the angle of attack is increased even more, laminar separation will occur at the leading edge. In figure 13, the curve for $\alpha = 14^\circ$ shows that the boundary-layer development along this leading-edge velocity peak is tending toward the laminar-separation limit ($H_{32} = 1.51509$). This tendency toward laminar separation can be eliminated by the introduction of segments having higher α^* values toward the leading edge. Experience indicates that the curves in figure 13 can be shifted to the right by a positive $\Delta\alpha^*$ about as far as they would be shifted to the left by a positive $\Delta\alpha$ of the same amount. Obviously, it is much easier to control the development of the shape factor by manipulating α^* values than by altering a given velocity distribution at only one angle of attack. Also, it should be noted that a variation in Reynolds number only shifts the curves in figure 13 vertically. Thus, no additional problems arise with varying Reynolds number.

The program contains an option whereby diagrams such as that in figure 13 can be plotted automatically if a boundary-layer analysis is performed. Through the use of this diagram, it is very easy to control the laminar boundary-layer development for a multipoint design as demonstrated in references 9 and 11.

Section Characteristics

For any given airfoil, it is easy to obtain the potential-flow velocity distributions at arbitrary angles of attack from the design or the analysis methods. The boundary-layer method can then be applied to these velocity distributions. The displacement thickness obtained from the boundary-layer development could be added to the airfoil contour and further analysis of this "fluid airfoil" using the panel method would result in a new velocity distribution. This type of displacement iteration, however, has not yet been incorporated into the program.

The boundary-layer characteristics at the trailing edge allow the Squire-Young formula (ref. 13) to be applied to obtain the drag

$$c_d = 2\delta_{2,te} \left(\frac{U_{te}}{U_\infty} \right)^{\frac{5+H_{12,te}}{2}} \quad (59)$$

where te refers to conditions at the trailing edge. This formula, however, was not developed for use at high values of H_{12} like those which result when the flow at the trailing edge is near separation. For these conditions, the predicted values of c_d are too high. Therefore, the formula was empirically modified to

$$c_d = 2\delta_{2,te} \left(\frac{U_{te}}{U_\infty} \right)^{\frac{5+H_{12,te}^*}{2}} \quad (60)$$

where

$$H_{12,te}^* = H_{12,te} \quad \text{for } H_{12,te} \leq 2.5$$

$$= 2.5 \quad \text{for } H_{12,te} > 2.5$$

This formula is applied at the trailing edge of the upper and lower surfaces separately. If boundary-layer separation is predicted at a velocity U_{sep} , the program does not perform any more boundary-layer computations. Instead, it corrects only the momentum thickness δ_2 by

$$\delta_{2,te} = \delta_{2,sep} \left(\frac{U_{sep}}{U_{te}} \right)^{\frac{5+H_{12,sep}}{2}}$$

Because this transfer is performed with $H_{12,sep} = 2.803$, which follows from equation (55) for $H_{32} = 1.46$, the drag as computed from equation (60) is

$$c_d = c_{d,sep} \left(\frac{U_{sep}}{U_{te}} \right)^{0.15}$$

Thus, a certain drag penalty due to separation is included although it is small and not empirically founded.

The lift and pitching-moment coefficients are determined from the potential flow. Some simple viscous corrections are then applied to these coefficients.

The lift-curve slope where no separation is present is reduced to 2π from its theoretical value. In other words, the potential-flow thickness effects are assumed to be offset by the boundary-layer displacement effects. If α_0 is the zero-lift angle as shown in figure 14, the lift coefficient with no separation is

$$c_{l,ns} = 2\pi(\alpha_c + \alpha_0) \quad (61)$$

Note that α_c is the angle of attack relative to the chord line and $\alpha_c + \alpha_0$ is the angle of attack relative to the zero-lift line α . In this situation, all angles are in radians. If a separation of length s_{sep} is predicted on the upper surface, an angle-of-attack correction

$$\Delta\alpha \approx -\frac{1}{2} \frac{s_{sep}}{c} (\delta_{us} + \alpha_c)$$

is used to determine a lift-coefficient correction

$$\Delta c_l = 2\pi \Delta\alpha = -\pi \frac{s_{sep}}{c} (\delta_{us} + \alpha_c) \quad (62)$$

where δ_{us} is the slope of the airfoil upper surface near the trailing edge, as shown in figure 14. A similar effect occurs in potential-flow wake theory (Helmholtz flow). (See ref. 14.) The corresponding correction is applied if a separation is predicted on the lower surface. Neither a positive Δc_l for the upper surface nor a negative Δc_l for the lower surface is allowed. These possibilities could occur for airfoils having a flap deflection.

The present method has been applied successfully at Reynolds numbers from 2×10^4 to 1×10^8 . See reference 9, for example.

IMPLEMENTATION OF THE THEORY

The program contains three major blocks corresponding to the three theoretical methods to be employed, as well as several other subroutines for special purposes. See "Flow Chart and Input Card Summary."

The program is available at a nominal fee through the following organization:

Computer Software Management Information Center (COSMIC)
112 Barrow Hall, University of Georgia
Athens, Georgia 30602

Request the program by the designation PROFILE LAR-12727.

AIRFOIL DESIGN SUBROUTINE

The entire potential-flow airfoil design method is contained in subroutine TRAPRO which calculates the airfoil coordinates x/c and y/c , the inner normal angles β , and the function $v(\alpha, \phi) / \cos\left(\frac{\phi}{2} - \alpha\right)$ which is required for the computation of the velocity $v(\alpha, \phi)$ at any point at any angle of attack. These results are stored in the blank COMMON arrays X, Y, ARG, and VF, respectively. Subroutine TRAPRO calls subroutine DRAW and some of the auxiliary subroutines discussed in "Utility Subroutines."

AIRFOIL ANALYSIS SUBROUTINES

The center of this portion of the program is subroutine PANEL which requires as input the airfoil coordinates x/c and y/c , which are stored in blank COMMON, and the spline-fit inner normal angles β , which are stored in array XP in blank COMMON. The panel method is applied at 0° and 90° angle of attack. The resulting velocity (vorticity) distributions are stored in array GAMMA which is a formal parameter in the subroutine. Simultaneously, the results are stored in arrays VF and ARG in blank COMMON. The lift coefficients for the two angles of attack are stored in array CAE which is also a formal parameter. The lift coefficient for 90° angle of attack is identical to the lift-curve slope. The zero-lift angle is stored as ALN in COMMON/PRAL/. Subroutine PANEL calls subroutine GAUSS.

BOUNDARY-LAYER SUBROUTINES

The heart of this portion of the program is subroutine GRS, which computes one integration step using equations (42) and (43). This subroutine can also compute the effect of suction on the boundary layer, although it cannot determine the required surface porosities. Thus, the suction options have greater significance if this subroutine is used independently. If this is done, the fixed transition positions must be specified in COMMON/TRIT/ when using transition modes 1 and 2 as discussed later in this section. Subroutine GRS calls subroutines GRUP, H12B, and UMP.

Subroutine GRUP provides the right-hand sides of equations (42) and (43) by calling subroutine CDCF to obtain C_D and C_f . Subroutine CDCF calls subroutine H12B to get H_{12} from H_{32} . The interconnection of these subroutines is shown in figure 15. Of course, it is possible to substitute other expressions for C_D , C_f , and H_{12} by replacing subroutines GRUP and H12B with other routines.

Subroutine GRS has as input parameters:

HVGL = H_{32} at the beginning of the step

D2 = δ_2 at the beginning of the step

UK = U at the beginning of the step

UK1 = U at the end of the step

DL = the length of the step divided by the chord

WRE = \sqrt{R}

MU = the transition mode

MA = the suction mode

The values of UK and DL are changed during the execution of subroutine GRS.
The transition modes are:

MU = 0 - transition at laminar separation

MU = 1 or 2 - fixed transition where the transition positions are contained in array XTRI in COMMON/TRIT/

MU = 3 - natural transition (See inequality (58).)

MU > 3 - transition with roughness factor $r = MU - 3$
(See inequality (58).)

The suction modes are:

MA = 0 - no suction

MA = 1, 2, or 3 - The suction is computed outside of subroutine GRS. In these situations, V and V1 are also input parameters where V is the suction velocity at the beginning of the step and V1 is the suction velocity at the end of the step.

MA = 4, 5, 6, or 7 - The suction is computed within subroutine GRS such that

$$H_{32} = a + b \ln R\delta_2 \quad (63)$$

(See ref. 8.) The values of a and b for the different suction modes are obtained from arrays AA and BB in COMMON/GRZK/.

MA = 8 - The suction velocity is

$$V = 25(1.98 - H_{32})\delta_2 \frac{dH_{12}}{dx} \quad (64)$$

according to a suggestion made by August Raszpet.¹

¹Head, Department of Aerophysics, Mississippi State University.

In all situations where $MA \geq 4$, the values of V and $V1$ are determined in subroutine GRS. Only $MA = 0$ is permitted when using subroutine GRS in the present version of the program.

The output parameters (besides V and $V1$) of subroutine GRS are:

HR = H_{32} at the end of the step

D2R = δ_2 at the end of the step

XA = the length within the step which is not separated

XU = the length within the step which is laminar

DCQ = the contribution of the step to the suction coefficient
(See ref. 8.)

Subroutine GRS also handles numerical instabilities by automatically reducing the step size thereby computing one step of length DL as many shorter "substeps." (See ref. 8.)

If $U = 0$ is specified in the input parameters, the stagnation point solution (eqs. (53)) is used. If $D2 = 0$ is specified, the Blasius solution (eqs. (54)) is used.

Subroutine GRP computes the boundary-layer development for any given velocity distribution and Reynolds number. It calls subroutine GRS for each step where a step is the length which extends from one airfoil coordinate to the next, except for the stagnation point which can occur between two coordinates. Thus, the number of airfoil coordinates determines the number of boundary-layer steps to be computed. Up to five Reynolds numbers and their associated transition modes can be handled in parallel. Subroutine GRS always stores the results of a step for use in the next step, if possible. This procedure saves much computing time if one velocity distribution is analyzed at several Reynolds numbers in parallel. Subroutine GRP is called from several places in the main routine. It can handle, for fixed Reynolds numbers, velocity distributions for many angles of attack with only one call, or it can be restricted to only one angle of attack if the Reynolds number is to depend on α .

The velocity distributions are computed in function $VPR(N,I)$. The value of the variable ITP in $COMMON/PRAL/$ indicates whether the velocity function was determined in the design or the analysis mode and, therefore, determines the velocity formula to be selected (automatically). The value of the variable N is the index of the airfoil coordinate at which the velocity is to be computed and the angle of attack is obtained from the I th value in array ALV in $COMMON/PRAL/$. The boundary-layer subroutines are, of course, not concerned with the source of the velocity function.

Subroutine GRP has as input parameters:

NAX = the number of angles of attack; if $NAX < 0$, $|NAX|$ indicates which single angle of attack to use

RE(JR) = the array of Reynolds numbers

MU(JR) = the array of transition modes

JR = the total number of Reynolds numbers for each α

ISTIPT = the pen number to be used for the boundary-layer plot

The results from subroutine GRP include:

- (1) The boundary-layer developments, which can be printed and plotted
- (2) The drag coefficient and the lengths of turbulent flow s_{turb} and separated flow s_{sep} for each surface at all angles of attack α and Reynolds numbers R ; these values are stored in the blank COMMON arrays CW, SA, and SU, respectively
- (3) A boundary-layer summary containing c_d , s_{turb} , and s_{sep} for each surface and the total c_d , c_l , and c_m for the airfoil; these values contain the viscous corrections computed in subroutine VISC (The boundary-layer summary can be printed and plotted.)

ADDITIONAL RELATED SUBROUTINES

Subroutine PADD inserts additional points along the airfoil surface using a spline.

Subroutine MOMENT computes the potential-flow pitching-moment coefficient C_m about the quarter-chord point and the section hinge-moment coefficient C_h about the flap-hinge point.

Subroutine FLAP generates a new set of airfoil coordinates which incorporate the deflection of a simple flap.

Subroutine FIXLES reads the coordinates of a given airfoil.

Subroutine CDCL plots the boundary-layer summary.

Subroutine DIA computes airfoil thickness.

Subroutine DIAGR plots the airfoil and its velocity distributions or the pressure envelope.

Logical functions YG4 and YG5 exclude portions of the plot generated by subroutine DIAGR to prevent certain confusing line intersections.

Subroutine STRAAK plots airfoils having various chord lengths.

Subroutine STRDR prints the coordinates in cm of the airfoils plotted by subroutine STRAAK.

UTILITY SUBROUTINES

Subroutine WANDEL(N,NF,M,MHB) translates an integer number N into alpha-numeric code. The result is contained in the array NF(M) with no leading zeros and MHB blank spaces following the number. The array NF can be written in format MA1.

Subroutine GAUSS(A,B,N,M,EPS) solves a system of N linear equations having M terms in the right-hand sides. The coefficient matrix is A(N,N) and the terms in the right-hand sides are contained in the matrix B(N,M). EPS is a lower limit for the determinant of A(N,N). If the determinant is less than EPS, a warning is printed in subroutine GAUSS. The solution replaces the input B(N,M).

Function RUND is a round-off procedure.

Functions SING and COSG calculate the sine and cosine of angles in degrees.

Subroutine QIP(X,Y,A) performs a quadratic interpolation which yields the three coefficients of a parabola through three points.

Subroutine SPLITZ(X,Y,NQ,BETA) performs a spline fit of the NQ points contained in arrays X(NQ) and Y(NQ). The result is the array of angles β normal to the spline curve at every point. This means that the tangent oriented in the sense of increasing point number is rotated 90° in the mathematically positive direction. Thus, β is the inner normal angle of an airfoil for which the points are numbered in sequence from the trailing edge forward along the upper surface to the leading edge with decreasing x and then along the lower surface to the trailing edge with increasing x. This sequence is used in both the design and the panel methods. It yields $-\pi < \beta < \pi$ with no ambiguity.

Subroutine CINT(P,Z,NQ,IZZ) numerically integrates P over NQ ordinates and stores the result in array Z. All points are assumed to be equidistant, the distance being 1. If IZZ = 0, the trapezoidal formula is used. If IZZ \neq 0, a third-order formula is used which assumes that the P values are periodic with NQ - 1. The initial value of Z(1) is 0.

INPUT AND OUTPUT (USER'S GUIDE)

PRINCIPLES OF INPUT

The sequence of execution within the program is controlled in a very flexible manner by the input cards themselves. Each input card has a name in the first four columns. Thus, an ALFA card has "ALFA" in the first four columns and an RE card has "RE" plus two blanks. These blanks are part of the name and, therefore, those two columns must not be used for other purposes.

The program reads this name and then branches to the appropriate portion of the program. After performing the computations, printing, or plotting initiated by the input card, the program returns to the starting point in the main routine and reads the next input card. In effect, the program is asking "What should I do next?"

Some of the cards initiate computation, printing, or plotting which relies on results generated by other cards. Thus, the sequence of cards is not completely arbitrary. See "Flow Chart and Input-Card Summary."

All of the data from a card remain unchanged until new data are read from a card of the same type. Thus, it is possible to leave some data words blank for many cards if the corresponding data words from a previous card are not to be changed. Exceptions to this rule will be described in detail.

Every job must be terminated by an ENDE card. This card leads to the STOP statement.

FORMAT OF DATA CARDS

All of the data cards except those containing airfoil coordinates are read in the same format - A4,3I1,I3,14F5.2. This means that each card contains in:

Columns 1-4, an alphanumeric card name

Columns 5-7, three integer numbers NUPA, NUPE, and NUPI having one digit each

Columns 8-10, one integer number NUPU having three digits

Columns 11-80, fourteen floating point numbers each of which is in format F5.2; these numbers are referred to as F-words $F_1 - F_{14}$ in the following description

First, a review of the use of format F5.2. Each number is read, using this format, from a fixed group of five columns on the card; all five columns are read and blanks are read as zeros. If no decimal point is used, the digits in the two right-hand columns are set behind the decimal point, the others ahead. For example,

		-	0	5
--	--	---	---	---

 is read as -0.05

+	7	5		
---	---	---	--	--

 is read as 75.00

If a decimal point is used, the second number of the format is neglected. For example,

.	0	0	3	7
---	---	---	---	---

 is read as 0.0037

-	3	7	9	.
---	---	---	---	---

 is read as -379.0

Thus, it is possible to read input numbers written with a decimal point which would not fit into format F5.2 without a decimal point.

For some of the input cards, the F-words are multiplied by a factor before they are used within the program. In the following description for example,

$$0.1 F_1 = K [F5.3]$$

might appear. When no decimal point is used in the F-word, this is equivalent to reading K in format F5.3. Thus, $K = 0.078$ can be punched

			7	8
--	--	--	---	---

 or

.	7	8		
---	---	---	--	--

, but not

.	0	7	8	
---	---	---	---	--

 which would result in $K = 0.0078$. So, if an effective format appears in brackets as shown above, it is valid only where a decimal point is not used.

OVERVIEW OF INPUT CARDS

The various input cards can be divided into the following four groups: input for airfoil design, input for airfoil analysis, options for both design and analysis modes, and options for boundary-layer analysis. The function of each input card is shown in "Flow Chart and Input-Card Summary."

The input for airfoil design is contained on three card types. The TRAL card contains the arc limits and their corresponding α^* values. The TRA2 card contains the pressure-recovery and closure-contribution parameters. The ABSZ card allows the number of points on the circle to be multiplied by a specified factor.

The input for airfoil analysis is also contained on three card types. The FXPR card reads a set of airfoil coordinates. The PAN card switches the program from the design mode to the analysis mode. The FLAP card alters the airfoil shape to that corresponding to the deflection of a simple flap.

The options for both design and analysis modes are contained on five card types. The ALFA card contains the angles of attack to be analyzed. The PUXY card punches the airfoil coordinates onto cards. The DIAG card plots velocity distributions. The STRD and STRK cards together plot airfoils having various chord lengths.

The options for boundary-layer analysis are contained on five card types. The RE card contains the transition modes and Reynolds numbers. The FLZW card contains the aircraft data necessary to compute the boundary-layer development

where the Reynolds number is varied with aircraft lift coefficient and local chord. The PLW card contains the additional aircraft data necessary to generate a polar. The PLWA card allows the aircraft data contained on the PLW card to be modified in specified increments. The CDCL card plots the boundary-layer summary, including section characteristics.

The ENDE card terminates the job.

INPUT FOR AIRFOIL DESIGN

As described in "Potential-Flow Airfoil Design Method," an airfoil design is specified by several arcs with limits ϕ_i having α^* values of α_i^* and the parameters for the recovery functions and the closure contributions which include ϕ_w , $\bar{\phi}_w$, ϕ_s , and $\bar{\phi}_s$. The arc limits as well as ϕ_w , $\bar{\phi}_w$, ϕ_s , and $\bar{\phi}_s$ are not specified in degrees but, instead, relative to point numbers. The airfoil points computed are images of equidistant points on the ζ circle. The number of points n_c on the circle can be specified but must be divisible by 4. Thus,

$$\Delta\phi = 2\pi/n_c = 360^\circ/n_c$$

and

$$v_i = \phi_i/\Delta\phi$$

$$\lambda = \phi_w/\Delta\phi$$

$$\bar{\lambda} = (2\pi - \bar{\phi}_w)/\Delta\phi$$

$$\lambda^* = \phi_s/\Delta\phi$$

$$\bar{\lambda}^* = (2\pi - \bar{\phi}_s)/\Delta\phi$$

A practical number of points n_c is 60. This means that $\Delta\phi = 360^\circ/60 = 6^\circ$. In this situation, $v_i = 15$ specifies an arc limit which corresponds to a point around the midchord of the upper surface of the airfoil, whereas $v_i = 30$ is near the leading edge and $v_i = 45$ is near the midchord of the lower surface. If v_i is an integer number, the arc limit specified by v_i corresponds to a point on the airfoil. If v_i is a decimal number, the arc limit occurs between two points on the airfoil. The same is true of the λ values which specify the beginnings of the pressure-recovery and closure-contribution regions. Of course, symmetric airfoils have $\bar{\lambda} = \lambda$ and $\bar{\lambda}^* = \lambda^*$.

The arc limit $v_{i,le}$ must not be specified. One arc limit, however, must be reserved for $v_{i,le}$. This is achieved by setting $v_{i,le} = 0$ in the input. The program then computes the correct value of $v_{i,le}$.

The input data for the design method is contained on three card types. The TRAL card specifies the arc limits v_i and their corresponding α_i^* values. The TRA2 card specifies the pressure-recovery and closure-contribution parameters used in equations (20) and (21), the iteration mode, and the amount of pressure recovery to be achieved by the closure contribution $K_H + \bar{K}_H$. The ABSZ card offers some additional options.

TRAL Card

NUPA and NUPE are neglected.

NUPI and NUPU = airfoil identification number (4 digits)

$$F_1 = v_1 \quad (F5.2)$$

$$F_2 = \alpha_1^* \quad \text{in degrees relative to the zero-lift line (F5.2)}$$

$$(F_3, F_4) = (v_2, \alpha_2^*) \quad \text{and so on}$$

If more than seven arcs are to be specified, another TRAL card must be included, with columns 5-10 identical to those on the first TRAL card. On this second TRAL card, $F_1 = v_8$, $F_2 = \alpha_8^*$, and so on. Up to four TRAL cards (28 arcs) are allowed for any one airfoil design. The number of points on the circle comes from the last arc limit $v_i = v_{I_a}$. This number must be divisible by 4.

The leading-edge arc limit must be specified by $v_{i,le} = 0$. The value of $v_{i,le}$ which is computed by the program must be between $v_{i,le-1}$ and $v_{i,le+1}$. Thus, if the value of $v_{i,le-1}$ is too large or the value of $v_{i,le+1}$ is too small, inequalities (18a) cannot be satisfied and no solution for $v_{i,le}$ can be found. In this situation, the program will print a message and then stop. Also, remember from inequality (18b) that $\alpha_{i,le}^* > \alpha_{i,le+1}^*$.

It is recommended that arc limits which do not correspond exactly to the points on the airfoil be specified. Arc limits which will fall between points (for example, $v_i = 16.5$) yield velocity distributions which are slightly smoothed. This was done on airfoil 1098 which is shown in figure 6. The TRAL card for this airfoil is:

```
TRAL 1098 3050 300 2750 1000 000 1200 5000 200
*****
```

which implies that $\Delta\phi = 6^\circ$ and, therefore,

$$v_1 = 23.5 \qquad \phi_1 = 141^\circ \qquad \alpha_1^* = 8^\circ$$

$$v_2 = 27.5 \qquad \phi_2 = 165^\circ \qquad \alpha_2^* = 10^\circ$$

$$v_3 = v_{i,le} = 0 \quad \phi_3 = \phi_{i,le} \quad \alpha_3^* = 12^\circ$$

$$v_4 = n_c = 60 \quad \phi_4 = 360^\circ \quad \alpha_4^* = 2^\circ < \alpha_3^*$$

TRA2 Card

NUPA, NUPE, NUPI, and NUPU are neglected. Columns 7-10 can, therefore, be used for the airfoil identification number, if desired.

$$F_1 = \lambda^* \quad (F5.2)$$

$$F_2 = \lambda \quad (F5.2)$$

F_3 = recovery specification mode RSM_{us} (F5.2) which determines the interpretation of F_4 and F_5 as follows:

RSM_{us}	$0.1F_4$	$0.1F_5$
0	K [F5.3]	μ [F5.3]
1	ω' [F5.3]	ω [F5.3]
2	μ [F5.3]	ω [F5.3]

$$F_6 = \bar{\lambda}^* \quad (F5.2)$$

$$F_7 = \bar{\lambda} \quad (F5.2)$$

F_8 = RSM_{1s} (F5.2) which determines the interpretation of F_9 and F_{10} as follows:

RSM_{1s}	$0.1F_9$	$0.1F_{10}$
0	\bar{K} [F5.3]	$\bar{\mu}$ [F5.3]
1	$\bar{\omega}'$ [F5.3]	$\bar{\omega}$ [F5.3]
2	$\bar{\mu}$ [F5.3]	$\bar{\omega}$ [F5.3]

F_{11} = ITMOD = the iteration mode to be used to achieve the desired value of K_S (F5.2)

$$0.1F_{12} = K_R = \text{the desired value of } K_S (= K_H + \bar{K}_H) \quad [F5.3]$$

$0.1F_{13} = K_{tol}$ = the tolerance on the achievement of K_R ; $K_{tol} = 0$, normally [F5.3]

F_{14} - Determines which integration scheme is to be used to compute the x/c and y/c airfoil coordinates (F5.2). If $F_{14} = 0$ (normal), a trapezoidal-rule integration is performed. If $F_{14} \neq 0$, a third-order integration is performed.

The following iteration modes can be used to achieve the desired trailing-edge angle as indicated by $K_H + \bar{K}_H$:

ITMOD = 0 - No iteration is performed; F_{12} and F_{13} are neglected and K_H and \bar{K}_H are determined by the other design parameters

ITMOD = 1 - All α_i^* for the upper surface ($i \leq i_{1e}$) are replaced by $\alpha_i^* + \Delta\alpha$

ITMOD = 2 - All α_i^* for the lower surface ($i > i_{1e}$) are replaced by $\alpha_i^* + \Delta\alpha$

ITMOD = 3 - All upper-surface α_i^* ($i \leq i_{1e}$) are replaced by $\alpha_i^* + \Delta\alpha$ and all lower-surface α_i^* ($i > i_{1e}$) are replaced by $\alpha_i^* - \Delta\alpha$

ITMOD = 4 - K is replaced by $K + \Delta K$

ITMOD = 5 - \bar{K} is replaced by $\bar{K} + \Delta K$

ITMOD = 6 - K is replaced by $K + \Delta K$ and \bar{K} is replaced by $\bar{K} + \Delta K$

ITMOD = 7 - $\alpha_{i,1e}^*$ is replaced by $\alpha_{i,1e}^* + \Delta\alpha$

ITMOD = 8 - $\alpha_{i,1e+1}^*$ is replaced by $\alpha_{i,1e+1}^* + \Delta\alpha$

ITMOD = 9 - $\alpha_{i,1e}^*$ is replaced by $\alpha_{i,1e}^* + \Delta\alpha$ and $\alpha_{i,1e+1}^*$ is replaced by $\alpha_{i,1e+1}^* - \Delta\alpha$

In all situations, $\Delta\alpha$ or ΔK is determined such that $K_H + \bar{K}_H = K_S = K_R$ which is specified by F_{12} . During the iteration, $\Delta\alpha$ is rounded off to two figures and K to three figures to the right of the decimal point. Therefore, K_S may not be exactly equal to K_R .

The TRA2 card initiates the design of the specified airfoil. The x/c and y/c airfoil coordinates and the inner normal angles are stored in blank COMMON arrays X, Y, and ARG, respectively, and the velocity function

$v/\cos\left(\frac{\phi}{2} - \alpha\right)$ in blank COMMON array VF.

Normally, the program prints all of the v_i values, including $v_{i,1e}$, and their corresponding α_i^* values as well as the pressure-recovery and closure-contribution parameters twice; once with the input values and once with the final values from the last iteration. Between these two tables, one line per iteration is printed which contains the values of K_S and $\Delta\alpha$ or ΔK for that iteration. Other print modes are controlled by the parameter mtr on the ABSZ card as discussed in the next section.

The TRA2 card for airfoil 1098 is shown below.

```
TRA2 1098 400 1450 200 1000 650 400 1450 200 1000 650 600 400 000 000
-----
```

The printout initiated by this TRA2 card with mtr = 1 follows:

TRANSCENDENTAL EQUATION RESULTS				AIRFOIL 1098		ITERATION 0		MODE 6	
NU	ALPHA*	OMEGA'	OMEGA	K	MU	K H	LAMBDA	LAMBDA*	
23.50	8.00	1.137	.650	.598	1.000	.608	14.50	4.00	
27.50	10.00								
32.01	12.00								
60.00	2.00	1.137	.650	.598	1.000	.091	14.50	4.00	
ITERATION 1		K S = -.311031		DELTA = -.07039044		ROUNDED = -.07			
ITERATION 2		K S = .390738		DELTA = -.00092388		ROUNDED = -.00			
ITERATION 3		K S = .400941		DELTA = .00009223		ROUNDED = 0.00			
TRANSCENDENTAL EQUATION RESULTS				AIRFOIL 1098		ITERATION 3		MODE 6	
NU	ALPHA*	OMEGA'	OMEGA	K	MU	K H	LAMBDA	LAMBDA*	
23.50	8.00	1.192	.639	.627	1.000	.459	14.50	4.00	
27.50	10.00								
32.01	12.00								
60.00	2.00	1.192	.639	.627	1.000	-.058	14.50	4.00	

ABSZ Card

Two options are controlled by this card. First, the print mode mtr for the design iteration can be changed and, second, the number of points NQ in the design subroutine can be changed.

If NQPA \neq 0, mtr = NUPE (I1):

mtr = 0 - No printout

mtr = 1 - Printout as described in the preceding section

mtr \geq 2 - Printout of every iteration

NUPI, NUPU, and F₁ are neglected.

If F₂ \neq 0, ABFA = F₂ (F5.2):

ABFA is a factor by which all v_i values on the TRA1 cards as well as λ , λ^* , $\bar{\lambda}$, and $\bar{\lambda}^*$ on the TRA2 card are multiplied. This factor allows the number of points NQ for a given airfoil design whose TRA1 and TRA2 cards have already been punched to be changed without having to repunch the TRA1 and TRA2 cards. Note that the

value of the last arc limit, $v_{I_a} \cdot ABFA$, must still be divisible by 4. Thus, the number of points $n_c = v_{I_a} = 60$ for a given airfoil can be changed to $n_c = 108$ by setting $ABFA = 1.8$. Remember that, if the arc limits on the TRAL cards were selected such that they would fall midway between the points on the airfoil, this will no longer be true if $ABFA \neq 1$.

The print mode mtr and factor ABFA remain in effect until the next ABSZ card is read. Thus, an ABSZ card with $F_2 = 1$ resets $ABFA = 1$. Initially, within the program, mtr is automatically set to 1 and ABFA to 1.

INPUT FOR AIRFOIL ANALYSIS

The potential-flow airfoil analysis method requires only a set of airfoil coordinates. Because these coordinates do not conveniently fit into the format used for all the other input cards, they are read by a separate subroutine FIXLES which is called if an FXPR card is used. This subroutine reads the coordinates into blank COMMON arrays X and Y in the correct sequence (i.e., from the trailing edge forward along the upper surface, around the leading edge, and back along the lower surface to the trailing edge). The output from the panel method is controlled by print mode mpa.

FXPR Card

If $NUPA \neq 0$, $mpa = NUPE$ (I1):

$mpa = 0$ - No printout

$mpa = 1$ or 2 - Only the headline containing the lift coefficients for 0° and 90° angle of attack and the zero-lift angle is printed

$mpa \geq 3$ - The headline plus the airfoil coordinates, the velocities for 0° and 90° angle of attack, and the slope angles are printed

If $NUPA = 9$, $mpa = NUPE$ but no panel-method analysis is performed. NUPI is neglected. The value of NUPU is transferred to subroutine FIXLES via variable ITP and allows various formats for the reading of the airfoil coordinates to be selected.

The print mode mpa is initially automatically set to 1 within the program.

The F-words specify additional points to be splined in between the original coordinates as read following an FXPR card. The five digits of F_i are denoted aabdd:

If $aa = 00$, one point is inserted at $x/c = 0.dd$ on the upper surface, if $b = 0$; on the lower surface, if $b = 1$.

If $aa \neq 00$, b points are inserted between points aa and $aa + 1$. The new points are spaced equidistantly along the straight line between the two original points, if $0.dd > x_{aa}/c$ and $0.dd > x_{aa+1}/c$; otherwise, the points are spaced such that, in the equation $x/c = 1/2 (1 + \cos \phi)$, the angles ϕ corresponding to the new points are equiangular.

The ability to insert additional points frequently allows the accuracy of the results to be improved if portions of the airfoil have sparse numbers of points. Near the trailing edge where the velocity is changing rapidly, it is recommended that the new points be spaced equiangularly on the circle. A value of $dd = 85$ has proven to be good for this purpose.

This capability must be used cautiously in one respect, however. The number of all the points in arrays X and Y behind the inserted points are increased by the number of inserted points. The simplest way to avoid difficulties is to specify the F-words with decreasing values of aa . This means that aa for F_1 should be greater than aa for F_2 which should be greater than aa for F_3 , and so on.

The following example illustrates the use of this capability. The NACA 6-series airfoils are defined by 26 points on each surface which results in a total of 51 points where the leading edge is point 26. (One of the two identical leading-edge points is omitted.) The points near the trailing edge ($x_1/c = 1.00$, $x_2/c \approx 0.95$, and $x_3/c \approx 0.90$), as well as those near the leading edge, are too widely spaced to obtain accurate results from the panel method. The FXPR card which follows

```
FXPR      502254912522125251250212501225
-----
```

inserts two points on each surface between $x/c = 0.95$ and 1.00 in the equiangular-spacing mode (words F_1 and F_6), one point on each surface between $x/c = 0.90$ and 0.95 in the equiangular-spacing mode (words F_2 and F_5), and one point between the leading edge and the first point on each surface (words F_3 and F_4) in the equidistant-spacing mode. Thus, a total of eight points is inserted.

Following the execution of subroutine PANEL, the airfoil coordinates are stored in blank COMMON arrays X and Y and the vorticities for $\alpha = 0^\circ$ and 90° , in arrays VF and ARG. Thus, all the data required by function VPR for computing the velocity at every point for every angle of attack are available.

Subroutine FIXLES handles the input of an airfoil to be analyzed in the following manner. First, the airfoil name is read in format I2A1 into array NAMP(12) in COMMON/PRAL/. Then, the airfoil coordinates x_i/c and y_i/c , where $i = 1, 2, \dots, n_q$, are read into blank COMMON arrays X(121) and Y(121). The sequence of the points begins at the trailing edge and extends forward

along the upper surface, around the leading edge, and then, back along the lower surface to the trailing edge. Thus, the trailing-edge point must appear twice, even if the trailing edge is sharp, whereas the leading-edge point should appear only once. The total number of points n_q is then written into blank COMMON variable NQ. The value of $n_q \leq 121$. Finally, the slope of the upper surface δ_{us} near the trailing edge is written into variable DLT in COMMON/PRAL/, the lower-surface slope δ_{ls} into DLTU. The definition of these slopes is such that, for symmetric airfoils, $\delta_{us} = \delta_{ls}$. Because most airfoils are not very smooth along the region which surrounds the trailing edge, it is recommended that some point $(x,y)_i$ on the upper surface and some point $(x,y)_j$ on the lower surface be selected such that $x_i/c \approx 0.9$ and $x_j/c \approx 0.9$. Thus,

$$\delta_{us} = y_i / (c - x_i)$$

and

$$\delta_{ls} = y_j / (c - x_j)$$

In the present version of the program, a simple form of subroutine FIXLES is included which reads the following cards:

- (1) A card with only the airfoil name in columns 1-12 (12A1)
- (2) A card with the number of upper-surface points MUP and the number of lower-surface points MLOW in columns 1-10 (2I5)
- (3) Four sets of cards containing the MUP x/c values, MUP y/c values, MLOW x/c values, and MLOW y/c values (8F10.5). Thus, all of these cards have eight values each, except the last card in each set, which could have less than eight values. The sequence of the points within each set is from the leading edge back to the trailing edge. The leading edge must appear in each set. One leading-edge point is automatically omitted during the execution of subroutine FIXLES. The total number of points NQ is therefore $MUP + MLOW - 1$

It is, of course, easy to adapt subroutine FIXLES to other card formats.

PAN Card

The PAN card switches the program from the design mode to the analysis mode and is, therefore, required if the panel method is to be employed following a design (TRA1 and TRA2 cards). It contains the same options regarding the print mode and the insertion of points as the FXPR card. No PAN card is required after an FXPR card.

NUPA and NUPE - See "FXPR Card"

F_i - See "FXPR Card"

FLAP Card

The shape of an airfoil analyzed by the panel method can be altered so as to correspond to the deflection of a simple flap. Because the panel method does not allow sharp corners in the airfoil surface, a transition arc between the flap and the forward portion of the airfoil must be introduced. Such a transition arc occurs "naturally" on the convex side of a real wing, but not on the concave side where a corner is formed. This concave corner, however, is smoothed by a locally separated region and, therefore, it is reasonable to introduce a transition arc there as well.

NUPA, NUPE, NUPI, and NUPU are neglected.

$0.01F_1$ = flap chord, c_f/c [F5.4]

$0.01F_2$ = y/c -position of the flap hinge [F5.4];
thus, the flap-hinge point is located at $x_h/c = 1.0 - F_1$
and $y_h/c = F_2$

$0.01F_3$ = one half of the transition arc length s_T/c [F5.4];
 s_T/c is typically 0.04 to 0.06, depending on the amount of
flap deflection

F_4 = flap deflection δ_f in degrees, positive downward (F5.2)

The panel method is called automatically after a FLAP card is read. A FLAP card, however, requires data from a preceding panel method analysis. Therefore, an airfoil generated in the design mode requires a PAN card before a FLAP card can be used. A sample FLAP card will be given after the description in "DIAG Card."

OPTIONS FOR BOTH DESIGN AND ANALYSIS MODES

After obtaining an airfoil in either the design or analysis mode, several options are available for generating additional printed, punched, and plotted output. All of the options are independent of the mode in which the airfoil was obtained.

ALFA Card

The ALFA card has two different functions. First, it initiates the storage of up to 14 values of α as well as the pitching-moment and hinge-moment coefficients for those values, all of which are available for other options. Second, it initiates additional output, called the x-y-v listing, containing the x/c and y/c airfoil coordinates and the velocity (or pressure-coefficient) distributions for all values of α . This listing is printed according to print mode mxy. Also, a listing of the moment coefficients is printed according to print mode mcm.

mcm = NUPA (I1):

mcm = 1 - Printout of moment coefficients for all values of α

mcm \neq 1 - No printout

mcm is always set to the value of NUPA if NUPU \neq 0 and, therefore, mcm is only in effect until the next ALFA card with NUPU \neq 0 is read.

If NUPA \neq 0, mxy = NUPE(I1):

mxy = 0 - No printout

mxy \neq 0 - The airfoil coordinates and the velocity (or C_p) distributions for all values of α (i.e., x-y-v listing) are printed.

If NUPA = 0 (normal), mxy remains as previously set. The preset value is mxy \neq 0.

If NUPA = 1, in addition to the above, the subroutine PANEL print mode mpa is set to NUPE + 1. The usual combination for moment-coefficient printouts is NUPA = 1 and NUPE = 0, which results in only the headlines from subroutine PANEL being printed between the moment-coefficient listings while the coordinate and velocity-distribution listings (x-y-v listings) are suppressed. This combination yields moment-coefficient listings which are easily surveyed.

NUPI = 0 - The values of α_i are interpreted as being relative to the zero-lift line, and the x-y-v listing contains velocity (v/U_∞) distributions.

NUPI = 1 - The values of α_i are interpreted as being relative to the chord line, and the x-y-v listing contains velocity distributions.

NUPI = 2 - The values of α_i are interpreted as being relative to the zero-lift line, and the x-y-v listing contains pressure-coefficient (C_p) distributions.

NUPI = 3 - The values of α_i are interpreted as being relative to the chord line, and the x-y-v listing contains pressure-coefficient distributions.

NUPU = n_α = the number of F-words to be read (I3)

NUPU = 0 - NUPI, n_α , and the values of α_i from the immediately preceding ALFA card with NUPU \neq 0 are used. If the first ALFA card contains NUPU = 0, the program will terminate.

F_i ($i = 1, n_\alpha$) - Values of α_i in degrees (F5.2)

If $F_i \leq -99$, $\alpha_i = \alpha_i^*$. This option is sometimes helpful if the values of α_i^* have been iterated.

Internally, the program uses only α relative to the zero-lift line. If NUPI = 1 or 3, all values of $\alpha_{c,i}$ are translated into values of α_i relative to the zero-lift line and then back to values of $\alpha_{c,i}$ relative to the chord line only for output. If $F_i \leq -99$, the value of α_i ($= \alpha_i^*$) is always relative to the zero-lift line. This option yields constant velocity over the i th segment of the airfoil. If NUPI = 1 or 3, the value of α_i is translated into the value of $\alpha_{c,i}$ relative to the chord line for output which, therefore, does not give α_i^* but instead, $\alpha_i^* - \alpha_0$. If $F_i \leq -99$ occurs with $i > I_a$ or before the design mode is used, the program will terminate.

The following cards illustrate some of the ALFA-card options.

```
ALFA21 4 100-9999 300 300
```

The moment coefficients are not printed ($mcm = NUPA = 2 \neq 1$); the x-y-v listing containing velocities is printed ($NUPA \neq 0$, therefore, $mxy = NUPE = 1$; $NUPI = 0$); and $\alpha_1 = 1^\circ$, $\alpha_2 = \alpha_2^*$, $\alpha_3 = 2^\circ$, and $\alpha_4 = 3^\circ$, all relative to the zero-lift line ($NUPI = 0$).

```
ALFA 3 5 200 400 600 800 1000
```

The moment coefficients are not printed ($mcm = NUPA = 0$), and mxy remains as previously set ($NUPA = 0$). If $mxy \neq 0$, the x-y-v listing containing pressure coefficients is printed ($NUPI = 3$) with $\alpha_i = 2^\circ, 4^\circ, 6^\circ, 8^\circ$, and 10° relative to the chord line ($NUPI = 3$).

```
ALFA
```

NUPI, n_α , and the values of α_i from the immediately preceding ALFA card with $NUPI \neq 0$ are used.

```
ALFA20
```

NUPI, n_α , and the values of α_i from the immediately preceding ALFA card with $NUPI \neq 0$ are used. The moment coefficients and the x-y-v listing are not printed ($NUPA = 2 \neq 0$, therefore, $mxy = NUPE = 0$).

ORIGINAL PAGE IS
OF POOR QUALITY

The following card sequence contains the cards required to print those results which are plotted in figures 4, 5, and 7.

TR41 1095 000 370 5000 -370

TR42 1095 400 1450 200-1000 612 400 1450 200-1000 612 000 000 000 000

DUFA 2 000 370

TR41 1096 000 400 5000 -100

TR42 1096 400 1450 200-1000 612 400 1450 200-1000 612 300 000 000 000

DUFA 2-9999 000

TR41 1097 1450 300 000 900 5000 200

TR42 1097 400 1450 000 627 1000 400 1450 000 627 1000 700 400 000 000

DUFA 4 300-9999 300 1200

PUXY Card

This card initiates only the punching of the airfoil coordinates x/c and y/c and the inner normal angles β onto cards via tape 4. Columns 1-5 on each punched card contain the airfoil name. If this name comes from the analysis mode, columns 3-7 of the name are used. If this name comes from the design mode, one blank followed by the four digits of the airfoil number are used. Columns 6-8 contain the number of the first point punched on that card. Columns 9-30 contain four groups of three numbers each which are the x/c , y/c , and β values for one point. No decimal points are punched. The format is such that the x/c and y/c values can be read in format F6.5 and the β values in format F6.4. The punching is performed in a separate subroutine PUDECK which is easily modified to allow other formats.

DIAG Card

This card initiates the plotting of two different diagrams.

NUPA and NUPE are neglected.

NUPI = 0 - A diagram containing the airfoil shape and the velocity distributions for all the values of α on the preceding ALFA card is plotted. If $F_1 \neq 0$, $100F_1$ = the length of the x-axis in mm [F5.0]. The preset value of F_1 is 177.8 mm (7.000 in.).

NUPI = 1 - A so-called pressure envelope diagram is plotted with $|C_{p,\min}|$ as the abscissa and α as the ordinate. All values of α from the preceding ALFA card are used. The range of α values which can be plotted is from -5° to 15° . The α -axis is only drawn as long as necessary within this range, however. If this range and, accordingly, the α -axis must be extended, the lower and upper limits can be specified by $F_1 = \alpha_{\min}$ in degrees (F5.2) and $F_2 = \alpha_{\max}$ in degrees (F5.2). If $F_1 = 0$, α_{\min} remains as previously set. If $F_2 = 0$, α_{\max} remains as previously set.

mbt = NJPU (I3) - This option allows several different sets of data to be plotted in one diagram.

mbt = 0 - Axes are drawn, one set of data is plotted, and the diagram is terminated (i.e., closed to further plotting).

mbt = 1 - Axes are drawn, one set of data is plotted, and the diagram is open to further plotting.

mbt = 2 - No axes are drawn and one set of data is plotted into the existing diagram which is then terminated.

mbt = 3 - No axes are drawn and one set of data is plotted into the existing diagram which remains open to further plotting.

Obviously, mbt is valid only for the one DIAG card in which it is contained.

It must be remembered that no other plots can be generated between two sets of data to be plotted in the same diagram.

The following card sequence plots into one diagram the two sets of velocity distributions for airfoil 1098 at $\alpha = 2^\circ, 8^\circ, 10^\circ, \text{ and } 12^\circ$ relative to the zero-lift line from the design and the analysis modes.

```

TRAI 1098 2350 300 2750 1000 000 1200 6000 300
-----
TRA2 1098 400 1450 200 1000 650 400 1450 200 1000 650 400 400 000 000
-----
ALFA 4 200 200 1000 1200
-----
DIAG 1
-----
PAN
-----
ALFA
-----
DIAG 2
-----

```

If the printing mode was not set prior to the first ALFA card, two x-y-v listings would be printed and between them a headline from the panel method. The second ALFA card can be left out. The second set of data would be plotted correctly without it although the second x-y-v listing would not be printed.

The following card sequence plots into one diagram the velocity distributions for an airfoil with and without flap deflections. Three velocity distributions, all at $\alpha = 0^\circ$ relative to the chord line, corresponding to flap deflections of 0° , 10° , and -10° are plotted. The cards shown must obviously be preceded by TRAI, TRA2, and PAN cards or by an FXPR card plus airfoil coordinates.

```

ALFA 1 1 000
-----
DIAG 1
-----
FLAP 2500 000 300 1000
-----
ALFA 1 1 000
-----
DIAG 3
-----
FLAP 2500 000 300-1000
-----
ALFA 1 1 000
-----
DIAG 2
-----

```

The specification of none of the parameters on the second and third ALFA cards (i.e., ALFA cards with only the card name on them) or having NUPU = 0 on them would result in the same diagram. If no x-y-v listings are to be printed, the first ALFA card should be

ALFA001 1 000

STRD Card

This card prepares the data for the plotting of airfoils having various chord lengths.

NUPA, NUPE, and NUPI are neglected.

NUPU controls the plot mode mxz for the subsequent plots initiated by STRK cards.

If NUPU \neq 0, mxz = NUPU (I3):

mxz > 0 - The chord line is plotted.

mxz < 0 - The chord line is not plotted.

If NUPU = 0, mxz remains as previously set. The preset value is mxz < 0.

100F₁ = YBL [F5.0]:

|YBL| = the vertical extent of the plot in mm. The sign of YBL determines the interpretation of F₂.

100F₂ = RUA [F5.0]:

RUA = the vertical shift in mm of the chord line from one airfoil to the next, if YBL < 0.

RUA = the vertical shift in mm between the upper surface of one airfoil and the lower surface of the next, if YBL > 0.

STRK Card

This card initiates a plot and a listing of the airfoil coordinates generated by the preceding TRAl and TRA2 or FXPR card for various chord lengths. An STRK card must, obviously, be preceded by an STRD card.

NUPA and NUPE are neglected.

NUPI = 9 - The listing is printed although the plot is suppressed.

NUPI \neq 9 - Both the listing and the plot are generated.

ORIGINAL PAGE IS
OF POOR QUALITY

NUPU = NC = the number of airfoils having various chord lengths to be plotted in one plotting procedure (I3)

NC > 0 - The chord lengths specified by the F-words on this STRK card are stored in the chord array starting from the beginning of the array.

NC < 0 - The chord lengths specified by the F-words on this STRK card are stored in the chord array starting immediately after the chord lengths already stored there.

|NC| > 14 - The diagram is not plotted and the program requires additional chord lengths.

|NC| ≤ 14 - The diagram containing all of the airfoils for which chord lengths have been stored so far is plotted. If NUPI ≠ 9, the listing is printed.

$0.01F_i = c_i$ = the chord lengths in m [F5.4]. Only $0.01F_i$ is used. If $F_i < 0$, the diagram containing the airfoils already plotted is terminated before the diagram containing the airfoil having chord c_i is initiated. Any airfoil having a chord $c_i < 1$ mm is not plotted. If the preceding diagram has been terminated, a new one will not be initiated for an airfoil having a chord $c_i < 1$ mm. It should be noted that a diagram cannot be terminated unless the last $c_i < -1$ mm.

Thus it is possible to plot several airfoils in one diagram. The following sequence of cards plots two different airfoils in one diagram. The airfoils both have chords of 0.500 m and are separated by 5 mm. The chord lines for both airfoils are plotted, and the diagram is terminated at the end of the card sequence. Obviously, the TRAL and TRA2 cards for the first airfoil must precede the first two cards in the sequence while the TRAL and TRA2 cards for the second airfoil must be included between the second and third cards.

TRAL and TRA2 cards (first airfoil)

```
STRD 1 500 5
```

```
STRK 1 5000
```

TRAL and TRA2 cards (second airfoil)

```
STRK 1 5000 -5
```

If the second STRK card had contained $NUPU = 0$, the chord from the preceding STRK card would have been used and the diagram would have remained open to further plotting.

OPTIONS FOR BOUNDARY-LAYER ANALYSIS

The boundary-layer method within the program requires the velocity-distribution data generated by the potential-flow airfoil method. Obviously, the program will terminate if boundary-layer computations are initiated without these data. The velocity distributions can be generated in either the design mode or in the analysis mode.

RE Card

This card specifies up to five pairs of data, each containing a transition mode MU and a Reynolds number R. Boundary-layer computations for every MU-R pair are performed using the velocity distributions $v(x,\alpha)$ for all of the values of α contained on the preceding ALFA card. The program will terminate if there is no ALFA card somewhere ahead of the RE card. The output from the boundary-layer analysis is printed and plotted according to print mode mpr and plot mode mpl. A summary of the results is stored in blank COMMON arrays CW, SA, and SU which remain available for other procedures.

If $NUPA \neq 0$, $mpr = NUPE (I1)$ and $mpl = NUPI (I1)$:

$mpr = 0$ - no printout

$mpr = 1$ - Only the summary which contains the lift, drag, and pitching-moment coefficients, including viscous corrections, as well as the arc lengths of turbulent and separated flows on both airfoil surfaces plus an indication of the laminar separation bubble analog is printed.

$mpr > 1$ - In addition to the summary, the boundary-layer development along both surfaces for every MU-R pair and all of the values of α contained on the preceding ALFA card is printed. This printout contains in the first two columns the arc length s/c starting at the stagnation point and the potential-flow velocity v/U_∞ . Then, for each MU-R pair, two columns are printed; the first one always contains H_{32} and the second one contains δ_2 if $mpr = 2$, $10^{-6}R\delta_2$ if $mpr = 3$, or δ_1 if $mpr = 4$. This second column is entitled DELTA2, RDELTA2, or DELTA1.

$mpl = 0$ - no plots

$mpl = i \leq 5$ - The boundary-layer development for the i th MU-R pair is plotted as R_{δ_2} versus H_{32} as described in "Boundary-Layer Development." Two plots are generated. The left one is for the upper surface of the airfoil and the right one for the lower surface. Each plot contains one curve for each of the values of α contained on the preceding ALFA card.

If $NUPA = 0$, mpr and mpl remain as previously set. The preset values are $mpr = 1$ and $mpl = 0$.

$NUPU$ is neglected.

F_1 - first digit = the suction mode MA (normally zero)

second digit = the transition mode MU

$10^5 F_2$ = Reynolds number = $10^{-3} R$ [F5.0]

If $F_2 = 0$, the MU-R pairs and the values of x_T/c from the preceding RE card are used.

$(F_3, F_4) = (MU_2, R_2)$ and so on up to $(F_9, F_{10}) = (MU_5, R_5)$

$F_{11} - F_{14}$ are only required if the transition mode $MU = 1$ or 2 (i.e., fixed transition). In this situation,

$0.01 F_{11} = (x_T/c)_{us}$ for $MU = 1$ [F5.4]

$0.01 F_{12} = (x_T/c)_{ls}$ for $MU = 1$ [F5.4]

$0.01 F_{13} = (x_T/c)_{us}$ for $MU = 2$ [F5.4]

$0.01 F_{14} = (x_T/c)_{ls}$ for $MU = 2$ [F5.4]

The values of x_T/c are only in effect until an RE card with $F_2 \neq 0$ is read. The transition modes are

$MU = 0$ - transition at laminar separation

$MU = 1$ or 2 - fixed transition at $(x_T/c)_{us}$ and $(x_T/c)_{ls}$

$MU = 3$ - natural transition (see inequality (58))

$MU > 3$ - transition with roughness factor $r = MU - 3$ (see inequality (58))

The following RE card initiates three boundary-layer computations for each value of α contained on the preceding ALFA card. The first computation is for $R = 1 \times 10^6$ with fixed transition at $(x/c)_{us} = 0.05$ and

$(x/c)_{1s} = 0.07$. The second computation is for $R = 3 \times 10^6$ with fixed transition at $(x/c)_{us} = 0.10$ and $(x/c)_{1s} = 0.12$. The third computation is for $R = 1 \times 10^5$ with natural transition. The boundary-layer development containing R_{δ_2} is printed for all three Reynolds numbers and plotted for $R = 1 \times 10^5$.

```

RE 122 01 100002 300002 100 500 700 1000 1200

```

The following typical card sequence generates the boundary-layer diagram, of which the left-hand plot is shown in figure 13, as well as the transcendental equation results listed on page 44.

```

TR91 1093 2250 300 2750 1000 000 1200 6000 200

```

```

TR92 1093 400 1450 200 1000 650 400 1450 200 1000 650 600 400 000 000

```

```

SLEP 200 300 1000 1200 1300 1400

```

```

RE 111 02 100003 3000

```

```

ENDP

```

This card sequence also generates the x-y-v listing and the boundary-layer summary which follow:

Reproduced from
best available copy.



AIRFOIL 1098	18.97%	2.00	8.00	10.00	12.00	13.00	14.00
N	X	Y	VELOCITY	DISTRIBUTIONS FOR THE ABOVE ANGLES OF ATTACK RELATIVE TO THE ZERO-LIFT LINE			
0	1.00000	0.00000	.788	.780	.776	.771	.768
1	.99623	.00097	.809	.806	.803	.799	.795
2	.98574	.00407	.870	.871	.870	.867	.863
3	.96952	.00915	.938	.945	.945	.944	.942
4	.94811	.01546	.978	.990	.992	.993	.992
5	.92147	.02275	.991	1.009	1.013	1.015	1.016
6	.88988	.03121	1.008	1.033	1.039	1.043	1.045
7	.85403	.04088	1.030	1.062	1.069	1.076	1.079
8	.81463	.05162	1.057	1.096	1.106	1.115	1.119
9	.77245	.06320	1.089	1.136	1.149	1.161	1.166
10	.72823	.07529	1.127	1.183	1.199	1.213	1.220
11	.68274	.08746	1.170	1.237	1.257	1.274	1.283
12	.63668	.09920	1.220	1.300	1.323	1.345	1.355
13	.59071	.10993	1.278	1.371	1.399	1.426	1.438
14	.54543	.11878	1.343	1.453	1.487	1.518	1.533
15	.50068	.12472	1.372	1.499	1.537	1.574	1.591
16	.45573	.12763	1.399	1.499	1.541	1.583	1.602
17	.41054	.12819	1.344	1.499	1.546	1.592	1.615
18	.36565	.12682	1.328	1.499	1.552	1.603	1.628
19	.32157	.12367	1.310	1.499	1.558	1.615	1.643
20	.27879	.11891	1.290	1.499	1.565	1.629	1.660
21	.23777	.11267	1.267	1.499	1.572	1.644	1.679
22	.19896	.10510	1.240	1.499	1.581	1.662	1.702
23	.16280	.09631	1.208	1.499	1.592	1.683	1.728
24	.12963	.08640	1.164	1.492	1.598	1.702	1.753
25	.09971	.07553	1.106	1.477	1.598	1.717	1.775
26	.07328	.06397	1.032	1.459	1.598	1.735	1.803
27	.05064	.05198	.937	1.435	1.598	1.759	1.839
28	.03203	.03975	.799	1.389	1.582	1.774	1.869
29	.01753	.02755	.597	1.308	1.542	1.774	1.889
30	.00720	.01585	.298	1.187	1.482	1.774	1.919
31	.00127	.00529	.198	.988	1.382	1.774	1.969
32	.00047	-.00298	1.184	.592	1.184	1.774	2.068
33	.00629	-.00989	1.201	.172	.172	.516	.687
34	.01868	-.01708	1.201	.482	.241	.000	.121
35	.03645	-.02431	1.201	.650	.465	.279	.186
36	.05916	-.03135	1.201	.756	.606	.455	.380
37	.08643	-.03802	1.201	.830	.704	.577	.513
38	.11788	-.04419	1.201	.883	.775	.666	.612
39	.15310	-.04971	1.201	.925	.831	.735	.687
40	.19166	-.05445	1.201	.958	.875	.790	.748
41	.23307	-.05827	1.201	.985	.911	.835	.797
42	.27665	-.06102	1.201	1.008	.941	.873	.839
43	.32247	-.06254	1.201	1.028	.967	.906	.875

AIRFOIL 1098	18.97%	2.00	8.00	10.00	12.00	13.00	14.00
N	X	Y	VELOCITY	DISTRIBUTIONS FOR THE ABOVE ANGLES OF ATTACK RELATIVE TO THE ZERO-LIFT LINE			
44	.36936	-.06252	1.201	1.045	.990	.934	.906
45	.41693	-.06079	1.201	1.060	1.010	.959	.933
46	.46524	-.05822	1.164	1.041	.997	.952	.929
47	.51509	-.04922	1.099	.993	.955	.916	.896
48	.56663	-.04074	1.041	.951	.915	.884	.867
49	.61917	-.03179	.991	.913	.885	.856	.841
50	.67190	-.02303	.948	.881	.856	.831	.817
51	.72391	-.01501	.910	.853	.831	.809	.797
52	.77422	-.00917	.878	.828	.810	.790	.780
53	.82181	-.00280	.851	.808	.792	.775	.766
54	.86565	.00093	.828	.792	.778	.763	.755
55	.90472	.00303	.809	.778	.766	.753	.747
56	.93808	.00364	.794	.768	.758	.747	.741
57	.96486	.00304	.785	.765	.756	.747	.742
58	.98433	.00177	.784	.768	.761	.753	.749
59	.99608	.00053	.786	.775	.769	.762	.759
60	1.00000	-.00000	.788	.780	.776	.771	.768

ALPHA = 4.90 DEGREES CMO = -.1237 ETA = 1.137

SUMMARY AIRFOIL 1098 ANGLE OF ATTACK RELATIVE TO THE ZERO-LIFT LINE ALPHA = 4.90 DEGREES
 † INDICATES BUBBLE ANALOG LONGER THAN .030

R = 1000000 MU = 3 R = 3000000 MU = 3

ALPHA = 2.00 DEGREES
 1 S TURB S SEP CD 2 S TURB S SEP CD
 UPPER .4623 .0043 .0042 .4775 0.0000 .0033
 LOWER .9519 0.0000 .0030† .5716 0.0000 .0024
 TOTAL CL = .217 CD = .0072 CL = .220 CD = .0057
 CM = -.1271 CM = -.1279

ALPHA = 8.00 DEGREES
 1 S TURB S SEP CD 2 S TURB S SEP CD
 UPPER .4890 .0189 .0060 .5062 .0084 .0046
 LOWER .5215 0.0000 .0021 .5329 0.0000 .0017
 TOTAL CL = .859 CD = .0080 CL = .871 CD = .0062
 CM = -.1438 CM = -.1466

ALPHA = 10.00 DEGREES
 1 S TURB S SEP CD 2 S TURB S SEP CD
 UPPER .4999 .0271 .0068 .5194 .0129 .0052
 LOWER .5128 0.0000 .0018 .5247 0.0000 .0015
 TOTAL CL = 1.067 CD = .0086 CL = 1.084 CD = .0067
 CM = -.1462 CM = -.1523

ALPHA = 12.00 DEGREES
 1 S TURB S SEP CD 2 S TURB S SEP CD
 UPPER .5118 .0428 .0078 .8077 .0614 .0109
 LOWER .5013 0.0000 .0016 .5145 0.0000 .0013
 TOTAL CL = 1.263 CD = .0094 CL = 1.238 CD = .0122
 CM = -.1501 CM = -.1449

ALPHA = 13.00 DEGREES
 1 S TURB S SEP CD 2 S TURB S SEP CD
 UPPER .6568 .0957 .0111 .8628 .1035 .0127
 LOWER .4935 0.0000 .0015 .5080 0.0000 .0012
 TOTAL CL = 1.298 CD = .0126 CL = 1.287 CD = .0139
 CM = -.1386 CM = -.1367

ALPHA = 14.00 DEGREES
 1 S TURB S SEP CD 2 S TURB S SEP CD
 UPPER .8327 .1751 .0160 .9342 .1489 .0158
 LOWER .4811 0.0000 .0014 .5000 0.0000 .0012
 TOTAL CL = 1.288 CD = .0174 CL = 1.326 CD = .0170
 CM = -.1236 CM = -.1287

FLZW Card

This card initiates aircraft-oriented boundary-layer computations, where the Reynolds number R varies with the aircraft lift coefficient C_L and the local chord c . A local twist angle θ , which must be specified relative to the zero-lift line of the entire wing, can also be included. The values of α specified on the preceding ALFA card are considered as the local angles of attack relative to the zero-lift line. If the values of α on the ALFA card were specified relative to the chord line, they are converted to α relative to the zero-lift line by adding the zero-lift angle before they are used in these computations. Thus, the aircraft lift coefficient is

$$C_L \approx 0.11(\alpha + \theta)$$

and the aircraft speed is

$$v = \frac{v_1}{\sqrt{C_L}}$$

where

$$v_1 = \sqrt{\frac{2gW}{\rho S}}$$

is the aircraft's speed at $C_L = 1$ and g is the acceleration due to gravity. To prevent v from attaining infinite values as $C_L \rightarrow 0$, a maximum aircraft speed v_{\max} is specified such that

$$v = \text{minimum} \left\{ \frac{v_1}{\sqrt{0.11(\alpha + \theta)}}, v_{\max} \right\}$$

This speed v is used to calculate the Reynolds number

$$R = \frac{Vc}{\nu}$$

which, therefore, depends not only on the local angle of attack and aircraft weight, but also on the local twist angle and local chord. Everything else is processed exactly as it is for an RE card, including the print and plot modes and storage of the summary of the results.

If $NUPA \neq 0$, $mpr = NUPE$ (I1) and $mpl = NUPI$ (I1).

If $NUPA = 0$, mpr and mpl remain as previously set. The preset values are $mpr = 1$ and $mpl = 0$.

MU = NUPU (I3):

MU is the transition mode as defined in "RE Card." Fixed transition (MU = 1 or 2) is allowed if the transition points are specified on a preceding RE card.

$F_1 = W/S =$ wing loading (aircraft mass/wing area) in kg/m^2 (F5.2)

$F_2 = V_{\max}$ = maximum aircraft speed in m/s (F5.2)

$0.1F_3 = \rho$ = air density in kg/m^3 [F5.3]

If $F_3 = 0$, ρ remains as previously set. The preset value is $\rho = 1.229 \text{ kg/m}^3$.

$F_4 = 10^6 \nu$ = kinematic viscosity in m^2/s (F5.2)

If $F_4 = 0$, ν remains as previously set. The preset value is $\nu = 13.6 \times 10^{-6} \text{ m}^2/\text{s}$.

$F_5 = c$ = chord length in m (F5.2)

$F_6 = \theta$ = local twist angle in degrees (F5.2)

$(F_7, F_8) = (c_2, \theta_2)$ and so on up to $(F_{13}, F_{14}) = (c_5, \theta_5)$

If $F_5 = 0$, the FLZW card does not initiate a boundary-layer computation. Instead, the values of V_{\max} , ρ , and ν are stored for a succeeding PLW card.

A sample FLZW card will be given after the description of the PLWA card.

PLW Card

This card initiates an aircraft-oriented boundary-layer computation which uses the given aircraft data and results in a polar for the given aircraft. The wing planform is specified and, therefore, the aspect ratio AR can be computed and the influence of the local chord lengths on the Reynolds number can be considered. Knowing AR allows the induced-drag coefficient $C_{D,i}$ to be computed from

$$C_{D,i} = \frac{1.03C_L^2}{\pi AR}$$

where the constant 1.03 assumes a good planform. The effect of wing twist is not considered. The parasite-drag area A_p (= parasite drag/dynamic pressure) is assumed to be a constant which means the aircraft parasite-drag coefficient is

$$C_{D,p} = \frac{A_p}{S}$$

where S is the wing area.

If only the influence of various airfoil sections is to be evaluated for a given aircraft configuration, the relative thicknesses of the airfoils must be considered. For this purpose, the PLW card contains a special option which allows the wing planform to be altered such that the absolute thickness of the different wings is a constant. To achieve this, a reference thickness t^*/c must be specified and, if an airfoil having a thickness t/c is to be evaluated, its chord lengths are multiplied by t^*/t before any evaluations are performed. A weight penalty due to the increased wing area which arises from the greater chord lengths is also computed. This procedure allows the influence of the airfoil on the overall aircraft performance to be evaluated in a somewhat more realistic manner than if the same planform were used for airfoils having different thickness ratios.

Everything else is processed as it is for an FLZW card. The values of α from the preceding ALFA card are used. Because the aircraft speed increases rapidly as the lift coefficient is decreased, more values of α should be specified in the lower lift-coefficient range. The print and plot modes cannot be changed by the PLW card. The values of V_{max} , ρ , and v from the preceding FLZW card are used.

NUPA, NUPE, and NUPI are neglected.

MU = NUPU (I3):

MU is the transition mode as defined in "RE Card." Fixed transition (MU = 1 or 2) is allowed if the transition points are specified on a preceding RE card.

$0.01F_1 = t^*/c =$ reference airfoil thickness [F5.4]

If $F_1 < 0$, $t^*/c = t/c =$ the thickness of the airfoil being evaluated.

If $F_1 = 0$, MU, t^*/c , W^* , DWS, A_p , and the reference planform remain as previously specified.

$F_2 = W^* =$ the aircraft mass in kg (F5.2)

$F_3 = DWS =$ the mass penalty factor in kg/m^2 (F5.2)

$10^{-3}F_4 = A_p =$ the parasite-drag area in m^2 [F5.5]

$F_5 = c^* =$ the chord length in m (F5.2)

$F_6 = db =$ the length in m of the spanwise section having chord length c^* (F5.2)

$(F_7, F_8) = (c_2^*, db_2)$ and so on up to $(F_{13}, F_{14}) = (c_5^*, db_5)$

The wing planform is specified in words $F_5 - F_{14}$ by spanwise sections of constant chord length. A linearly tapered wing of 15-m span having a root chord of 1.2 m and a tip chord of 0.6 m could, therefore, be described by three 5-meter sections ($db = 5$) having average chord lengths c^* of 1.1 m, 0.9 m, and 0.7 m. The same planform could be described by five 3-meter sections ($db = 3$) having average chord lengths c^* of 1.14 m, 1.02 m, 0.90 m, 0.78 m, and 0.66 m. Note that the sum of all of the db values must equal the span of the aircraft, not the half span.

The chord length which is finally evaluated is

$$c = c^* \frac{t^*}{t}$$

The wing area is

$$S = S^* \frac{t^*}{t}$$

where the reference wing area is

$$S^* = \sum c^* db$$

The aircraft mass is

$$W = W^* + (S - S^*) DWS$$

Thus, if no alteration of the planform is desired, F_1 must be negative, which sets $t^*/c = t/c$. A sample PLW card will be given after the description of the PLWA card.

PLWA Card

This card initiates some simple modifications to the polar generated by the preceding PLW card. Thus a PLW card must precede a PLWA card, but any number of PLWA cards can succeed one PLW card.

The PLWA card allows the wing area S , aircraft mass W , and parasite-drag area A_p to be modified. Only the effect of these modifications on the lift and parasite-drag coefficients is considered. Because the lift coefficients are affected, so too are the aircraft speeds and the induced-drag coefficients. No influence on the Reynolds number from the altered chord lengths and speeds is considered. The profile-drag coefficients from the computations initiated by the preceding PLW card are used.

NUPA, NUPE, and NUPI are neglected.

NMOD = NUPU = the number of modifications (I3)

$F_1 = \Delta S =$ the change in wing area in m^2 (F5.2)

$F_2 = \Delta W =$ the change in aircraft mass in kg (F5.2)

$10^{-3}F_3 = \Delta A_p =$ the change in parasite-drag area in m^2 [F5.5]

The program adds $n\Delta S$ to S , $n\Delta W$ to W , and $n\Delta A_p$ to A_p and prints the resulting polars, where n varies from 1 to NMOD. Thus, if NMOD = 3, three new polars would be generated if only one parameter is to be changed; the other changes must be zero.

The following card sequence illustrates the use of FLZW, PLW, and PLWA cards. Obviously, the two sets of cards must be preceded by the TRAL and TRA2 cards or FXPR card plus airfoil coordinates of the first and second airfoils.

TRAL and TRA2 cards or FXPR card plus airfoil coordinates (first airfoil),

```
ALF0200 14 000 100 200 300 400 500 600 700 800 900 1000 1100 1200 1300
```

```
FLZW110 3 600010000 1239 1260 120 100 100 000 80 -100
```

```
ALF0 14 60 80 110 160 250 350 500 700 900 1100 1200 1500 1700 1900
```

```
PLW 3 -140000 500 4000 100 500 80 500 60 500
```

TRAL and TRA2 cards or FXPR card plus airfoil coordinates (second airfoil)

```
PLW
```

```
PLWA 3 50 300 000
```

```
PLWA 2 000 000 200
```

After the design or analysis of the first airfoil, the above card sequence initiates the following computations:

- (1) An aircraft-oriented boundary-layer computation for a wing loading of 60 kg/m^2 ; a maximum speed of 100 m/s ($= 360 \text{ km/h}$); chord lengths of 1.2 m , 1.0 m , and 0.8 m ; twist angles of 1° , 0° , and -1° ; and angles of attack from 0° to 13°

- (2) A polar computation including the corresponding aircraft-oriented boundary-layer computation, for a reference airfoil thickness equal to the thickness of the given airfoil, an aircraft mass of 400 kg, a mass penalty factor of 5 kg/m², and a parasite-drag area of 0.04 m² (The wing planform is defined by a 5-m spanwise section having a chord of 1.0 m, a 5-m section having a chord of 0.8 m, and a 5-m section having a chord of 0.6 m. Thus, the span of the aircraft is 15 m, the wing area 12 m², and the aspect ratio 18.75. The planform is independent of the thickness of the airfoil being evaluated because $t^*/c = t/c$. The maximum speed is obtained from the preceding FLZW card and is, therefore, 360 km/h. The angles of attack contained on the second ALFA card are specified such that the high-speed end of the polar is analyzed as thoroughly as the low-speed end.)

The x-y-v listings are suppressed by the first ALFA card. The boundary-layer development listing and the corresponding plot are suppressed by the FLZW card which specifies that only the boundary-layer summary be printed. If the computation initiated by the FLZW card is not desired, this card still cannot be eliminated but, instead, must be included with $F_5 = 0$.

After the design or analysis of the second airfoil, the final three cards initiate the following computations:

- (1) The same aircraft-oriented boundary-layer and polar computations as performed for the first airfoil, except that the chord lengths are now $c_i = c_i^* \frac{t^*}{t}$ where t^* is equal to the thickness of the first airfoil (Thus, the wing area is altered and a weight penalty is assessed. If these alterations are to be suppressed, the PLW card must contain $F_1 < 0$ although this cannot be specified without repeating the other F-words as contained on the preceding PLW card.)
- (2) Three additional polar computations, each based upon a 0.5-m² increase in wing area and a 3-kg increase in mass starting from the values contained on the preceding PLW card
- (3) Two more polar computations, each based upon a 0.002-m² increase in parasite-drag area starting from the values contained on the preceding PLW card, not the values generated by the preceding PLWA card

CDCL Card

This card initiates the plotting of the boundary-layer summary which contains the lift, drag, and pitching-moment coefficients including the viscous corrections. The boundary-layer summary listed on page 60 is plotted in figure 16. The labeling of the various curves was not generated by the program but was instead added to the diagram for clarification. Note that the abscissa for the transition and separation curves is not x/c but, instead, $1 - s/c$

where s is the arc length of turbulent or separated flow. The curves for the first MU-R pair are solid followed by up to four different types of broken lines each of which corresponds to a different MU-R pair. It is possible to change the types of broken lines which are drawn.

A CDCL card can also follow an FLZW or PLW card. It should be remembered, however, that for these cards the Reynolds number varies with lift coefficient. The curves which are plotted correspond to the various chord lengths and, from the FLZW card, to the twist angles as well.

As previously described for the DIAG and STRK cards, it is possible to plot the results from several computations in one diagram. The range of lift coefficients which can normally be plotted is from -0.4 to 2.4. If the lift coefficients to be plotted by the first CDCL card extend outside this range, the c_l -axis is automatically extended to encompass their values. This is not true for the lift coefficients to be plotted in the same diagram by succeeding CDCL cards. Accordingly, the upper and lower limits of the lift coefficients to be plotted can be specified on the CDCL card. It should be noted that the lift-coefficient range can only be extended and, therefore, the specification of an upper limit of, say, 2.0 would have no effect. The upper limit only of the drag coefficients to be plotted can also be specified on the CDCL card. This limit can have any value greater than zero. The origin of the c_d -axis is not necessarily zero, however. If the minimum drag coefficient to be plotted is greater than 0.01, which frequently occurs at low Reynolds numbers, the origin of the c_d -axis is automatically shifted in steps of 0.005. Thus, the boundary-layer summary listing should be checked before labeling the c_d -axis.

NUPA = 0 - The diagram containing the boundary-layer summary is plotted.

NUPA = 1 - The types of broken lines to be drawn are specified by the F-words. The description of this option follows the details of the diagram.

NUPE and NUPI are neglected.

mbt = NUPU (I3) - This option allows several different sets of data to be plotted in one diagram.

mbt = 0 - Axes are drawn, one set of data is plotted, and the diagram is closed to further plotting.

mbt = 1 - Axes are drawn, one set of data is plotted, and the diagram is open to further plotting.

mbt = 2 - No axes are drawn, and one set of data is plotted into the existing diagram which is then closed to further plotting.

mbt = 3 - No axes are drawn, and one set of data is plotted into the existing diagram which then remains open to further plotting.

Obviously, mbt is valid only for the one CDCL card in which it is contained.

$F_1 = c_{l,\min}$ = the lower limit of the lift coefficients to be plotted (F5.2);
thus, if the lift coefficient to be plotted is
 $c_l < c_{l,\min} < -0.4$, it will not be plotted

If $F_1 = 0$, $c_{l,\min} = -0.4$.

$F_2 = c_{l,\max}$ = the upper limit of the lift coefficients to be plotted (F5.2);
thus, if the lift coefficient to be plotted is
 $c_l > c_{l,\max} > 2.4$, it will not be plotted

If $F_2 = 0$, $c_{l,\max} = 2.4$.

F_3 is neglected.

$0.01F_4 = c_{d,\lim}$ = the upper limit of the drag coefficients to be plotted
[F5.4]; thus, if the drag coefficient to be plotted
is $c_d > c_{d,\lim}$, it will not be plotted

If $F_4 = 0$, $c_{d,\lim}$ is essentially infinity.

If $NUPA = 1$, the types of broken lines to be drawn are specified by the F-words as follows:

F_1 - The five digits of F_1 are denoted n_1 to n_5 .

F_2 - The five digits of F_2 are denoted m_1 to m_5 .

$F_3 - F_{14} = l_i$ = the length of the line or the space in mm (F5.2)

If j is the number of the MU-R pair (or chord length-twist angle pair), the line type is specified by m_j numbers from the array l_i , beginning with $i = n_j$. The first length l_i ($i = n_j$) is drawn with the pen down (i.e., a line), the next length l_{i+1} , with the pen up (i.e., a space), and so on. After the last length l_i ($i = n_j + m_j - 1$) is drawn (pen up), the cycle is repeated starting with the first length l_i pen down. If $m_j = 0$, a solid line is drawn. The variable m_j must be divisible by 2 and ≤ 8 . The preset values are

$$m_j = 0, 2, 4, 6, 8$$

$$n_j = 1, 1, 3, 3, 3$$

$$l_i = 5, 1.5, 10, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5$$

These specify the following line types:

- j = 1
- j = 2 Pen down, 5 mm; pen up, 1.5 mm . . .
- j = 3 Pen down, 10 mm; pen up, 1.5 mm; pen down, 1.5 mm; pen up, 1.5 mm . . .
- j = 4 { Long lines = pen down, 10 mm
- j = 5 { Short lines and spaces = 1.5 mm

```
CDCL1 1115502442 500 200 100 200 1000 200 200 200  

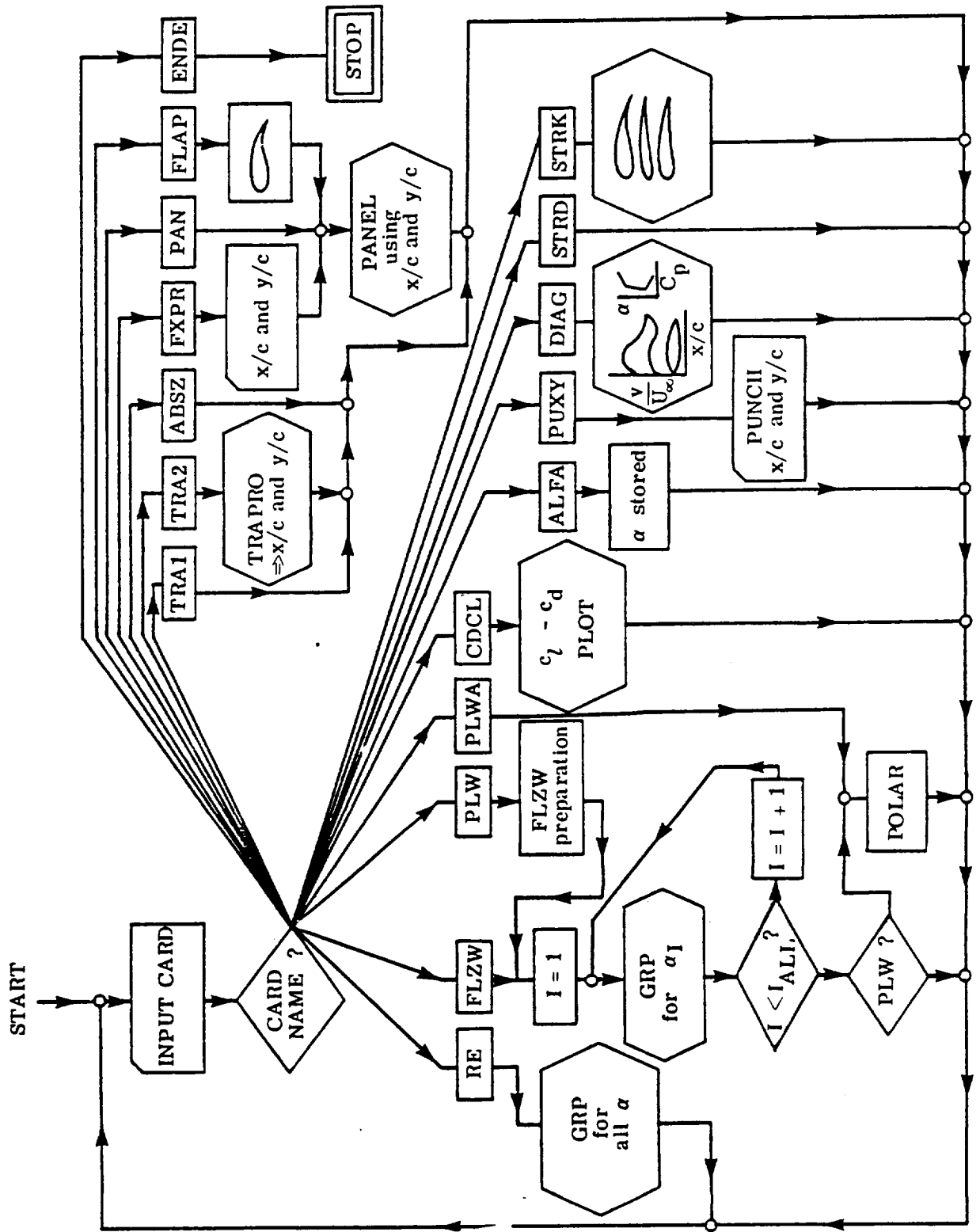
=====
```

The CDCL card above specifies the following line types:

- j = 1
- j = 2 Pen down, 5 mm; pen up, 2 mm . . .
- j = 3 Pen down, 5 mm; pen up, 2 mm; pen down, 1 mm; pen up, 2 mm . . .
- j = 4 Pen down, 10 mm; pen up, 2 mm; pen down, 2 mm; pen up, 2 mm . . .
- j = 5 Pen down, 10 mm; pen up, 2 mm . . .

FLOW CHART AND INPUT-CARD SUMMARY

This section presents a flow chart of the main routine of the program and an input-card summary followed by some rules concerning the sequence of the input cards.



Preceding page blank

Page	Columns 1-4	5	6	7	8-10	11-15	16-20	21-25	26-30	31-35
	Card name	NUJA	NUPE	NUPI	NUPU	F ₁	F ₂	F ₃	F ₄	F ₅
41	TRAI			Airfoil identification number		v ₁ (F5.2)	α ₁ ⁺ (F5.2)	v ₂	α ₂ ⁺	etc.
42	TRA2			Airfoil identification number		λ* (F5.2)	λ (F5.2)	00000 00100 00200	k (F5.3) ω' (F5.3) μ (F5.3)	μ (F5.3) ω (F5.3)
44	ARSZ	0 1	Print mode mtr				ABFA (F5.2)			
45	FXPR	0 1	Print mode mpa			aabdd	aabdd	etc.		
47	FAN	0 1	Print mode mpa			aabdd	aabdd	etc.		
48	FLAP					c _g /c (F5.4)	y _h /c (F5.4)	z _h /2c (F5.4)	δ _g , deg (F5.2)	
48	ALFA	0 1 2	Print mode mxy Print mode mpa Print mode mxy	(*)	n _α	α ₁ , deg (F5.2)	α ₂	etc.		
51	FINV									
52	DIAG			0 - Velocity distribution 1 - Pressure envelope	Plot mode mbt	(†)	.			
54	STRD				Plot mode mxz	± YBL , mm (F5.0)	RUA, mm (F5.0)			
54	STRK			Print and plot modes	NC	c ₁ , m (F5.4)	c ₂	etc.		
56	RE	0 1	Print mode mpr	Plot mode mpl		MU ₁ (F5.3)	10 ⁻³ R ₁ (F5.0)	MU ₂	R ₂	etc.
61	FLZW	0 1	Print mode mpr	Plot mode mpl	MU	w/s, kg/m ² (F5.2)	v _{max} , m/s (F5.2)	ρ, kg/m ³ (F5.3)	10 ⁶ v, m ² /s (F5.2)	c ₁ , m (F5.2)
62	PLW				MU	t*/c (F5.4)	w*, kg (F5.2)	DWS, kg/m ² (F5.2)	A _p , m ² (F5.5)	c ₁ [*] , m (F5.2)
64	PLWA				NMOD	ΔS, m ² (F5.2)	ΔW, kg (F5.2)	ΔA _p , m ² (F5.5)		
66	CDCL				Plot mode mbt	c _{l,min} (F5.2)	c _{l,max} (F5.2)		c _{d,lim} (F5.4)	
40	ENDE									

*ALFA card:

Listing contains	α relative to	
	Zero-lift line	Chord line
	Where NUPI =	
v/u _∞	0	1
C _p	2	3

†DIAG card:

NUPI = 0, F₁ = length of x-axis, mm [F5.0] ≠ 0
 NUPI = 1, F₁ = α_{min}, deg (F5.2) ≠ 0
 NUPI = 1, F₂ = α_{max}, deg (F5.2) ≠ 0

SUMMARY

36-40	41-45	46-50	51-55	56-60	61-65	66-70	71-75	76-80
F ₆	F ₇	F ₈	F ₉	F ₁₀	F ₁₁	F ₁₂	F ₁₃	F ₁₄
$\bar{\lambda}^*$ (F5.2)	$\bar{\lambda}$ (F5.2)	00000 00100 00200	$\bar{\kappa}$ [F5.3] $\bar{\omega}'$ [F5.3] $\bar{\omega}$ [F5.3]	\bar{u} [F5.3] \bar{w} [F5.3] \bar{w} [F5.3]	ITMOD (F5.2)	κ_R [F5.3]	$\kappa_{tol}(=0)$ [F5.3]	Integration scheme (F5.2)
					MU = 1 (x_T/c) _{US} [F5.4]	MU = 1 (x_T/c) _{LS} [F5.4]	MU = 2 (x_T/c) _{US} [F5.4]	MU = 2 (x_T/c) _{LS} [F5.4]
θ_1 , deg (F5.2)	c_2	θ_2	etc.					
db_1 , m (F5.2)	c_2^*	db_2	etc.					

Rules for input-card sequence:

<u>Card name</u>	<u>Must be preceded by</u>
TRA1	---
TRA2	TRA1 card
ABSZ	---
FXPR	---
PAN	TRA2 card
FLAP	FXPR or PAN card
ALFA	TRA2 or FXPR card
PUXY	TRA2 or FXPR card
DIAG	ALFA card
STRD	---
STRK	TRA2 and STRD or FXPR and STRD cards
RE	ALFA card
FLZW	ALFA card
PLW	FLZW card
PLWA	PLW card
CDCL	RE or FLZW or PLW card

DIAG and CDCL cards containing `mbt = 1` or `3` initiate plotting into a diagram which remains open to further plotting. These diagrams must be closed by a card of the same type containing `mbt = 2` before a plot to be generated by another card type can be initiated.

DIAG, PUXY, and STRK cards are not permitted between RE , FLZW, PLW , and CDCL cards.

SYSTEM-DEPENDENT FEATURES OF THE PROGRAM

LANGUAGE

The entire program is written in the FORTRAN IV language. No special options from any given computer system are used other than plotting routines. Therefore, it should be possible to adapt the program to any computer having a FORTRAN IV compiler. The only difficulty which has surfaced so far concerned the DATA statements which initialized variables contained in labeled COMMON blocks. These few DATA statements can be replaced by arithmetic expressions if problems arise.

CORE-STORAGE REQUIREMENTS

The core-storage capability required to run the program depends on the computer and the compiler to be used. On a Control Data 6600 computer system, the program requires approximately 110 000 octal or 37 000 decimal words of core memory when the FTN compiler with `OPT = 2` is used. However, if the usual number of points, 61, is used to define the airfoil, the matrix `A(120,120)`, which is the last array in blank COMMON, can be reduced to `A(60,60)` or `A(70,70)` if

flap deflections are to be analyzed. This saves 10 800 or 9500 decimal words, respectively. In some computer systems, it is possible to truncate the blank COMMON block by simply specifying less memory in the job cards. The array A must contain at least 363 words, even if the analysis mode is not used, where A is the array for the coefficient matrix of the equation system. The array A is used in several subroutines for temporary storage purposes.

No overlaying or segmentation is done in the program. If this is desired, it must be done by the user but it will not be easily accomplished.

INPUT AND OUTPUT

The input and output tape numbers are variables as follows:

ILES = input tape number (card reader)

IDRU = output tape number (line printer)

ISTA = output tape number (card punch)

These numbers are transferred through COMMON/EA/ to all subroutines containing READ or WRITE statements. A DATA statement sets ILES = 5, IDRU = 6, and ISTA = 4.

Carriage control of the printer is performed by function NZPZ(I,K). This function internally counts the lines printed on a page and provides the first character of each line in format A1. Function NZPZ works as follows:

I = 0 - No advance before printing

I = 1 - Space vertically one line, then print

I = 2 - Space vertically two lines, then print

I = 3 - Eject to the first line of the next page, then print

For I = 0, 1, or 2, K more lines on the page of concern must be available after the line to be printed. If not, a new page will be started by setting NZPZ = NNESE. The number of lines per page is specified in line 8 of function NZPZ

```
IF (M+NZEIZA.GT.46) GO TO 1
```

where 46 is the number of lines per page.

PLOTTING SOFTWARE

The program contains a number of useful plotting options. All the diagrams are plotted by only a few subroutines. Subroutine POLZUG plots a polygonal

solid line. A logical external function is included which allows the exclusion of some areas of the plot. If nothing is to be excluded, function TRUE is called. Subroutine SMOOTH plots a spline curve, calling subroutine POLZUG. Subroutine ACHS plots x- and y-axes with linear or logarithmic scales, again calling subroutine POLZUG. Subroutine BLOT produces broken lines and is only called from subroutine POLZUG.

All of the plotting in the program is based upon four very fundamental subroutines: DEFINE, TPLOT, GCLOSE, and FINISH. In their present forms, these subroutines are adapted to plotting software used at NASA Langley Research Center. To maintain the adaptability of the program to other software systems, no labels are written on any of the diagrams plotted by the program.

Subroutine DEFINE(PL,Q,H,XU,XO,YU,YO) prepares for plotting variables x and y in a frame Q mm long in the x -direction and H mm long in the y -direction. The parameters in this subroutine are

XU = the x -value at the left end of the frame,

XO = the x -value at the right end of the frame,

YU = the y -value at the bottom of the frame, and

YO = the y -value at the top of the frame.

Scaling must be performed such that the variables x and y appearing in subroutine TPLOT are plotted correctly within the defined frame. In the present version of the program, this is accomplished by COMMON/GRMIN/ (which is unique to NASA Langley Research Center) as well as the four statements which set the values of the variables contained in COMMON/GRMIN/.

If the software system to be used requires an initialization prior to the first plot, this must be done between the statements "IF (MPL.EQ.1) GO TO 1" and "MPL = 1" in subroutine DEFINE. The statements "CALL PSEUDO" and "CALL LEROY" initialize the plotting system used at NASA Langley Research Center. Thus, the statements which must be adapted to the software system to be used are "CALL PSEUDO" and "CALL LEROY," "CALL CALPLT(...)," COMMON/GRMIN/, and the four statements which set XMIN, YMIN, XSCALE, and YSCALE.

The logical variable PL can be used for switching from one plotting unit to another. This is accomplished in subroutines GRP, DIAGR, and CDCL by using the pen number ISTIFT, which is normally 1 although it can be set in the main routine. The variable ISTIFT also permits the use of different pens on a given plotter. In the present version of the program, neither the switching of pens nor the switching of plotters is performed.

Subroutine TPLOT(X,Y,NPEN,NORIG) moves the pen from its present position to the point (X,Y) in the coordinate system specified by subroutine DEFINE. During the movement of the pen, the pen is up if NPEN = 0 and the pen is down if NPEN \neq 0. NORIG is the variable by which the origin of the coordinate

system can be shifted. This variable is not used in the present version of the program. All the statements in subroutine TPLOT except the first one and the last two are unique to NASA Langley Research Center.

Subroutine GCLOSE terminates a given plot. In some software systems, it is possible to accomplish this by shifting the origin Q mm + some space to the right. In this situation, it is necessary to transfer Q via labeled COMMON from subroutine DEFINE to subroutine GCLOSE. The statement "CALL NFRAME" does this automatically but is unique to NASA Langley Research Center.

Subroutine FINISH terminates all plotting and can be used if the plotting software requires additional processing after completion of the final plot. The statement "CALL CALPLT(0.,0.,999)" is unique to NASA Langley Research Center.

CONCLUDING REMARKS

A method for the design of airfoils with prescribed velocity-distribution characteristics, a method for the analysis of the potential flow about given airfoils, and a method for the analysis of boundary layers have been combined. With this combined method, airfoils with prescribed boundary-layer characteristics can be designed and airfoils with prescribed shapes can be analyzed. The design method is based on conformal mapping. The potential-flow analysis method employs panels with distributed surface singularities. The geometry of the panels is determined by a spline fit of the airfoil coordinates, with the end points of the panels being the input airfoil coordinates themselves. The singularities used are vorticities distributed parabolically along each panel. The boundary-layer method is an integral method which can be applied at Reynolds numbers from 2×10^4 to 1×10^8 . A FORTRAN IV computer program for the numerical evaluation of these methods is available through COSMIC. The program and its input options are described in detail.

This program, then, represents a mathematical model of the two-dimensional viscous flow around airfoils - a computer wind tunnel. The cost of a computer analysis of an airfoil is significantly less than the cost of the corresponding wind-tunnel test. Thus, wind tunnels should be employed increasingly to perform investigations concerning fundamental phenomena. The results from such investigations could then be incorporated into the computer wind tunnel and, thereby, allow airfoils to be theoretically developed for specific applications with a high degree of confidence.

Langley Research Center
National Aeronautics and Space Administration
Hampton, VA 23665
March 14, 1980

APPENDIX

PROGRAM LISTING

```

PROGRAM PROFILE (INPUT,OUTPUT,TAPE4,TAPE5,TAPE6)
DIMENSION XF(121),YF(121),BETAF(121)
DIMENSION AM(7,7),AV(7)
DIMENSION V(14),MARKEN(20),ALCA(14),CAE(2)
DIMENSION RE(5),MA(5),MU(5),T(42)
DIMENSION TM(5),ALS(5),RER(5),MUR(5)
DIMENSION TST(5),BANT(5),CW(5,2,14),SU(5,2,14),SA(5,2,14)
COMMON P1(121),P(121),XP(121),YP(121),PUFF(14),AGAM(14),X(121),
1Y(121),DS(122),VF(121),ARG(121),ANI(28),ALFA(29),IZZ,KFU,NQ,NUPRO,
2JAB,JST,CH,ETA,ABFA,PI,BOGEN,DARG,PURES(13),FUW(60,7),RS(60)
COMMON XTF,SHA,XFL(10),GAHHA(121,2),AMAT(120,120)
COMMON /GRZK/CDK,AA(7),BB(7)
COMMON/PRAL/DLT,DLTU,ALN,ALV(14),NAL,ITP,NAMP(12),CHL(14),CRL(14)
1 ,CPV(2),ALT(4,2),DARF,ITIT1,ITIT2
COMMON/PLTM/MPL,MGC,XZEH,YZEH,MSPLI
COMMON/EA/ILES,IDRU,ISTA,NNESE
COMMON/TRIT/DLV,SUMP,XTRI(4),NU,ND
COMMON /LINING/BROKL(12),NLINE(5),NPAR(5),J
EQUIVALENCE(XF(1),FUW(1,1)),(YF(1),FUW(3,3)),(BETAF(1),FUW(9,5))
EQUIVALENCE (CW(1,1,1),P1(1)),(SA(1,1,1),P(20)),
1 (SU(1,1,1),XP(39))
DATA ILES,IDRU,ISTA,NNESE/5,6,4,1H1/
DATA MARKEN/4HTRA1,4HTRA2,4HALFA,4HAGAH,4HABSZ,4HSTRK,4HENDE,
14HDIAG,4HRE ,4HSTRD,4HFLZW,4HPLWA,4HPLW ,4HTRF ,4HAPPR,4HCDCL,
24HPAN ,4HFXPR,4HFLAP,4HPUXY/
DATA CPV/9HVELOCITY ,9HPRESSURE /,ALT/4H ZER,4HO-LI,4HFT L,
*4HINE ,4H CHO,4HRD L,4HINE ,4H /,KBLT/1H /
DATA MGC,ISTIFT,MXZ,CDK/0,1,-1,.01/
DATA ZAEH,DICHTE/13.6E-6,.12533/
MPL=0
PI = 3.141592654
BOGEN = 0.0174532925199
ABFA = 1.0
AGAM(2)=1.
AGAM(3)=1.
AGAM(6)=1.
AGAM(8)=0.
AGAM(10)=1.
9 MTR=0
11 READ(ILES,2)MARKE,NUPA,NUPE,NUPI,NUPU,PUFF
2 FORMAT(A4,3I1,I3,14F5.2)
DO 12 I=1,20
IF (MARKE.EQ.MARKEN(I)) GO TO 13
12 CONTINUE
14 WRITE(IDRU,3) MARKE
3 FORMAT (11H INCORRECT ,A4,5H CARD)
GO TO 11
C TRALTRA2ALFAAGAHABSZSTRKENDEDIAGRE STRDFLZWPLWAPLW TRF APPRCDC
C PAN FXPRFLAPPUXY

```


APPENDIX

```

13 GO TO(15,22,333,14,142,90,150,104,30,112,30,71,60,14,14,160,
    *170,180,190,106),I
C   TRA1 CARD
15 NUPRO=NUPU+1000*NUPI
    IF(MTR.EQ.0)JST=0
    I=0
18 I=I+1
    ANRI=RUND(PUFF(I)*ABFA,1000.)
    IF(ANRI.NE.0.)GO TO 20
    IF(JST.NE.0.)GO TO 21
    JST=MTR+1
20 MTR=MTR+1
    ANI(MTR)=ANRI
    I=I+1
    ALFA(MTR)=PUFF(I)
    IF(I.NE.14)GO TO 18
21 JAB=MTR
    GOTO 11
C   TRA2 CARD
22 DO 23 I=1,13
23 PURES(I)=RUND(PUFF(I),1000.)
    MSPLI=0
    ITP=0
    IZZ=INT(PUFF(14))
    CALL TRAPRO
    XDA=0.
    YDA=0.
    DEFLG=0.
    GO TO 9
C   RE CARD
25 IF(PUFF(2).EQ.0.) GO TO 28
    DO 27 J=1,5
    RERX= PUFF(2*J)
    IF(RERX.EQ.0.)GO TO 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
26 DO 29 J=1,4
29 XTRI(J)=PUFF(J+10)*.01
28 CALL GRP(NAL,RE,MU,JR,ISTIFT)
    MSPLI=0
    JP=JR
    GOTO 11
C   FLZW CARD
30 IF(NUPA.EQ.0) GO TO 31
    AGAM(6)=FLOAT(NUPE)
    AGAM(8)=FLOAT(NUPI)

```

APPENDIX

```

31 IF(I.EQ.9) GO TO 25
   IF(PUFF(2).EQ.0.) GO TO 36
   GDF = PUFF(1)
   VMAX = PUFF(2)
   IF(PUFF(3).NE.0.) DICHTE = .1*PUFF(3)/9.806
   IF(PUFF(4).NE.0.) ZAEH = PUFF(4)*1.E-6
   IF(PUFF(5).EQ.0.) GO TO 50
   D = 0.
   DO 34 J = 1,5
   JZ = 2*J + 3
   IF(PUFF(JZ).EQ.0.) GO TO 36
   TM(J)=PUFF(JZ)
   ALS(J)=PUFF(JZ+1)
   MUR(J) = NUPU
34 JT = J
36 IZT=NZPZ(2,6*NAL+2)
   JP=JT
   WRITE(IDRU,37)IZT,NAMP,(ALTX(J,ITIT2),J=1,4)
37 FORMAT (A1,36HAIRCRAFT-ORIENTED SUMMARY AIRFOIL ,12A1,3X,
  *31HANGLE OF ATTACK RELATIVE TO THE,4A4)
   IVMAX=INT(VMAX*3.6)+1
   IZT=NZPZ(2,0)
   WRITE(IDRU,38)IZT,GDF,IVMAX,DICHTE,ZAEH
38 FORMAT (A1,6H W/S =,F6.2,8H KG/SQ.M,3X,7HV MAX =,I4,5H KM/H,3X,
  *5HRHD =,F5.3,13H KG*S E2/M E4,3X,4HNU =,F10.8,7H SQ.M/S)
   IZT=NZPZ(2,0)
   WRITE(IDRU,40)IZT,(KBLT,TH(J),ALS(J),J=1,JT)
40 FORMAT (A1,5X,5(A1,4X,3HC =,F5.2,8H THETA =,F5.2))
41 V1 = SQRT(2.*GDF/DICHTE)
   DO 48 I=1,NAL
   IVS = -I
   DO 46 J = 1,JT
   IF(ALV(I)-ALS(J))42,44,42
42 VALF = V1/SQRT(.11*ABS(ALV(I)-ALS(J)))
   IF(VALF - VMAX)46,46,44
44 VALF = VMAX
46 RER(J) = VALF*TM(J)/ZAEH
48 CALL GRP(IVS,RER,MUR,JT,ISTIFT)
   MSPLI=0
49 IF(D)72,50,72
50 GO TO 11
C PLW CARD
60 IF(PUFF(1))62,68,62
62 DST = PUFF(1)*.01
   GST = PUFF(2)
   DGF = PUFF(3)
   CWSF = PUFF(4)*.001
   DO 66 J = 1,5
   JZ = 2*J+3

```

APPENDIX

```

        IF(PUFF(JZ))64,68,64
64  TST(J) = PUFF(JZ)
    BANT(J) = PUFF(JZ+1)
    MUR(J) = NUPU
66  JT = J
68  BF = 0.
    FST = 0.
    NF = 0
    CALL DIA(X,Y,NQ,D)
    IF(DST.LT.0.) DST = 0
    DO 70 J = 1,JT
    TH(J) = TST(J)*DST/D
    ALS(J) = 0.
    BF = BF + BANT(J)
70  FST = FST + BANT(J)*TST(J)
    FF = FST*DST/D
    GEW = GST + (FF-FST)*DGF
    GDF = GEW/FF
    GO TO 36
C  PLWA CARD
71  NAN = NUPU
    CWSFU=CWSF
    FAU = FF
    GAU = GEW
    DO 88 NF = 1,NAN
    FF = FF+PUFF(1)
    GEW = GEW+PUFF(2)
    CWSF=CWSF+.001*PUFF(3)
    V1 = SQRT(2.*GEW/(DICHTE*FF))
72  IZT=NZPZ(3,NAL+10)
    WRITE(IDRU,74)IZT,NAMP
74  FORMAT (A1,25HAIRCRAFT POLAR AIRFOIL ,12A1)
    CWS = CWSF/FF
    IZT=NZPZ(2,0)
    WRITE(IDRU,76)IZT
76  FORMAT (A1,39H B(M) S(SQ.M) S*(SQ.M) W(KG) W*(KG),
    *3X,3HT/C,3X,6H(T/C)*,2X,8HAP(SQ.M),2X,3HCDP)
    IZT=NZPZ(1,0)
    WRITE(IDRU,78)IZT,BF,FF,FST,GEW,GST,D,DST,CWSF,CWS
78  FORMAT (A1,F6.2,2F8.2,2F8.0,4F8.4)
    IZT=NZPZ(2,0)
    WRITE(IDRU,80)IZT
80  FORMAT (A1,54H ALPHA CL CDP CDT V(KM/H) VS(M/S) L
    */D)
    DO 84 I = 1,NAL
    CA = 0.
    CWP = 0.
    DO 82 J = 1,JT
    BATF=BANT(J)*TST(J)/FST

```

APPENDIX

```

CALL VISC(I,J,CANT,CWNT,CMDU)
CA=CA+CANT*BATF
82 CWP=CWP+CWNT*BATF
   CWGES = CWP + CWS + 1.03*CA*CA*FF/(PI*BF*BF)
   IF(ABS(CA).LT..01)CA=.01
   VKMH = 3.6*V1/SQRT(ABS(CA))
   VS = VKMH*CWGES/(3.6*ABS(CA))
   GLTZ = CA/CWGES
   IZT=NZPZ(1,0)
84 WRITE(IDRU,86) IZT,ALV(I),CA,CWP,CWGES,VKMH,VS,GLTZ
86 FORMAT (A1,F6.2,F8.3,2F8.4,F8.1,F9.3,F8.2)
   IF(NF.EQ.0) GO TO 50
88 CONTINUE
   CWSF = CWSFU
   FF=FAU
   GEW = GAU
   GO TO 50
C   STRK CARD
90 IF(NUPU)96,100,92
92 NT = 0
94 DO 98 J = 1,14
   IF(PUFF(J)) 96,100,96
96 NT = NT + 1
98 T(NT) = PUFF(J)*10.
   IF(IABS(NUPU).GT.14) GO TO 11
100 CALL STRDR(T,NT)
   IF(NUPI.NE.0) GO TO 11
   DO 102 I = 1,NT
102 CALL STRAAK(T(I),RUA,YBL,MXZ,ISTIFT)
   GO TO 11
C   DIAG CARD
104 CALL DIAGR(ISTIFT,NUPU,NUPI)
   GO TO 11
C   PUXY CARD
106 CALL PUDECK
   GO TO 11
C   STRD CARD
112 IF(NUPU.NE.0) MXZ = NUPU
   IF(PUFF(1).NE.0.) YBL = 100.*PUFF(1)
   IF(PUFF(2).NE.0.) RUA = 100.*PUFF(2)
   GO TO 11
C   ABSZ CARD
142 IF(NUPA.NE.0) AGAH(3)=FLOAT(NUPE)
   IF(PUFF(2).NE.0.) ABFA=PUFF(2)
   GO TO 11
C   ENDE CARD
150 IF(MGC.NE.0) CALL GCLOSE
   IF(MPL.NE.0)CALL FINISH
   STOP

```

APPENDIX

```

C   CDCL CARD
160 IF(NUPA.EQ.0)GO TO 166
    BL1=PUFF(1)+.005
    BL2=PUFF(2)+.005
    DO 162 K=1,5
        NLINE(K)=INT(BL1*(10.**K-3))-10*INT(BL1*(10.**K-4))
162 NPAR(K) =INT(BL2*(10.**K-3))-10*INT(BL2*(10.**K-4))
    DO 164 K=1,12
164 BROKL(K)=PUFF(K+2)
    GO TO 11
166 CALL CDCL(NUPU,JP,ISTIFT)
    GO TO 11
C   FXPR CARD
180 ITP=NUPU
    CALL FIXLES
    MSPLI=0
C   PAN CARD
170 IF(MSPLI.EQ.0)CALL SPLITZ(X,Y,NQ,XP)
    IF(NUPA.NE.0) AGAM(10)=FLOAT(NUPE)
    IF(NUPA.EQ.9) GO TO 11
    DO 172 I=1,14
        IF(PUFF(I).EQ.0.)GO TO 172
        MEIG=INT(PUFF(I))
        MEI=MEIG/10
        KEI =MEIG-10*MEI
        XSTX=ABS(PUFF(I)-FLOAT(MEIG))
        CALL PADD(X,Y,XP,NQ,MEI,KEI,XSTX)
172 CONTINUE
    DO 174 I=1,NQ
        XF(I)=X(I)
        YF(I)=Y(I)
174 BETAF(I)=XP(I)
        DLTR=DLT
        DLTUR=DLTU
        XDA=0.
        YDA=0.
        FLCH=0.
        DEFLG=0.
        NQRS=NQ
176 NKR=NQ-1
        CALL PANEL(NKR,AMAT,GAMMA,CAE)
        DARG = ALN
        GO TO 11
C   FLAP CARD
190 CHORD = XF(1)
    FLCH=PUFF(1)
    XDA=(1.-.01*FLCH)*CHORD
    YDA=.01*PUFF(2)*CHORD
    ARCL=.01*PUFF(3)*CHORD

```

APPENDIX

```

DEFLG=PUFF(4)
DLT=DLTR+DEFLG
DLTU=DLTUR-DEFLG
DEFL=DEFLG*BOGEN
ARCLU=.01*PUFF(5)*CHORD
MSPLI=1
CALL FLAP(XF,YF,BETAF,NQRS,XDA,YDA,ARCL,ARCLU,DEFL,X,Y,XP,NQ)
GO TO 176
C ALFA CARD
333 IF(NUPA.EQ.0) GO TO 330
MOMAG=NUPA
AGAM(2)=FLOAT(NUPE)
IF(NUPA.EQ.1) AGAM(10)=FLOAT(NUPE+1)
330 IF(NUPU.EQ.0) GO TO 335
DO 331 I=1,14
331 ALCA(I)=PUFF(I)
328 ITIT1=NUPI/2+1
ITIT2=NUPI-2*ITIT1+3
DARF=0.
IF(ITIT2.NE.1)DARF=1.
NAL=IABS(NUPU)
IF(NAL.GT.14)NAL=4
335 DO 334 I=1,NAL
PA=ALCA(I)
IF(PA.LE.-99.) PA=RS(30+I)
IF(PA.GT.-99.) PA=PA+DARF*DARG
334 ALV(I)=PA
CALL MOMENT(X,Y,NQ,XDA,YDA,DEFLG,MOMAG)
IF(AGAM(2).EQ.0.)GO TO 11
NZF=NQ+3
IF(ITP.EQ.2.AND.AGAM(10).EQ.1.)NZF=0
CALL DIA(X,Y,NQ,THK)
THKP=100.*THK
DO341 N=1,NQ
ND=N-1
XDR=X(N)
YDR=Y(N)
DO 340M=1,NAL
V(M)=ABS(VPR(N,M))
VQ=1.-V(M)*V(M)
IF(ITIT1.EQ.2)V(H)=VQ
340 CONTINUE
NZN=NZPZ(1,NZF)
IF(NZN.NE.NNESE.AND.N.NE.1)GO TO 341
NZF=0
DO 339 M=1,NAL
339 P(M)=ALV(M)-DARF*DARG
332 IF(ITP.EQ.1)WRITE(IDRU,337)NZN,NUPRO,THKP,(P(M),M=1,NAL)
337 FORMAT (A1,8HAIRFOIL ,I4,F8.2,1HZ,F9.2,13F8.2)

```

APPENDIX

```

IF(ITP.EQ.2)WRITE(IDRU,336)NZT,NAMP,THKP,FLCH,DEFLG,(P(M),M=1,NAL)
336 FORMAT (A1,8HAIRFOIL ,12A1,F8.2,11HZ THICKNESS,F10.2,6HZ FLAP,
*F8.2,19H DEGREES DEFLECTION/23X,14F8.2)
IF(ITP.EQ.2)NZT=NZPZ(1,0)
NZT=NZPZ(1,0)
WRITE(IDRU,338)NZT,CPV(ITIT1),(ALT(X,M,ITIT2),M=1,4)
338 FORMAT (A1,3H N,7X,1HX,8X,1HY,5X,A9,60HDISTRIBUTIONS FOR THE ABOVE
*E ANGLES OF ATTACK RELATIVE TO THE,4A4)
NZT=NZPZ(1,0)
341 WRITE(IDRU,342)ND,XDR ,YDR ,(V(M),M=1,NAL)
342 FORMAT (I4,F10.5,F9.5,14F8.3)
IF(ITP.EQ.2)GO TO 11
NZT=NZPZ(1,0)
WRITE(IDRU,344)NZT,DARG,CM,ETA
344 FORMAT (A1,8HALPHAO =,F5.2,8H DEGREES,3X,5HCMO =,F7.4,3X,
*SHETA =,F6.3)
GOTO11
END
SUBROUTINE PADD(X,Y,BETA,NQ,MEI,KEI,XST)
DIMENSION X(65),Y(65),BETA(65)
COMMON/PLTH/MPL,MGC,XZEH,YZEH,MSPLI
IF(MSPLI.EQ.0)CALL SPLITZ(X,Y,NQ,BETA)
M=MEI
K=KEI
IF(M.NE.0)GO TO 14
XV=X(1)
DO 10 I=2,NQ
XM=X(I)
DX=XM-XV
IF(K.GT.0.AND.DX.LT.0.)GO TO 10
IF((XM-XST)*DX.GE.0.)GO TO 12
10 XV=XM
I=NQ
12 M=I-1
K=1
14 X0=X(1)
XM=X(M)
XMP=X(M+1)
DX=XMP-XM
PH1=ATAN2(SQRT(XM*(X0-XM)),XM-.5*X0)
PH2=ATAN2(SQRT(XMP*(X0-XMP)),XMP-.5*X0)
DY=Y(M+1)-Y(M)
DLTA=ATAN2(DX,-DY)
GA=BETA(M)-DLTA
GAS=DLTA-BETA(M+1)
TG=SIN(GA)/COS(GA)
TGS=SIN(GAS)/COS(GAS)
KS=NQ
2 KR=KS+K

```

APPENDIX

```

X(KR)=X(KS)
Y(KR)=Y(KS)
BETA(KR)=BETA(KS)
KS=KS-1
IF(KS.GT.M)GO TO 2
FKP=FLOAT(K+1)
DO 4 KE=1,K
XI=FLOAT(KE)/FKP
IF(XM.GE.XST.OR.XMP.GE.XST)XI=(.5*XO*(1.+COS(PH1+XI*(PH2-PH1)))
1 -XM)/DX
IF(MEI.EQ.0)XI=(XST-XM)/(XMP-XM)
ETA=XI*(1.-XI)*(TG*(1.-XI)+TGS*XI)
ME=M+KE
X(ME)=X(M)+XI*DX-ETA*DY
Y(ME)=Y(M)+ETA*DX+XI*DY
4 BETA(ME)=ATAN(TG*(1.-XI)*(1.-3.*XI)+TGS*XI*(2.-3.*XI))+DLTA
NQ=NQ+K
RETURN
END
SUBROUTINE MOMENT(X,Y,NQ,XDA,YDA,DEFLG,MOMAG)
DIMENSION X(NQ),Y(NQ)
COMMON/EA/ILES,IDRU,ISTA,NNESE
COMMON/PRAL/DLT,DLTU,ALN,ALV(14),NAL,ITP,NAMP(12),CML(14),CRL(14)
1 ,CPV(2),ALT(4,2),DARF,ITIT1,ITIT2
IF(MOMAG.NE.1) GO TO 4
NZN=NZPZ(2,NAL+4)
WRITE(IDRU,1)NZN,NAMP,XDA,YDA
1 FORMAT (A1,8HAIRFOIL,12A1,3X,22HHINGE POINT AT X/C =,
*F6.4,3X,5HY/C =,F6.4)
NZN=NZPZ(1,0)
WRITE(IDRU,2)NZN,DEFLG
2 FORMAT (A1,6H ALPHA,6X,2HCH,7X,2HCH,4X,7HDELTA =,F6.2,8H DEGREES)
4 DO 31 M=1,NAL
XD=.25
YD=0.
DO 20 K=1,2
SWITCH=1.
DROV=-1.
DX=X(1)-XD
DY=Y(1)-YD
ROV=DX*DX+DY*DY
VOV=VPR(1,M)**2
CR=0.
DO 10 I=2,NQ
DX=X(I)-XD
DY=Y(I)-YD
RO=DX*DX+DY*DY
DRO=RO-ROV
VO=VPR(I,M)**2

```


APPENDIX

```

IF(XD.EQ.0..OR.K.EQ.1)GO TO 5
IF(DRQV.GE.0..OR.DRQ.LT.0.)GO TO 5
CR=CR+SWITCH*RQV*VQV
SWITCH=-SWITCH
5 IF(SWITCH.GT.0.)CR=CR+RQ*VQV-RQV*VQ
RQV=RQ
DRQV=DRQ
10 VQV=VQ
CR=.25*CR
IF(K.EQ.1)CM=CR
XD=XDA
20 YD=YDA
CML(M)=CM
CRL(M)=CR
IF(MOMAG.NE.1) GO TO 31
NZN=NZPZ(1,0)
30 WRITE(IDRU,32)NZN,ALV(M),CM,CR
31 CONTINUE
22 FORMAT (A1,F6.2,F9.4,F10.5)
RETURN
END
SUBROUTINE SPLITZ(X,Y,NQ,BETA)
COMMON/EA/ILES,IDRU,ISTA,NNESE
COMMON/PLTH/MGC,MPL,XZEH,YZEH,MSPLI
COMMON GAP1(1175),IGAP(6),GAP3(753),U(121),V(121),DS(121)
DIMENSION X(NQ),Y(NQ),BETA(NQ)
DATA DBSOLL/1.E-6/
MSPLI=1
NK=NQ-1
DO 10 N=1,NQ
XN=X(N)
YN=Y(N)
IF(N.EQ.1)GO TO 8
DX=XN-XV
DY=YN-YV
M=N-1
AN=SQRT(DX*DX+DY*DY)
DS(M)=AN
IF(M.EQ.1)GO TO 6
U(M)=DXV*DX+DYV*DY
V(M)=DXV*DY-DYV*DX
DD=-(AV/(AV+AN))*ATAN2(V(M),U(M))
BETA(M)=SIN(DD)/COS(DD)
6 AV=AN
DXV=DX
DYV=DY
8 XV=XN
10 YV=YN
BETA(NQ)=0.

```

APPENDIX

```

NIT=0
DBM=0.
12 NIT=NIT+1
DBM=0.
AV=DS(1)
GS=BETA(2)
GN=.5*GS
DO 20 N=2,NK
UN=U(N)
VN=V(N)
GP=(UN*GS+VN)/(VN*GS-UN)
AN=DS(N)
GPS=BETA(N+1)
IF(N.EQ.NK)GPS=.5*GP
GSQ=GS*GS
GPQ=GP*GP
E=AV*(2.*GP-GPS)*(1.+GPQ)-AN*(2.*GS-GN)*(1.+GSQ)
ES=AN+AV
IF(GSQ+GPQ.LT..09)GO TO 18
E=AV*(1.+3.*GPQ-GP*GPS)*(UN-GP*VN)/(UN-GS*VN)+AN*(1.+3.*GSQ
1 -GS*GN)
18 DE=.5*E/ES
IF(DBM.LT.ABS(DE))DBM=ABS(DE)
GS=GS+DE
BETA(N)=GS
AV=AN
GN=(UN*GS+VN)/(VN*GS-UN)
20 GS=GPS
IF(DBM.GT.DBSOLL.AND.NIT.LE.15)GO TO 12
DO 30 N=1,NQ
XN=X(N)
YN=Y(N)
IF(N.EQ.1)GO TO 28
DX=XN-XV
DY=YN-YV
DLT=ATAN2(DX,-DY)
IF(N.EQ.2)BETA(1)=DLT+ATAN(.5*BETA(2))
BETA(N)=DLT-ATAN(BETA(N))
28 XV=XN
30 YV=YN
BETA(NQ)=DLT-ATAN(.5*GN)
RETURN
END
SUBROUTINE FLAP(X,Y,BETA,NQ,XDA,YDA,ARCL,ARCLU,DEFL,XF,YF,BETAF,
* NQF)
DIMENSION X(NQ),Y(NQ),BETA(NQ),XF(NQ),YF(NQ),BETAF(NQ),NAB(2,2),
1 NPA(2)
DO 2 N=1,NQ
XF(N)=X(N)

```

APPENDIX

```

YF(N)=Y(N)
2 BETAF(N)=BETA(N)
NQF=NQ
NSUCH=NQ/2
DO 22 M=1,2
AQ=1.E10
DO 10 N=1,NSUCH
NP=N+(M-1)*NSUCH
DTX=XDA-X(NP)
DTY=YDA-Y(NP)
ANQ=DTX*DTX+DTY*DTY
IF(ANQ.GE.AQ)GO TO 10
NDP=NP
AQ=ANQ
DBT=ATAN2(DTY,DTX)-BETA(NP)
10 CONTINUE
IF(SIN(DBT).GE.0.)NDP=NDP-1
XVDP=X(NDP)
YVDP=Y(NDP)
DX=X(NDP+1)-XVDP
DY=Y(NDP+1)-YVDP
DLT=ATAN2(DX,-DY)
GA=BETA(NDP)-DLT
GAS=DLT-BETA(NDP+1)
TG=SIN(GA)/COS(GA)
TGS=SIN(GAS)/COS(GAS)
U=XDA-XVDP
V=YDA-YVDP
SDL=SIN(DLT)
CDL=COS(DLT)
DL=SQRT(DX*DX+DY*DY)
XIDA=(U*SDL-V*CDL)/DL
ETADA=(U*CDL+V*SDL)/DL
XIT=XIDA
12 XKP=1.-XIT
DEIT=ETADA-XIT*XKP*(XKP*TG+XIT*TGS)
XIN=XIDA+DEIT*(XKP*(1.-3.*XIT)*TG+XIT*(3.*XKP-1.)*TGS)
IF(ABS(XIN-XIT).LT..00001)GO TO 14
XIT=XIN
GO TO 12
14 AA=DL*SQRT((XIT-XIDA)*(XIT-XIDA)+DEIT*DEIT)
AL=ARCL+SIN(.5*DEFL)*SIGN(AA,DX*DEIT)
IF(ARCLU.GT..00001.AND.DX.GT.0.)AL=AL-ARCL+ARCLU
IF(AL.LT..0001*X(1))AL=.0001*X(1)
XST=XKP
DO 20 L=1,2
ND=3-2*L
ALR=-XST*DL-AL
NR=NDP+L-1

```

APPENDIX

```

16 NV=NR+ND
   DXS=X(NV)-X(NR)
   DLS=SQRT((Y(NV)-Y(NR))**2+DXS*DXS)
   IF(ALR.GE.0.)GO TO 18
   ALR=ALR+DLS
   NR=NR-ND
   DXLV=DXS/DLS
   GO TO 16
18 XBR=X(NV)+ALR*DXLV
   NOU=2*M-3
   CALL PADD(XF,YF,BETAF,NOF,0,NCU,XBR)
   NAB(M,L)=NV+2*M-1
20 XST=XIT
22 CONTINUE
   CDF=COS(DEFL)
   SDF=SIN(DEFL)
   M=1
   N=1
   DO 30 L=1,NOF
     XF(N)=XF(L)
     YF(N)=YF(L)
     BETAF(N)=BETAF(L)
     IF(L.EQ.NAB(M,1))NPA(M)=N
     IF(L.GT.NAB(1,1).AND.L.LT.NAB(2,2))GO TO 26
     BETAF(N)=BETAF(N)-DEFL
     DX=XF(N)-XDA
     DY=YF(N)-YDA
     XF(N)=XDA+DX*CDF+DY*SDF
     YF(N)=YDA-DX*SDF+DY*CDF
26 IF(L.EQ.NSUCH)M=2
   IF((L-NAB(M,1))*(L-NAB(M,2)).GE.0)N=N+1
30 CONTINUE
   NOF=N-1
   CALL PADD(XF,YF,BETAF,NOF,NPA(2),3,XF(1))
   CALL PADD(XF,YF,BETAF,NOF,NPA(1),3,XF(1))
   RETURN
   END
SUBROUTINE PANEL(NK,A,GAMMA,CAE)
COMMON P1(121),P(121),XP(121),YP(121),PUFF(14),AGAM(14),X(121),
1Y(121),DS(122),VF(121),ARG(121),ANI(28),ALFA(29),IZZ,KFU,NQ,NUPRO,
2JAB,JST,CH,ETA,ABFA,PI,BOGEN
COMMON/PRAL/DLT,DLTU,ALN,ALV(14),NAL,ITP,NAMP(12),CHL(14),CRL(14)
1,CPV(2),ALT(4,2),DARF,ITIT1,ITIT2
COMMON/EA/ILES,IDRU,ISTA,NNESE
DIMENSION GAMMA(NK,2),A(NK,NK),BETA(121),DELTA(120),G(121,2),
1CAE(2),ZZL(120),DA(4)
EQUIVALENCE (G(1,1),VF(1)),(BETA(1),XP(1)),(DELTA(1),P1(1)),
1(ZZL(1),P(1))
LOGICAL WAKE

```

APPENDIX

```

DATA FED/.00001/
MAUS=IDRU
ITP=2
NAGA=INT(AGAM(10))
DELTA(1)=.5*(BETA(1)+BETA(NQ))-PI
XHM=X(NQ)
YHM=Y(NQ)
DO 2 N=1,NQ
  XN=X(N)
  YN=Y(N)
  DX=XN-XHM
  DY=YN-YHM
  DL=SQRT(DX*DX+DY*DY)
  DS(N)=DL
  IF(N.NE.1)GO TO 22
  WAKE=DL.GT..0001*XN
  IF(.NOT.WAKE)GO TO 24
22 DELTA(N)=ATAN2(DX,-DY)
24 XHM=XN
  YP(N)=DELTA(N)
2  YHM=YN
  IF(DELTA(1).GT.0.)DELTA(1)=DELTA(1)-2.*PI
  EDGE1=BETA(1)-DELTA(1)
  EDGE2=BETA(NQ)-DELTA(1)-2.*PI
  COSAB=COS(EDGE1)
  SINAB=SIN(EDGE1)
  COSAQ=COS(EDGE2)
  SINAQ=SIN(EDGE2)
  ZZL(1)=PI*.5*(SINAB-SINAQ)+COSAB+COSAQ
  DO 20 M=1,NQ
  IF(.NOT.WAKE.AND.M.EQ.NQ)GO TO 20
  BEM=BETA(M)
  XMRE=X(M)
  YMRE=Y(M)
  IF(M.GT.1.AND.M.LT.NQ)GO TO300
  XMRE=.5*(X(1)+X(NQ))
  YMRE=.5*(Y(1)+Y(NQ))
  BEM=DELTA(1)+FLOAT(M/NK)*.5*PI
300 CBM=SIN(BEM)
  SBM=-COS(BEM)
  IF(M.LE.NK)A(M,1)=0.
  G(M,1)=-CBM*6.283185
  G(M,2)=-SBM*6.283185
  XNR=XN
  YNR=YN
  DO 9 NKOR=1,NQ
  XNP=X(NKOR)
  YNP=Y(NKOR)
  DAT=0.

```

APPENDIX

```

DAS=0.
DAY=0.
IF(.NOT.WAKE.AND.(NKOR.EQ.1.OR.NKOR.EQ.M+NK))GC TO 8
IF(NKOR.EQ.1.AND.M.EQ.NQ)GO TO 8
N=NKOR-1
IF(N.EQ.0)N=1
DU=XNP-XNR
DV=YNP-YNR
DL=DS(NKOR)
CP=DU/DL
SP=DV/DL
GA=BETA(N)-DELTA(NKOR)
GAS=DELTA(NKOR)-BETA(NKOR)
XAA=XNR
YAA=YNR
DLR=DL
IF(M.NE.NKOR.AND.M.NE.NKOR-1)GO TO 200
IF(M.NE.1.AND.M.NE.NQ)GO TO 4
IF(.NOT.WAKE)GO TO 6
IF(NKOR.EQ.1)GO TO 7
200 DX=XMRE-XNR
DY=YMRE-YNR
U=(DX*CP+DY*SP)/DL
V=(DY*CP-DX*SP)/DL
PK=1.
PKL=1.
IF(NKOR.EQ.1)GO TO 204
EMAX=ABS(GA+GAS)/8.
RAB=ABS(V)
IF(U.LT.0..OR.U.GT.1.)RAB=AMAX1(RAB,ABS(U),ABS(U-1.))
FAB=EMAX/(FED*RAB)
IF(FAB.LT.1.)GO TO 204
PK =FLOAT(INT(FAB*.333333)+1)
TG=SIN(GA)/COS(GA)
TGS=SIN(GAS)/COS(GAS)
202 XI=PKL/PK
ETA=XI*(1.-XI)*(TG*(1.-XI)+TGS*XI)
XZ=XAA+XI*DU-ETA*DV
YZ=YAA+ETA*DU+XI*DV
DUZ=XZ-XNR
DVZ=YZ-YNR
DL =SQRT(DUZ*DUZ+DVZ*DVZ)
CP=DUZ/DL
SP=DVZ/DL
DX=XMRE-XNR
DY=YMRE-YNR
U=(DX*CP+DY*SP)/DL
V=(DY*CP-DX*SP)/DL
204 VQ=V*V

```

APPENDIX

```

UQ=U*U
DEN1=U-.333333
DEN = DEN1*DEN1+VQ
IF(DEN.LT.10000.)GO TO 205
WU=.5*V/DEN
WV=-.5*DEN1/DEN
DEN1=U-.6666667
DEN=DEN1*DEN1+VQ
WUS=.5*V/DEN
WVS=-.5*DEN1/DEN
DWU=0.
DWV=0.
GO TO 206
205 ULM=U-1.
TAD=ATAN2(V,U*ULM+VQ)
TAL=.5*ALOG((VQ+UQ)/(VQ+ULM*ULM))
WU=(1.-U)*TAD+V*TAL
WV=V*TAD-(1.-U)*TAL-1.
WUS=U*TAD-V*TAL
WVS=1.-U*TAL-V*TAD
DWU=(U-UQ+VQ)*TAD+V*(2.*U-1.)*TAL-V
DWV=(UQ-VQ-U)*TAL+(2.*U-1.)*(V*TAD-.5)
206 FNMX=WU*CP-WV*SP
FNMV=WV*CP+WU*SP
FNMXS=WUS*CP-WVS*SP
FNMYS=WVS*CP+WUS*SP
DFX=DWU*CP-DWV*SP
DFY=DWV*CP+DWU*SP
DAX=(FNMX*CBM+FNMV*SBM)/PK
DAZ=(FNMXS*CBM+FNMYS*SBM)/PK
DAU=DFX*CBM+DFY*SBM
IF(NKOR.EQ.1)DAX=DAX*COSAB+(FNMX*SBM-FNMV*CBM)*SINAB
IF(NKOR.EQ.1)DAZ=-DAZ*COSA0-(FNMXS*SBM-FNMYS*CBM)*SINA0
DAT=DAT+DAX*(PK+1.-PKL)+DAZ*(PK-PKL)
DAS=DAS+DAX*(PKL-1.)+DAZ*PKL
DAY=DAY+(DAU/PK+DAX*(PKL-1.)*(PK+1.-PKL)+DAZ*PKL*(PK-PKL))/PK
IF(PKL.GT.PK-.01)GO TO 8
PKL=PKL+1.
XNR=XZ
YNR=YZ
GO TO 202
4 FLOM=FLOAT(NKOR-M)
DAS=.5*GAS+(FLOM-.3333333)*(GA-GAS)
DAT=.5*GA+(FLOM-.6666667)*(GA-GAS)+FLOM*PI
DAY=(GA+GAS)/12.
GO TO 8
6 DAT=ALOG(DS(NQ)/DL)
DAS=-1.
GO TO 8

```

APPENDIX

```

7 DAS=PI*.5*(COSAB-COSAQ)-SINAB-SINAQ
8 DO 100 L=1,4
100 DA(L)=0.
   IF(NKOR.EQ.1)GO TO 106
   IF(NKOR.GE.3)FM=DS(NKOR-1)/DLR
   IF(NKOR.LE.NK)FP=DS(NKOR+1)/DLR
   IF(NKOR.EQ.2)GO TO 102
   IF(NKOR.EQ.NQ)GO TO 104
   DAH=DAY/2.
   DA(1)=-DAH/(FM*(FM+1.))
   DA(2)=DAH*(1./FM-1./((FM+1.)) + DAT
   DA(3)=DAH*(1./FP-1./((FM+1.)) + DAS
   DA(4)=-DAH/(FP*(FP+1.))
   GO TO 108
102 DA(2)=DAT-DAY/(FP+1.)
   DA(3)=DAS+DAY/FP
   DA(4)=-DAY/(FP*(FP+1.))
   GO TO 108
104 DA(1)=-DAY/(FM*(FM+1.))
   DA(2)=DAT+DAY/FM
   DA(3)=DAS-DAY/(FM+1.)
   GO TO 108
106 DA(3)=DAS+DAT
108 DO 116 L=1,4
   NE=NKOR+L-3
   IF(NE.GT.NQ.OR.NE.LT.1)GO TO 116
   IF(NE.LT.NQ)GO TO 110
   IF(M.LT.NQ)A(M,1)=A(M,1)-DA(L)
   IF(M.EQ.NQ)ZZL(1)=ZZL(1)-DA(L)
   GO TO 116
110 IF(M.EQ.NQ)GO TO 112
   IF(L.EQ.4)A(M,NE)=DA(L)
   IF(L.LT.4)A(M,NE)=A(M,NE)+DA(L)
   GO TO 116
112 IF(L.EQ.4)ZZL(NE)=DA(L)
   IF(L.LT.4)ZZL(NE)=ZZL(NE)+DA(L)
116 CONTINUE
   10 XNR=XNP
   9 YNR=YNP
   20 CONTINUE
   IF(.NOT.WAKE)A(1,NK)=A(1,NK)-1.
   IF(WAKE)GO TO 139
   DD 136 N=1,NK
136 ZZL(N)=0.
   ZZL(1)=1.
   W=(1.+DS(3)/DS(2))**.66667
   ZZL(3)=.5/(W-1.)
   ZZL(2)=-W*ZZL(3)
   W=(1.+DS(NK)/DS(NQ))**.66667

```


APPENDIX

```

      ZZL(NK-1)=-.5/(W-1.)
      ZZL(NK)=-W*ZZL(NK-1)
139 DO 144 N=1,NK
      DO 141 M=1,2
      GR=0.
      DO 140 L=1,NK
140 GR=GR+G(L,M)*A(L,N)
      IF(WAKE)GR=GR+G(NQ,M)*ZZL(N)
141 GAMMA(N,M)=GR
      DO 143 M=N,NK
      BW=0.
      DO 142 L=1,NK
142 BW=BW+A(L,N)*A(L,M)
      BW=BW+ZZL(M)*ZZL(N)
143 DELTA(M)=BW
      DO 144 M=N,NK
144 A(M,N)=DELTA(M)
      DO 146 N=1,NK
      DO 146 M=N,NK
146 A(N,M)=A(M,N)
      CALL GAUSS(A,GAMMA,NK,2,0.)
      DO 152 M=1,2
      CA=0.
      DO 150K=2,NK
150 CA=CA+(DS(K)+DS(K+1))*GAMMA(K,M)
152 CAE(M)=CA/X(1)
      ALN=ATAN2(CAE(1),CAE(2))/BOGEN
      IF(NAGA.EQ.0)GO TO 48
      NZT=NZPZ(2,NAL+3)
      IF(AGAM(2).NE.0.)NZT=NZPZ(3,0)
      WRITE(HAUS,42)NZT,NAHP,CAE,ALN
42 FORMAT (A1,23HPANEL METHOD AIRFOIL ,12A1,3X,4HCA =,F8.5,1H,,
      *F8.5,3X,8HALPHAO =,F5.2,8H DEGREES)
      IF(NAGA.LT.3)GO TO 48
      NZT=NZPZ(1,0)
      WRITE(HAUS,43) NZT
43 FORMAT (A1,3H N,7X,1HX,9X,1HY,8X,4HBETA,6X,6HGAMMA1,5X,6HGAMMA2,
      *5X,4HTAU1,6X,4HTAU2,6X,2HDS)
      DO 44 K=1,NK
      NZT=NZPZ(1,0)
      KD=K-1
      GA=BETA(K)-YP(K+1)
      GAS=YP(K)-BETA(K)
44 WRITE(HAUS,46)NZT,KD,X(K),Y(K),BETA(K),GAMMA(K,1),GAMMA(K,2),GA,
      1 GAS ,DS(K+1)
46 FORMAT (A1,I3,2F10.5,3F11.5,2F10.5,F9.5)
      NZT=NZPZ(1,0)
      WRITE(HAUS,46)NZT,NK,X(NQ),Y(NQ),BETA(NQ),GAMMA(1,1),GAMMA(1,2)
48 DO 50 K=1,NK

```

APPENDIX

```

VF(K)=GAMMA(K,1)
50 ARG(K)=GAMMA(K,2)
VF(NQ)=-GAMMA(1,1)
ARG(NQ)=-GAMMA(1,2)
RETURN
END
SUBROUTINE WANDEL(N,NF,M,MHB)
DIMENSION NF(M),NVGL(11)
DATA NVGL/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H /
NW=N
DO 1 K=1,M
MR=M+1-K
L=11
IF(NW.EQ.0.OR.K.LE.MHB)GO TO 1
NX=NW/10
L=NW-10*NX+1
NW=NX
1 NF(MR)=NVGL(L)
RETURN
END
FUNCTION VPR(N,I)
COMMON GAP(876),VF(121),ARG(121),GAPS(57),IZZ,KFU,NQ
COMMON/PRAL/DLT,DLTU,ALN,ALV(14),NAL,ITP,NAMP(12),CML(14),CRL(14)
1 ,CPV(2),ALTX(4,2),DARF,ITIT1,ITIT2
DATA ALR,ANR/99.,99./
IF(ITP-1)2,2,4
2 VPR=VF(N)*COSG(180.*FLOAT(N-1)/FLOAT(NQ-1)-ALV(I))
GO TO 8
4 IF(ALV(I).EQ.ALR.AND.ALN.EQ.ANR)GO TO 6
ALR=ALV(I)
CAST=COSG(ALR-ALN)
SAST=SING(ALR-ALN)
ANR=ALN
6 VPR=CAST*VF(N)+SAST*ARG(N)
8 RETURN
END
SUBROUTINE FIXLES
DIMENSION NAM(8)
COMMON GAP(512),X(121),Y(121),GAPS(421),IDU,KDU,NQ,IGA(3),GAPT(4),
1BOGEN
COMMON/EA/ILES,IDRU,ISTA,NNESE
COMMON/PRAL/DLT,DLTU,ALN,ALV(14),NAL,ITP,NAMP(12),CML(14),CRL(14)
1 ,CPV(2),ALTX(4,2),DARF,ITIT1,ITIT2
READ (ILES,2) NAMP
2 FORMAT (12A1)
READ (ILES,3) MUP,MLOW
3 FORMAT (2I5)
NQ=MUP+MLOW-1
CALL RDC (X,MUP,-1,1)

```

APPENDIX

```

CALL RDC (Y,MUP,-1,1)
CALL RDC (X,MUP,1,NQ)
CALL RDC (Y,MUP,1,NQ)
NT=NQ/12
DLT=Y(NT)/(BOGEN*(1.-X(NT)))
NT=NQ-NT+1
DLTU=-Y(NT)/(BOGEN*(1.-X(NT)))
RETURN
END
SUBROUTINE RDC (ARR,NB,ND,NE)
DIMENSION ARR(121),BF(8)
COMMON /EA/ILES,IDRU,ISTA,NNESE
N=NB
5 READ (ILES,10) BF
10 FORMAT (8F10.5)
DO 15 M=1,8
ARR(N)=BF(M)
IF (N.EQ.NE) RETURN
15 N=N+ND
GO TO 5
END
SUBROUTINE DEFINE(PL,Q,H,XU,XC,YU,YO)
COMMON/PLTM/MPL,MGC,XZEH,YZEH,MSPLI
COMMON /GRMIN/XMIN,YMIN,XSCALE,YSCALE
LOGICAL PL
IF (MPL.EQ.1) GO TO 1
CALL PSEUDO
CALL LEROY
MPL=1
1 XMIN=XU
YMIN=YU
CALL CALPLT (0.,0.,-3)
IF(MGC.NE.0)CALL GCLOSE
MGC=1
XZEH=Q/(XC-XU)
YZEH=H/(YO-YU)
XSCALE=XZEH/25.4
YSCALE=YZEH/25.4
RETURN
END
SUBROUTINE GCLOSE
COMMON/PLTM/MPL,MGC,XZEH,YZEH,MSPLI
CALL NFRAME
MGC=0
RETURN
END
SUBROUTINE FINISH
CALL CALPLT (0.,0.,999)
RETURN

```

APPENDIX

```

END
SUBROUTINE SMOOTH(IS,X,Y,BT,NQ,A)
COMMON GAP1(1175),IGAP(6),GAP3(753),XPL(121),YPL(121)
COMMON/PLTM/MPL,MGC,XZEH,YZEH,MSPLI
COMMON/EA/ILES,IDRU,ISTA,NNESE
DIMENSION X(NQ),Y(NQ),BT(NQ)
LOGICAL A
EXTERNAL A
DATA HS/.4/
IF(MSPLI.EQ.0)CALL SPLITZ(X,Y,NQ,BT)
XV=X(1)
YV=Y(1)
BTV=BT(1)
I=1
XPL(I)=XV
YPL(I)=YV
DO 10 N=2,NQ
XR=X(N)
YR=Y(N)
BTR=BT(N)
DX=XR-XV
DY=YR-YV
DL=SQRT(DX*DX+DY*DY)
DLT=ATAN2(DX,-DY)
GA=SIN(BTV-DLT)/COS(BTV-DLT)
GAS=SIN(DLT-BTR)/COS(DLT-BTR)
IF(ABS(GA).LT..4.AND.ABS(GAS).LT..4)GO TO 4
MSM=N-1
NZZ=NZPZ(2,0)
WRITE(IDRU,2)NZZ,GA,GAS,MSM,N
2  FORMAT (A1,38HWARNING - SUBROUTINE SMOOTH HAS SLOPES,F6.3,4H AND,
*F6.3,15H BETWEEN POINTS,I3,4H AND,I3)
4  ESS =ABS(GA-2.*GAS)
   ES =ABS(GAS-2.*GA)
   IF(ES.GT.ESS)ESS=ES
   NAB=INT(SQRT(DL*XZEH*ESS/HS))+1
   FNAB=FLOAT(NAB)
   DO 6 K=1,NAB
   U=FLOAT(K)/FNAB
   V=U*(1.-U)*(GA*(1.-U)+GAS*U)
   I=I+1
   XPL(I)=XV+U*DX-V*DY
   YPL(I)=YV+U*DY+V*DX
   IF(I.LT.121)GO TO 6
   CALL POLZUG(IS,XPL,YPL,121,.TRUE.,A)
   I=1
   XPL(I)=XPL(121)
   YPL(I)=YPL(121)
6  CONTINUE

```

APPENDIX

```

XV=XR
YV=YR
10 BTV=BTR
IF(I.GT.1)CALL POLZUG(IS,XPL,YPL,I,.TRUE.,A)
RETURN
END
SUBROUTINE BPLOTT(X,Y,N,M)
COMMON/PLTM/MPL,MGC,XZEH,YZEH,MSPLI
COMMON/LINING/BROK(12),NLINE(5),NPAR(5),J
DIMENSION BF(8)
DATA NLINE/2*1,3*3/,NPAR/0,2,4,6,8/,BROK/5.,1.5,10.,9*1.5/
IF(M.EQ.2)GO TO 20
IF(M*ND.GE.2)GO TO 4
IF(M.EQ.0.OR.N.EQ.1)CALL TPLOT(X,Y,N,0)
IF(ND.LE.1)GO TO 10
K=1
SPL=BF(1)*.5
GO TO 10
4 DX=XZEH*(X-XV)
DY=YZEH*(Y-YV)
DS=SQRT(DX*DX+DY*DY)
6 SPT=SPL
IF(SPL.GT.DS)SPT=DS
XE=XV+SPT*DX/(DS*XZEH)
YE=YV+SPT*DY/(DS*YZEH)
NPEN=(K-2*(K/2))*N
IF(NPEN.NE.0.OR.SPL.LE.DS)CALL TPLOT(XE,YE,NPEN,0)
IF(SPT.EQ.DS)GO TO 8
K=K+1
IF(K.GT.ND)K=1
SPL=SPL+BF(K)
GO TO 6
8 SPL=SPL-DS
10 XV=X
YV=Y
RETURN
20 ND=NPAR(J)
IF(ND.LT.2)RETURN
NAN=NLINE(J)
NEN=NAN+ND-1
SB=0.
DO 22 K=NAN,NEN
22 SB=SB+BROK(K)
AB=AINT(X/SB)+1.
DO 24 K=1,ND
KS=K+NAN-1
24 BF(K)=X*BROK(KS)/(AB*SB)
RETURN
END

```

APPENDIX

```

SUBROUTINE POLZUG(IST,R,S,J,ALG,B)
COMMON/PLTH/MPL,MGC,XZEH,YZEH,MSPLI
DIMENSION R(J),S(J)
LOGICAL B,ALG
DATA X,Y/O.,O./
IF(ALG)GO TO 4
PTL=0.
DO 2 K=2,J
DX=XZEH*(R(K)-R(K-1))
DY=YZEH*(S(K)-S(K-1))
2 PTL=PTL+SQRT(DX*DX+DY*DY)
CALL BPLLOT(PTL,O.,O,2)
4 MAN=0
NP=1
NPD=1
NEND=J
IF(XZEH*ABS(R(1)-X)+YZEH*ABS(S(1)-Y).LE.XZEH*ABS(R(J)-X)+YZEH*ABS
1 (S(J)-Y))GO TO 10
NP=J
NPD=-1
NEND=1
10 X=R(NP)
Y=S(NP)
NPEN=0
IF(B(X,Y))NPEN=1
IF(MAN.NE.O)GO TO 12
MAN=1
IF(ALG)CALL TPLLOT(X,Y,O,O)
IF(.NOT.ALG)CALL BPLLOT(X,Y,O,O)
GO TO 19
12 IF(NPEV.EQ.NPEN)GO TO 18
X1=X
Y1=Y
DO 16 K=1,7
XZ=.5*(X1+XV)
YZ=.5*(Y1+YV)
NZ=0
IF(B(XZ,YZ))NZ=1
IF(NZ.NE.NPEV)GO TO 14
XV=XZ
YV=YZ
GO TO 16
14 X1=XZ
Y1=YZ
16 CONTINUE
IF(ALG)CALL TPLLOT(XZ,YZ,NPEV,O)
IF(.NOT.ALG)CALL BPLLOT(XZ,YZ,NPEV,1)
18 IF(NPEN.EQ.O)GO TO 19
IF(ALG)CALL TPLLOT(X,Y,NPEN,O)

```

APPENDIX

```

IF(.NOT.ALG)CALL BPLOT(X,Y,NPEN,1)
19 IF(NP.EQ.NEND)RETURN
   XV=X
   YV=Y
   NPEV=NPEN
   NP=NP+NPD
   GO TO 10
   END
   SUBROUTINE TPLOT(X,Y,NPEN,NORIG)
   COMMON /GRMIN/XMIN,YMIN,XSCALE,YSCALE
   A=(X-XMIN)*XSCALE
   B=(Y-YMIN)*YSCALE
   IPEN=3-NPEN
   CALL CALPLT (A,B,IPEN)
   RETURN
   END
   LOGICAL FUNCTION TRUE(X,Y)
   TRUE = .TRUE.
   RETURN
   END
   SUBROUTINE ACHS(M,UA,U1,U2,SA,SE,DS,DS2,A)
   DIMENSION X(3),Y(3)
   LOGICAL A
   EXTERNAL A
   MXY=2*(M/2)-M
   DS3=DS2
   DS1=DS
   Y(1)=UA
   Y(2)=UA
   XS=DS1*FLOAT(INT(SA/DS1))
   IF(SA.LT.0.)XS=XS-DS1
   X(1)=XS
   IF(M.LE.2) GO TO 1
   FKT=1./ALOG(10.)
   X(1)=FKT*ALOG(X(1))
   DS3=1.
1  XS=XS+DS1
   X(2)=XS
   IF(M.GT.2)X(2)=FKT*ALOG(XS)
   X(3)=X(2)
   Y(3)=U1
   IF(ABS(DS3*FLOAT(INT(X(3)*1.0001/DS3))-X(3)).GT..0001)GO TO 2
   Y(3)=U2
   IF(M.GT.2)DS1=10.*DS1
2  IF(MXY.EQ.0)CALL POLZUG(1,Y,X,3,.TRUE.,A)
   IF(MXY.NE.0)CALL POLZUG(1,X,Y,3,.TRUE.,A)
   IF(XS.GE.SE-.0001)RETURN
   X(1)=X(2)
   GO TO 1

```

APPENDIX

```

END
FUNCTION NZPZ(LL,M)
C  FUNCTION - LINE COUNTING
  DIMENSION IT(4)
  DATA IT(1),IT(2),IT(3),IT(4)/1H+,1H ,1H0,1H1/,NZEIZA/100/
  L=LL
  IF(L.GE.3) GO TO 1
  NZEIZA=NZEIZA+L
  IF (M+NZEIZA.GT.46) GO TO 1
  GO TO 2
1  L=3
  NZEIZA=1
2  NZPZ=IT(L+1)
  RETURN
END
SUBROUTINE GAUSS(A,B,N,M,EPS)
  DIMENSION A(N,N),B(N,M)
  COMMON/EA/ILES,IDRU,ISTA,NNESE
  EPS = ABS(EPS)
  DO 3 J=1,N
  H = .0
  DO 4 I=J,M
  Z = ABS(A(I,J))
  IF(Z.LT.H) GOTO 4
  K = I
  H = Z
4  CONTINUE
  IF(H.GT.EPS) GOTO 5
6  FORMAT(30H SINGULARITY ALARM IN SUBROUTINE GAUSS)
  WRITE(IDRU,6)
  RETURN
5  IF(K.EQ.J) GOTO 7
  DO 8 I=J,N
  H = A(J,I)
  A(J,I) = A(K,I)
8  A(K,I) = H
  DO 1 I=1,M
  H = B(J,I)
  B(J,I) = B(K,I)
1  B(K,I) = H
7  DO 9 I=1,N
  IF(I.EQ.J) GOTO 9
  H = A(I,J)/A(J,J)
  DO 10 K=J,N
10 A(I,K) = A(I,K)-A(J,K)*H
  DO 2 K=1,M
2  B(I,K) = B(I,K)-H*B(J,K)
9  CONTINUE
  H = 1./A(J,J)

```


APPENDIX

```

DO 11 I=J,N
11 A(J,I) = A(J,I)*H
DO 12 I=1,M
12 B(J,I) = B(J,I)*H
3 CONTINUE
RETURN
END
SUBROUTINE CDCL(MCDCL,JR,ISTIFT)
DIMENSION XMK(2)
COMMON CW(5,2,14),SA(5,2,14),SU(5,2,14),CLP(14),CDP(14),PG(36),
*PUFF(14),GAPS(677),IGAP(6),GAP(382),PGI(70),YMK(2)
COMMON/PRAL/DLT,DLTU,ALN,ALV(14),NAL,ITP,NAMP(12),CML(14),CRL(14)
1 ,CPV(2),ALTX(4,2),DARF,ITIT1,ITIT2
COMMON /LINING/BROK(12),NLINE(5),NPAR(5),J
LOGICAL PL,TRUE
EXTERNAL TRUE
KA=2
IF(MCDCL.GE.2)GO TO 2
ALMIN=0.
PL=ISTIFT.GE.5
ALMAX=0.
CMMIN=0.
CMMAX=0.
CLMAX=0.
CLMIN=0.
CDMIN=1.
CDLIM=1.
IF(PUFF(4).NE.0.)CDLIM=.01*PUFF(4)
KA=1
2 DO 14 K=KA,2
DO 12 J=1,JR
IPL=0
DO 10 I=1,NAL
SA1=SA(J,1,I)
SA2=SA(J,2,I)
IF(SA1+SA2.GT..65) GO TO 10
CALL VISC(I,J,CL,CD,CMV)
IF(CD.GT.CDLIM)GO TO 10
ALI=ALV(I)-ALN
IF(K.GT.KA)GO TO 8
CLMAX=AMAX1(CLMAX,CL)
CLMIN=AMIN1(CLMIN,CL,-4.*CMV)
IF(ALMAX.LT.ALI)ALMAX=ALI
IF(ALMIN.GT.ALI)ALMIN=ALI
IF(CMMIN.GT.CMV)CMMIN=CMV
IF(CMMAX.LT.CMV)CMMAX=CMV
CDMIN=AMIN1(CDMIN,CD)
IF(K.EQ.1)GO TO 10
8 IPL=IPL+1

```

APPENDIX

```

CLP(IPL)=CL
CDP(IPL)=CD
PG(IPL)=-4.*CMV
PGI(IPL)=CDAX+.0005*ALI
PGI(IPL+14)=CDAX-.01*SA1+.01
PGI(IPL+28)=CDAX-.01*SA2+.01
PGI(IPL+42)=CDAX-.01*SU(J,1,I)+.01
PGI(IPL+56)=CDAX-.01*SU(J,2,I)+.01
10 CONTINUE
IF(K.LT.2)GO TO 12
CALL POLZUG(ISTIFT,CDP,CLP,IPL,.FALSE.,TRUE)
DO 11 KC=1,5
LOC=1+14*(KC-1)
11 CALL POLZUG(ISTIFT,PGI(LOC),CLP,IPL,.FALSE.,TRUE)
CALL POLZUG(ISTIFT,PGI,PG,IPL,.FALSE.,TRUE)
12 CONTINUE
IF(K.GE.2)GO TO 14
CLMI=AMINI(PUFF(1),CLMIN-.3)
CDMI=0.
IF(CDMIN.GT..01)CDMI=RUND(CDMIN,200.)-.005
CDAX=CDMI+.025
CDAN=CDMI-.003
CLMA=AMAX1(2.4,PUFF(2),CLMAX+.3)
CDQ=CDAX+.013
SCF=1.
QUER=10000.*SCF*(CDQ-CDAN)
HOCH=100.*SCF*(CLMA-CLMI)
CALL DEFINE(PL,QUER,HOCH,CDAN,CDQ,CLMI,CLMA)
14 CONTINUE
IF(MCDCL.EQ.1.OR.MCDCL.GE.3)RETURN
CLMAX=RUND(CLMAX+.06,10.)
CDQ=CDQ-.003
CT1=CDMI-.0002
CT2=CDMI-.0005
CALL ACHS(2,CDMI,CT1,CT2,CLMIN,CLMAX,.1,.5,TRUE)
XMK(1)=CDMI+.0005
XMK(2)=CDMI+.004
DO 13 J=1,JR
YMK(1)=CLMAX-.10*FLOAT(J)
YMK(2)=YMK(1)
13 CALL POLZUG(ISTIFT,XMK,YMK,2,.FALSE.,TRUE)
CALL ACHS(1,0.,-.02,-.05,CDMI,CDQ,.001,.005,TRUE)
CT1=CDAX+.0002
CT2=CDAX+.0005
CALL ACHS(2,CDAX,CT1,CT2,CLMIN,CLMAX,.1,.5,TRUE)
CT1=CDAX-.0002
CT2=CDAX-.0005
CT3=-4.*CMIN
CT4=-4.*CMAX

```

APPENDIX

```

CALL ACHS(2,CDAX,CT1,CT2,CT4,CT3,.2,.4,TRUE)
CT1=CDAX+.0005*(ALMIN-2.)
CDQ=CDAX+.0005*(ALMAX+1.)
CALL ACHS(1,0.,.02,.05,CT1,CDQ,.0005,.0025,TRUE)
CALL GCLOSE
RETURN
END
SUBROUTINE VISC(I,J,CL,CD,CM)
COMMON CW(5,2,14),SA(5,2,14),SU(5,2,14)
COMMON/PRAL/DLT,DLTU,ALN,ALV(14),NAL,ITP,NAMP(12),CML(14),CRL(14)
1,CPV(2),ALT(4,2),DARF,ITIT1,ITIT2
ALRC=ALV(I)
D1=AMAX1(ALRC-ALN+DLT,0.)
D2=AMIN1(ALRC-ALN-DLTU,0.)
SA1=SA(J,1,I)
SA2=SA(J,2,I)
DCL=.5*(D1*SA1+D2*SA2)
CL=.11*(ALRC-DCL)
CD=ABS(CW(J,1,I))+ABS(CW(J,2,I))
CM=CML(I)+.0275*DCL*(ABS(1.-SA1-SA2)**1.5)
RETURN
END
SUBROUTINE GRP(NAX,RE,MU,JR,ISTIFT)
COMMON CW(5,2,14),SA(5,2,14),SU(5,2,14),HP(32),RP(32),PUFF(14),
*AGAM(14),X(121),Y(121),DS(122),GAP(299),IDU,KDU,NQ,LDU(3),
*GAPS(383),WRE(5),H(5),D2(5),CAD(5),SAR(5),SUR(5),CWD(5),DD(5),
*CHV(5),CPTR(5),BBL(5),CD(5),LD(5),LR(5)
COMMON/BL/HVGL,D2I,UR,UK1,DSZ,WR,MT,MA,V,V1,HR,D2R,XA,XU,DCQ,USTR,
*VSTR
COMMON/PRAL/DLT,DLTU,ALN,ALV(14),NAL,ITP,NAMP(12),CML(14),CRL(14)
1,CPV(2),ALT(4,2),DARF,ITIT1,ITIT2
COMMON/EA/ILES,IDRU,ISTA,NNESE
COMMON/TRIT/DLV,SUMP,XTRI(4),NU,ND
DIMENSION RE(5),MU(5),SFTX(2),D2TX(3)
LOGICAL TRUE,PL
EXTERNAL TRUE
DATA RETX,STXT/3HR =,2H S/,SFTX/5HUPPER,5HLOWER/
DATA D2TX/7H DELTA2,7H DELTA2,7H DELTA1/,HTX/3HH32/
DATA KBL,KST/1H ,1H*/ ,HBL,8BLI/-1.6,.03/
MA=0
V=0.
V1=0.
KOTX=INT(AGAM(6))-1
MAG=INT(AGAM(8))
NKR=NQ-1
HABLC = 2.5
DO 2 J=1,JR
H(J)=1.
LR(J)=INT(RE(J)*1.E-3)

```

APPENDIX

```

2 WRE(J)= SORT(RE(J))
  NALA = I
  NALE = IABS(NAX)
  IF(NAX.LT.0) NALA = NALE
  DO 38 IA = NALA, NALE
  IF(MAG.EQ.0.OR.IA.NE.1)GO TO 3
  PL=ISTIFT.GE.5
  CALL DEFINE(PL,315.,200.,1.43,1.89,.4,4.4)
  CALL ACHS(4,1.46,1.457,1.62,10.,10000.,10.,1.,TRUE)
  CALL ACHS(1,1.,.96,.92,1.46,1.62,.01,.05,TRUE)
  CALL ACHS(4,1.66,1.657,1.82,10.,10000.,10.,1.,TRUE)
  CALL ACHS(1,1.,.96,.92,1.66,1.82,.01,.05,TRUE)
  FKT=1./ALOG(10.)
  HP(1)=1.51509
  HP(2)=1.51509
  HP(3)=1.62
  RP(1)=1.
  RP(2)=2.6656
  RP(3)=3.5039
  CALL POLZUG(1,HP,RP,3,.TRUE.,TRUE)
  DO 5 IP=1,3
5 HP(IP)=HP(IP)+.2
  CALL POLZUG(1,HP,RP,3,.TRUE.,TRUE)
3 DO 38 JU =1,2
  IP=0
  ND = 2*JU -3
  NDSD=JU-ITP
  NEND=1+NKR*(JU-1)
  DO 6 K=1,NKR
  IF(VPR(K,IA).LE.0.)GO TO 7
6 CONTINUE
7 NU=K+JU-2
  IF(ABS(VPR(NU,IA)).LT..1)NU=NU+ND
  NUNSD = NU - NDSD
  DSR=DS(NUNSD)*VPR(NU,IA)/(VPR(NU,IA)-VPR(NU-ND,IA))
  S=0.
  UK=0.
  DO 12 J=1,JR
12 BBL(J)=0.
  IF(KDIX.LE.0) GO TO 26
4 IZT=NZPZ(2-ND,IABS(NU-NEND)+2)
  IZP=IZT
  IF(IZT.NE.NNESE)GO TO 10
  ALWR=ALV(IA)-DARF*ALN
  WRITE(IDRU,8)IZT,NAMP,ALWR,(ALTX(J,ITIT2),J=1,4)
8 FORMAT (A1,25HBOUNDARY LAYER AIRFOIL ,12A1,10H ALPHA =,F6.2,
  *24H DEGREES RELATIVE TO THE,4A4)
  IZT=NZPZ(1,0)
10 WRITE (IDRU,14) IZT,SFTX(JU),(RETX,LR(J),MU(J),J=1,JR)

```

APPENDIX

```

14 FORMAT (A1,1X,A5,8H SURFACE,3X,5(4X,A3,I6,8H000 MU =,I2))
   IF(IZP.NE.MNESE)GO TO 26
20 IZT=NZPZ(1,0)
   WRITE (IDRU,22) IZT,(J,HTX,D2TX(KDTX),J=1,JR)
22 FORMAT (A1,6X,1HS,8X,2HU ,5(4X,I1,4X,A3,4X,A7))
   GO TO 26
24 NUNSD = NU - NDS
   DSR = DS(NUNSD)
26 UK1=ABS(VPR(NU,IA))
   S = S + DSR
   DO 27 J=1,JR
   DSZ = DSR
   UR = UK
   HVGL=H(J)
   MT=MU(J)
   D2I=D2(J)
   WR=WRE(J)
   CALL GRS
   DD(J) = D2R
   IF(KDTX.EQ.2)DD(J)=UK1*RE(J)*D2R*1.E-6
   IF(KDTX.NE.3)GO TO 28
   CALL H12B(HR,H12R,QTS)
   DD(J)=D2R*H12R
28 IF(UK.NE.0.)GO TO 30
   SAR(J)=0.
   SUR(J)=0.
30 SAR(J)= SAR(J)+XA
   IF(HVGL*HR.GE.0.)GO TO 33
   UTR=UK+(UK1-UK)*XU/DSR
   CPTR(J)=1.-UTR*UTR
   HVGL=-1.51509
33 SUR(J)= SUR(J)+XU
   IF(HVGL.GT.HBBL.AND.HR.LE.HBBL) BBL(J)=S-SUR(J)-(DSR-XU)
   *(HBBL-HR)/(HVGL-HR)
   H(J)=HR
27 D2(J)=D2R
   IF(MAG.LE.0)GO TO 31
   IF(ABS(H(MAG)).GT.1.62)GO TO 31
   RPL=FKT*ALOG(UK1*RE(MAG)*D2(MAG))
   IF(RPL.LT.1..OR.RPL.GT.4.)GO TO 31
   IP=IP+1
29 RP(IP)=RPL
   HP(IP)=ABS(H(MAG))+.2*FLOAT(JU-1)
   IF(IP.LT.32)GO TO 31
   CALL POLZUG(1,HP,RP,IP,.TRUE.,TRUE)
   IP=1
   GO TO 29
31 IF(KDTX.LE.0) GO TO 35
   IZT=NZPZ(1,0)

```

APPENDIX

```

WRITE(IDRU,32) IZT,S,UK1,(H(K),DD(K),K=1,JR)
32 FORMAT (A1,F9.5,F8.3,5(F13.4,F10.6))
35 IF(NU-NEND)34,36,34
34 UK = UK1
   NU = NU +ND
   GO TO 24
36 IF(IP.GE.2)CALL POLZUG(1,HP,RP,IP,.TRUE.,TRUE)
   IF(KDIX.NE.3)GO TO 37
   IZT=NZPZ(1,0)
   WRITE(IDRU,39) IZT,(KBL,CPTR(K),K=1,JR)
39 FORMAT (A1,17X,5(A1,5X,11HCP(TRANS) =,F6.2))
37 DO 38 J=1,JR
   CALL H12B(H(J),H12R,OTS)
   IF(H12R.GT.HABLC)H12R = HABLC
   IF(H(J).LT.0..AND.BBL(J).EQ.0.) BBL(J)=SUR(J)
   BBF=2.
   IF(BBL(J).GT.BBLI) BBF=-2.
   CW(J,JU,IA)=(UK1**((2.5+.5*H12R))*D2(J)*BBF
   SA(J,JU,IA) = S - SAR(J)
38 SU(J,JU,IA) = S - SUR(J)
   I=0
   ITZ=NZPZ(2,4*(NALE-NAI+1)+2)
   IF(ITZ.NE.NNESE.AND.NAX.LT.0)GO TO 48
44 WRITE(IDRU,46) ITZ,NAMP,(ALT(X,J,ITITZ),J=1,4),ALN
46 FORMAT (A1,18HSUMMARY AIRFOIL ,12A1,3X,
*31HANGLE OF ATTACK RELATIVE TO THE,4A4,3X,
*8HALPHAO =,F5.2,8H DEGREES)
   ITZ=NZPZ(1,0)
   WRITE(IDRU,47) ITZ,BBLI
47 FORMAT (A1,38H * INDICATES BUBBLE ANALOG LONGER THAN,F5.3)
   ITZ=NZPZ(2,0)
48 WRITE(IDRU,49) ITZ,(RETX,LR(J),MU(J),J=1,JR)
49 FORMAT (A1,5X,5(5X,A3,I6,10H000 MU =,I2))
   IF(I.NE.0) GOTO 51
   DO 60 I=NAI,NALE
51 ITZ=NZPZ(2,4)
   IF(ITZ.EQ.NNESE)GO TO 44
   ALWR=ALV(I)-DARF*ALN
   WRITE (IDRU,50) ITZ,ALWR
50 FORMAT (A1,8H ALPHA =,F6.2,8H DEGREES)
   ITZ=NZPZ(1,0)
   WRITE (IDRU,101) ITZ,(J,STXT,J=1,JR)
101 FORMAT (A1,4X,5(I5,A2,5H TURB,3X,5HS SEP,4X,2HCD))
   DO 52 K=1,2
   DO 53 J=1,JR
   CWPM=CW(J,K,I)
   LD(J)=KBL
   IF(CWPM.LT.0.) LD(J)=KST
53 CD(J)=ABS(CWPM)

```

APPENDIX

```

ITZ=NZPZ(1,0)
52 WRITE (IDRU,54) ITZ, SFTX(K), (SU(J,K,I),SA(J,K,I),CD(J),LD(J),J=1,JR)
54 FORMAT (A1,2X,A5,F9.4,F8.4,F7.4,A1,4(F10.4,F8.4,F7.4,A1))
DO 56 J = 1,JR
56 CALL VISC(I,J,CAD(J),CWD(J),CMV(J))
ITZ=NZPZ(1,0)
WRITE (IDRU,58) ITZ, (KBL,CAD(J),CWD(J),J=1,JR)
58 FORMAT (A1,6H TOTAL,A1,3X,4HCL =,F6.3,5H CD =,F6.4,
*4(A1,4X,4HCL =,F6.3,5H CD =,F6.4))
ITZ=NZPZ(1,0)
WRITE (IDRU,102) ITZ, (KBL,CMV(J),J=1,JR)
102 FORMAT (A1,5(A1,14X,4HCM =,F7.4))
60 CONTINUE
IF(NALE.EQ.NAL.AND.MAG.NE.0)CALL GCLOSE
RETURN
END
SUBROUTINE UMP(U,D2,H,FUM)
COMMON P(512),X(121),Y(121)
COMMON/TRIT/DLV,SUMP,XTRI(4),NU,ND
COMMON/BL/DUM(5),WRE,H,MD,DU(7),US,DUMY
FUM=-18.4
IF(H.LE.0)GO TO 9
IF(M.GE.3)GO TO 4
MXT=2*M-1+(ND+1)/2
N=NU-ND
FUH=X(N)+SUMP*(X(NU)-X(N))/DLV -XTRI(MXT)
GO TO 9
4 FUH=ALOG(WRE*WRE*D2*U)+21.74+.36*FLOAT(M-3)-18.4*M
9 RETURN
END
SUBROUTINE CDCF( RET,H,CD,CF,H12)
COMMON /GRZK/CDK,AA(7),BB(7)
CALL H12B(H,H12,EPST)
IF(H)1, 3,2
1 RZ = ALOG((H12-1.)*RET)
CD = CDK*EXP(-.166692*RZ)
CF = .045716*EXP(-1.26*H12 -.232*RZ)
GO TO 3
2 CD = ((H*6.8377961- 20.521103)*H + 15.707952)/RET
CF = EPST/RET
3 RETURN
END
SUBROUTINE GRUP(H,D2,U,Z,DZ,DD2,V)
COMMON /GRZK/CDK,AA(7),BB(7)
COMMON/BL/DUM(4),DL,WRE,MD,MA,DU(7),US,DUMY
Z = D2*ABS(H)
RET = WRE*WRE*U*D2
CALL CDCF( RET,H,CD,CF,H12)
IF(MA)1,1,2

```

APPENDIX

```

1 V = 0.
  GO TO 13
2 GO TO(13,13,13,6,6,5,5,3),MA
3 IF(H)4,1,1
4 V = 25.*(H+1.98)*D2*US
  GO TO 12
5 IF(H)6,1,1
6 B = BB(MA)
  PSI = AA(MA)
  IF(B.NE.0.)PSI=B*ALOG(RET)+PSI
  IF(PSI.GE.1.52.AND.PSI.LE.1.99)GO TO 11
  B=0.
  IF(PSI.LT.1.52)PSI=1.52
  IF(PSI.GT.1.99)PSI=1.99
11 HZ = SIGN(PSI,H)
  CALL CDCF( RET,HZ,CDS,CFS,H12S)
  V=(U*(CDS-(PSI+B)*CFS)+D2*US*(B-PSI+H12*(B+PSI)))/(B+ABS(H)-1.)
12 IF(V)13,1,1
13 VDU =V/U
  USU =US/U
  DD2 = (CF-(2.+H12)*USU*D2+VDU)*DL
  DZ = (CD - 3.*Z*USU + VDU)*DL
14 RETURN
  END
  SUBROUTINE H12B(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 = ((-.03172850655*H12+.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-89.58214201)*H + 79.87084472
  GO TO 5
  4 H12 = (H-1.36364)/(H*4.36364 + 5.36364)
  5 RETURN
  END
  SUBROUTINE GRS
  COMMON/BL/HVGL,D2,UK,UK1,DL,WRE,MU,MA,V,V1,HR,D2R,XA,XU,DCQ,USTR,
  *VSTR
  COMMON/TRIT/DLV,SUMP,XTRI(4),NU,ND
  DATA EAP,EUM,HE/.000005,.0001,-1.0/
  BIT=0.
  H=HVGL
  IF(MA)3,101,3
101 V1 = 0.
  V = 0.
  IF(HE.LE.0..OR.UK.LE.0.)GO TO 3

```


APPENDIX

```

1 IF (ABS(UK-UKV)+ABS(UK1-UK1V)+ABS(DL-DLV)+ABS(H-HV)-.1E-6)2,3,3
2 IF (ABS(D2+WRE-D2V+WREV).GE..1E-6) GO TO 3
  BIT = 1.
  D2E = D2E+WREV/WRE
3 D2V=D2
  HV = H
  UKV= UK
  UK1V = UK1
  DLV = DL
  WREV = WRE
  DCO = 0.
  XA = DL
  XU = DL
  VV = V
  IF(DL)4,4,5
4 HE=H
  D2E= D2
  GO TO 50
5 USTR = (UK1-UK)/DL
  IF(MA.LT.4)VSTR = (V1-V)/DL
  IF(UK)6,7,6
6 IF((UK1-UK)/UK - 1.18,8,7
7 D2E = .290043/(WRE*SQRT(USTR))
  HE = 1.619977
  GO TO 50
8 IF(D2)9,9,10
9 D2E = (.664108/WRE)*SQRT(2.*DL/(UK1+UK))
  HE = 1.572584
  GO TO 50
10 X = 0.
  SUMP=X
  ISTAB = 0
  IF(H.LT.0.)XU = 0.
  IF(BIT)13,13,11
11 CALL UMP(UK,D2,H,FUM)
  GO TO 40
13 IF(X +DL -DLV)12,12,19
12 XS = X
  UK = UKV + XS*USTR
  VG= VV + VSTR*XS
  IF(H)14,14,16
14 HAP = 1.46
  IF(HAP+EAP +H)20,20,15
15 XA = X
  CALL H12B(H,H12,EPST)
  D2E = D2*(UK/UK1)**((5.+H12)*.5)
  HE = H
  GO TO 50
16 HAP = 1.515095

```

APPENDIX

```

SUMP=X
IF(HAP+EAP-H)17,18,18
17 CALL UMP(UK,D2,H,FUM)
   IF(FUM+EUM)20,18,18
18 H = -H
   XU = X
   ISTAB = 0
19 DL = DLV - X
   GO TO 12
20 CALL GRUP(H,D2,UK,Z,DZ,DD2,VG)
   ESP = .002
   IF(MA-3)22,22,21
21 ESP = .001
   IF(X.EQ.0.)V = VG
22 XS = XS + .5*DL
   D2M = D2 + .5*DD2
   ZM = Z + .5*DZ
   UM = UKV + USTR*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 TO 13
30 IF(HM - HAP+EAP)31,32,32
31 DL = .5*DL*(ABS(H)-HAP)/(ABS(H)-HM)
   GO TO 13
32 IF(H.LT.0.)HM = -HM
   IF(MA.LT.4)VM = VV + XS*VSTR
   CALL GRUP(HM,D2M,UM,ZM,D2M,DD2M,VM)
   ISG = 3
   D2E = D2 + DD2M
   IF(D2E)25,25,33
33 HE = (Z+D2M)/D2E
   ISG = 4
   IF(HE-2.)34,25,25
34 HDIFF = HE-ABS(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+EAP)37,38,38
37 DL = DL*(ABS(H)-HAP)/(ABS(H)-HE)
   GO TO 13

```

APPENDIX

```

38 IF(H)39,39,40
39 HE = -HE
   GO TO 42
40 UE = UK + USTR*DL
   SUMP=X+DL
   CALL UMP(UE,DZE,HE,FUME)
   IF(FUME-EUM)42,42,41
41 DL = DL*FUM/(FUM-FUME)
   GO TO 13
42 DCQ = DCQ + DL*VM
   X = X + DL
   IF(X -DLV)47,50,50
47 H = HE
   DZ = DZE
   GO TO(43,13,43,45,13,43),ISGV -
43 IF(HS-.1*ESP)44,13,13
44 DL = 2.*DL
   ISTAB = ISTAB-1
   GO TO 13
45 IF(HDIFF)44,13,13
49 STP = 1/(ISTAB-9)
50 HR = HE
   DZR = DZE
   RETURN
   END
   FUNCTION RUND(A,B)
   RUND = (AINT(A*B+SIGN(.5,A)))/B
   RETURN
   END
   FUNCTION COSG(A)
   COSG = COS(A*.0174532925199)
   RETURN
   END
   FUNCTION SING(A)
   SING = SIN(A*.0174532925199)
   RETURN
   END
   FUNCTION CSLG(A,B)
   CSLG = ALOG(ABS(SING(A-B)))
   RETURN
   END
   SUBROUTINE DRAW(WC,WS,WL,DK,DM,FL,AG,MA)
C   CALL DRAW(WC,WS,WL,DRAK(J),DRAM(J),FLA(J),ABGR,MKP)
C   WHEN MKP = 0, DK = KAPPA; OTHERWISE, DK = K.
   COSP = COSG(AG*FL)
   FK = DK
   IF(MA)2,1,2
   1 FK = FK*(1.-COSP)/(1.+COSP)
   2 SINP = SING(AG*FL)

```

APPENDIX

```

WL = -DM*ALOG(1.+FK)
IF (FL.EQ.0.) FK = 1.
BETA = (COSP-1.)/FK + COSP
BQM1 = BETA**2 - 1.
WUBEO = SQRT(ABS(BQM1))
U = (1.+BETA)*SINP/(1.+COSP)
IF (BQM1) 3, 5, 4
3 WF = ALOG(ABS((WUBEO+U)/(WUBEO-U)))
GO TO 6
4 WF = 2.* ATAN(U/WUBEO)
GO TO 6
5 WF = 0.
6 WC = (WUBEO*WF - SINP - BETA*FL*AG*1.745329E-2)*DM
WS = (COSP-1.)*(1.-(1./FK + 1.)*ALOG(1.+FK))*DM
RETURN
END
SUBROUTINE TRAPRO
DIMENSION FLS(2), FLA(2), DRAK(2), DRAM(2), AC(4,3), D(3),
1 WSI(2), WCI(2), FINT(3), A(4), HK(2), R(3), FKERN(30)
COMMON/EA/ILES, IDRU, ISTA, NNESE
COMMON/PRAL/DLT, DLTU, ALN, ALV(14), NAL, ITP, NAMP(12), CHL(14), CRL(14)
1 , CPV(2), ALTX(4,2), DARF, ITIT1, ITIT2
COMMON P1(121), P(121), XP(121), YP(121), PUFF(14), AGAM(14), X(121),
1Y(121), DS(122), VF(121), ARG(121), ANI(28), ALFR(29), IZZ, KFU, NQ, NUPRO,
2 JAB, JST, CM, ETA, ABFA, PI, BOGEN, DARG, PURES(13), GAP(450), ALFA(29)
DATA ABSZ/0./
CALL WANDEL(NUPRO, NAMP, 12, 5)
ALFR(JAB+1)=0.
ABZT=ANI(JAB)
IF (ABS(ABZT-ABSZ).LT..1) GO TO 14
IB=INT(.25*ABZT+.1)
HQ=2*IB
NKR=2*NQ
ABSZ=FLOAT(NKR)
ABGR=360./ABSZ
HABGR=.5*ABGR
DO 8 M=1, IB
ARI=FLOAT(MQ+1-2*M)*HABGR
8 FKERN(M)=ABGR*COSG(ARI)/(SING(ARI)*PI)
14 MAGAM=INT(AGAM(3))
NQ=NKR+1
IF (MAGAM.EQ.0) GO TO 22
NZT=NZPZ(3,0)
WRITE(IDRU,82)NZT
22 DO 23 I=1, 29
23 ALFA(I)=ALFR(I)
I=1
J=1
24 FLS(J)= PURES(I)*ABFA

```

APPENDIX

```

CALL DRAW(WC,WS,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.
   GOTO 34
26 WI = COSG(ABGR*FLA(J))
   IF(PURES(I+2)-1.)27,30,29
27 DRAK(J)= .1*PURES(I+3)
28 DRAM(J)= .1*PURES(I+4)
   GOTO 34
29 DRAK(J)=((.1*PURES(I+4))*(-10./PURES(I+3))-1.)*(1.+WI)/(1.-WI)
   DRAK(J)= RUND(DRAK(J),1000.)
   DRAM(J) = .1*PURES(I+3)
   GO TO 34
30 AA = .05*(1.-WI)*PURES(I+3)
   WILN = ALOG(.1*PURES(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) = RUND(FM,1000.)
   DRAS = .05*PURES(I+3)*(WI+1.)/FM
   DRAK(J) = RUND(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)+WSI(1))
   D(2) = -WLI *(WCI(2)+WCI(1))
   D(3) = WCI(1)*WSI(2)-WCI(2)*WSI(1)
   ITMOD=INT(PURES(11))
   RUF=100.
   IF(ITMOD.GE.4.AND.ITMOD.LE.6)RUF=1000.
   ITMR=ITMOD
   SHKS = .1*PURES(12)
   HKST=.1*ABS(PURES(13))
35 DO 36 J=1,3
36 AC(1,J)= 0.
   ALIV = 0.
   SINAI = 0.

```

APPENDIX

```

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) = COSAI-CSAIP
AC(4,2) = SINAI-SNAIP
AC(4,3) = 0.
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) = -FIIP*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-JAB-1)42,43,43
42 ALIV = ALFA(J)
SINAI = SNAIP
COSAI = CSAIP
FNI = ANI(J)
J=J+1
GO TO 37
43 DO 47 J=1,2
IF(FLA(J)) 47,47,49
49 CALL DRAW(WC,WS,WL,DRAK(J),DRAM(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
DO 52 J=1,4
A(J) = 0.
DO 52 I=1,3
52 A(J) = A(J)+D(I)*AC(J,I)
C SOLUTION OF TRANSCENDENTAL EQUATION
53 I=0
FV = 9.E9
PHISH = .5 *(ALIS+ALISP)

```

APPENDIX

```

60 CSLI = CSLG(PHISH,ALIS)
   CSLIP = CSLG(PHISH,ALISP)
   FP = A(1)+A(2)*CSLI+A(3)*CSLIP+A(4)*BOGEN*(90.+PHISH)
   IF(I.GE.20) GO TO 66
   IF(ABS(FP)-ABS(FV).LT.-.5E-9) GO TO 62
   I = I + 1
   PHISH = PHISH - PDIF
   GO TO 60
62 PDIF = -FP / (A(2)/(PHISH-ALIS) + A(3)/(PHISH-ALISP))
   I = I + 1
65 FV = FP
   PHISH = PHISH + PDIF
   IF(PHISH.LT.ALIS.AND.PHISH.GT.ALISP) GO TO 60
   WRITE(IDRU,64)MER,ITMOD
64 FORMAT (66HOTRASCENDENTAL EQUATION HAS DIVERGED. CHECK TRAI AND
   *TRA2 CARDS.,12H ITERATION,I2,8H MODE ,I1)
   STOP
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))/D(2)
   HK(2) = (FINT(1)*WLI+FINT(3)*WCI(1))/D(2)
   HKS = HK(1)+HK(2)
   IF(ITMOD.EQ.0.OR.ABS(HKS-SHKS).LT.HKST) GO TO 74
   IF(MAGAM.LT.2.AND.MAGAM-MER.NE.1) GO TO 100
   GO TO 76
74 ITMOD = 0
   IF(MAGAM.EQ.0) GO TO 300
76 NZT = NZPZ(2,JAB+4)
   WRITE(IDRU,77)NZT,NUPRC,MER,ITMR
77 FORMAT (A1,42HTRASCENDENTAL EQUATION RESULTS AIRFOIL ,I4,
   *12H ITERATION,I2,8H MODE ,I1)
   NZT = NZPZ(1,0)
   WRITE(IDRU,78)NZT
78 FORMAT (A1,69H NU ALPHA* OMEGA' OMEGA K MU K H
   * LAMBDA LAMBDA*)
   JH = 1
   DO 85 JN=1,JAB
79 NZT = NZPZ(1,0)
   IF(JN.NE.1.AND.JN.NE.JAB) GO TO 83
   X1 = .5*(1.+COSG(FLA(JH)*ABGR))
   WHK = (1.+DRAK(JH)*(1.-X1)/X1)**(-DRAM(JH))
   WSTR = DRAM(JH)*DRAK(JH)/X1
   WRITE(IDRU,82)NZT,ANI(JN),ALFA(JN),WSTR,WHK,DRAK(JH),DRAM(JH),HK(J
   1H),FLA(JH),FLS(JH)
82 FORMAT (A1,F6.2,F8.2,F8.3,F7.3,3F8.3,F8.2,F7.2)
   JH = 2
   GO TO 85

```

APPENDIX

```

83 WRITE(IDRU,82)NZT,ANI(JN),ALFA(JN)
85 CONTINUE
   IF(ITMOD.EQ.0) GO TO 300
100 IF(MER)103,102,103
102 DAL = .1
   GO TO 104
103 IF(HKS-HKSV.EQ.0.)GO TO 74
   DAL = (SHKS-HKS)*DAL/(HKS-HKSV)
   DALD=DAL
   IF(ITP.EQ.0)DAL=RUND(DAL,RUF)
   IF(MAGAM.EQ.0.)GO TO 1004
   NZT=NZPZ(2,0)
   WRITE(IDRU,1003)NZT,MER,HKS,DALD,DAL
1003 FORMAT (A1,10H ITERATION,I2,3X,5HK S =,F9.6,3X,7HDELTA =,F12.8,
   *3X,9HROUNDED =,F6.2)
1004 IF(DAL.EQ.0.)GO TO 74
   IF(MER.GE.3.AND.ABS(DALV).LE.ABS(DAL))GO TO 74
104 DALV=DAL
   IF(ITMOD.GE.4)GO TO 113
   DO 111 J=1,JAB
   IF(ITMOD.NE.2.AND.J.LE.JST) ALFA(J)=ALFA(J)+DAL
   IF(ITMOD.NE.1.AND.J.GT.JST) ALFA(J)=ALFA(J)-DAL
111 CONTINUE
   GO TO 112
113 IF(ITMOD.GE.7)GO TO 114
   IF(ITMOD.NE.5)DRAK(1)=DRAK(1)+DAL
   IF(ITMOD.NE.4)DRAK(2)=DRAK(2)+DAL
   GO TO 112
114 IF(ITMOD.NE.8)ALFA(JST)=ALFA(JST)+DAL
   IF(ITMOD.NE.7)ALFA(JST+1)=ALFA(JST+1)-DAL
112 HKSV=HKS
   MER =MER+1
   GOTO 35
300 AK = .5*(COSG(PHISH-ALFA(JST+1))/SING(PHISH-ALFA(JST+1))
1   -COSG(PHISH-ALFA(JST))/SING(PHISH-ALFA(JST)))
   AKP =AK*180./9.8696044
   PHIM = 0.
   NU=1
   I= 1
   ANU =0.
   JH=0
   VI= 0.
302 JH=JH+1
   FF1 = COSG(ABGR*FLA(JH))
   FF2 = DRAK(JH)/(1.+FF1)
   FG1 = COSG(ABGR*FLS(JH))
   FG3 = .6/(FG1-1.)
304 VI= VI - CSLG(PHIM-90., ALFA(I))
   GO TO 310

```


APPENDIX

```

306 ARGN = ANU
   IF(ANU.GT..5* ABSZ)ARGN= ABSZ - ANU
   CSP = COSG(ARGN*ABGR)
   F=0.
   IF(ARGN.LT.FLA(JH))F=DRAM(JH)*ALOG((CSP-FF1)*FF2+1.)
   G=0.
   IF(ARGN.LT.FLS(JH))G=-HK(JH)*ALOG(1.-((CSP-FG1)*FG3)**2)
   P(NU)= F+G+CSLG(ANU*HABGR-90.,ALFA(I)) + VI
   P1(NU)=P(NU)-AK*ABS(SING((ANU*HABGR - 90.) - PHISH))
   NU = NU + 1
   ANU= ANU+ 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=0.
   B2=0.
   DO 324 I=1,NKR
   PS=PS+P(I)
   BI = 2*(I-1)
324 B2 = B2 + SING(BI*ABGR)*P(I)
   V1 = 2.*EXP(PS/ABSZ)
   SXI = .00000000
   SY=0.
   DO328 N=1,NQ
   Q=0.
   DO326 M=1,IB
   MN = N + 1 + MQ - 2*M
   MM = 2*N - MN
   IF(MN.GT.NKR) MN = MN - NKR
   IF(MM.LT.1) MM = MM + 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) = 0 - 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*COSG(ARG(N))
   SXI= SXI+ XP(N)
328 SY = SY + YP(N)
   SX = SXI
   XPK = SX/(ABSZ - 1.)
   YPK = SY/(ABSZ -1.)

```

APPENDIX

```

DO 329 N=2,NKR
XP(N)=XP(N)-XPK
329 YP(N)=YP(N)-YPK
CALL CINT(XP,X,NQ,IZZ)
CALL CINT(YP,Y,NQ,IZZ)
RQV = 0.
DO 330 N=2,NKR
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)=SORT(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
DARG = 19.09859 *(3.*YNAS/XNAS - (YNAS/XNAS)**3)
IF(ABS(SX)+ABS(SY).LT..0001*ABSZ) GO TO 335
SX=STREF*SX*200.
SY=STREF*SY*200.
NZT=NZPZ(2,0)
WRITE (IDRU,334) NZT,SX,SY
334 FORMAT (A1,14HWARNING - SX =,F6.3,3X,4HSY =,F6.3)
335 DO 331 N=2,NQ
XR=X(N)
X(N)= 1.-B*Y(N)-AT*XR
Y(N)= B*XR -AT*Y(N)
ARG(N) = ARG(N) - DARG
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)
NHKW=NQ/12
DLT = Y(NHKW)/(BOGEN*(1.-X(NHKW)))
NPKW=NQ-NHKW+1
DLTU=-Y(NPKW)/(BOGEN*(1.-X(NHKW)))
X(1) = 1.
332 ARG(1) = ARG(1) - DARG
346 ITP=1
ALN=DARG
IF(PURES(13).GE.0.)GO TO 11
PURES(12)=10.*HKS
PURES(13)=.00001
11 RETURN

```

APPENDIX

```

END
SUBROUTINE CINT(P,Z,NQ,IZZ)
DIMENSION P(NQ),Z(NQ)
FCINT=0.
IF(IZZ.NE.0) FCINT=.083333333333
Z(1)=0.
PVV=P(NQ-1)
PV=P(1)
PL=P(2)
DO 10 N=2,NQ
NZ=N+1-(NQ-1)*(N/NQ)
PZ=P(NZ)
Z(N)=Z(N-1)+PL+PV+(PL+PV-PZ-PVV)*FCINT
PVV=PV
PV=PL
10 PL=PZ
RETURN
END
SUBROUTINE QIP(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(3)-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 DIA(X,Y,NP,D)
DIMENSION U(3),V(3),W(3),Z(3),A(3),B(3),X(121),Y(121)
D = 0.
XV=X(1)
DO 4 N=2,NP
NR = NP-1
XS=X(N)
1 IF(X(NR)-XS)3,2,2
2 NR = NR - 1
GO TO 1
3 IF(NR-N.LE.3)GO TO 5
DNN=Y(N)-Y(NR)-(Y(NR+1)-Y(NR))*(X(N)-X(NR))/(X(NR+1)-X(NR))
IF(DNN.LE.0)GO TO 4
NH=N
NRM=NR
D=DNN
4 XV=XS
5 TA=1.
YF=0.
IIT=0
7 IIT=IIT+1
IF(IIT.GT.8)GO TO 11
DO 6 I=1,3

```

APPENDIX

```

NO=NM+2-I
NU=NRM-2+I
U(I)=TA*(X(NO)+YF*Y(NO))
V(I)=TA*(Y(NO)-YF*X(NO))
W(I)=TA*(X(NU)+YF*Y(NU))
6 Z(I)=TA*(Y(NU)-YF*X(NU))
CALL QIP(U,V,A)
CALL QIP(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(XST.GE.U(1).AND.XST.LE.U(3))GO TO 21
IF(XST.GT.U(3))NM=NM-1
IF(XST.LT.U(1))NM=NM+1
GO TO 7
21 IF(XST.GE.W(1).AND.XST.LE.W(3))GO TO 22
IF(XST.GT.W(3))NRM=NRM+1
IF(XST.LT.W(1))NRM=NRM-1
GO TO 7
22 IF(ABS(YS)-.0001)9,8,8
8 YF=YF+YS/TA
TA=1./SQRT(1.+YF*YF)
GO TO 7
9 D = A(1)-B(1)+(A(2)-B(2)+(A(3)-B(3))*XST)*XST
11 RETURN
END
SUBROUTINE DIAGR(ISTIFT,ISTIM,MENV)
COMMON/PRAL/DLT,DLTU,ALN,ALV(14),NAL,ITP,NAMP(12),CML(14),CRL(14)
1 ,CPV(2),ALT(4,2),DARF,ITIT1,ITIT2
COMMON P1(121),P(121),XP(121),YP(121),PUFF(14),AGAM(14),X(121),
1Y(121),DS(122),VF(121),ARG(121),ANI(28),ALFA(29),IZZ,KFU,NO,NUPRO
EXTERNAL TRUE,YG4,YG5
LOGICAL PL,TRUE,YG4,YG5
DATA RZE,ALMIN,ALMAX/177.8,-5.,15./
PL=ISTIFT.GE.5
IF(ISTIM.GE.2)GO TO 10
XMX=0.
IF(MENV.EQ.1)GO TO 4
IF(PUFF(1).NE.0.)RZE=100.*PUFF(1)
QUER=1.3*RZE
HOCH=1.5*RZE
CALL DEFINE(PL,QUER,HOCH,-.2,1.1,-.2,1.3)
GO TO 10
4 ALMI=0.
ALMA=0.
IF(PUFF(1).NE.0.)ALMIN=PUFF(1)
IF(PUFF(2).NE.0.)ALMAX=PUFF(2)
HOCH=10.*(ALMAX-ALMIN)
CALL DEFINE(PL,260.,HOCH,-.3,4.3,ALMIN,ALMAX)

```

APPENDIX

```

10 DO 348 M=1,NAL
    VMX=0.
    DO 349 N=1,NQ
        V=.5*ABS(VPR(N,M))
        VQ=V*V
        IF(VMX.LT.VQ)VMX=VQ
        IF(XMX.LT.V)XMX=V
349 P(N)=V
    IF(MENV.NE.1)CALL POLZUG(ISTIFT,X,P,NQ,.TRUE.,YG4)
348 P1(M)=4.*VMX-1.
    IF(MENV.EQ.1)GO TO 20
    DO 350 I=1,NQ
350 P(I)=Y(I)+.1
    CALL SMOOTH(ISTIFT,X,P,XP,NQ,TRUE)
    GO TO 30
20 CALL POLZUG(ISTIFT,P1,ALV,NAL,.TRUE.,TRUE)
    ALMI=AMIN1(ALMI,ALV(1))
    ALMA=AMAX1(ALMA,ALV(NAL))
30 IF(ISTIM.EQ.1.OR.ISTIM.GE.3)RETURN
    IF(MENV.EQ.1)GO TO 40
    XMX=AMIN1(XMX,1.25)
    CALL ACHS(2,0.,-.01,-.02,0.,XMX,.05,.25,YG5)
    CALL ACHS(1,0.,-.01,-.02,0.,1.,.1,.5,TRUE)
    GO TO 50
40 CALL ACHS(1,0.,-.2,-.5,0.,4.,.1,.5,TRUE)
    CALL ACHS(2,0.,-.04,-.1,ALMI,ALMA,1.,5.,TRUE)
50 CALL GCLOSE
    RETURN
    END
    LOGICAL FUNCTION YG4(X,Y)
    A=X
    YG4=Y.GE..2.AND.Y.LE.1.25
    RETURN
    END
    LOGICAL FUNCTION YG5(X,Y)
    YG5=X.LT.-.0001.OR.ABS(Y-.1).GT..025
    RETURN
    END
    SUBROUTINE PUDECK
    COMMON P1(121),P(121),XP(121),YP(121),PUFF(14),AGAM(14),X(121),
1Y(121),DS(122),VF(121),ARG(121),ANI(28),ALFA(29),IZZ,KFU,NQ,NUPRO,
2JAB,JST,CM,ETA,ABFA,PI,BOGEN
    COMMON/PRAL/DLT,DLTU,ALN,ALV(14),NAL,ITP,NAMP(12),CML(14),CRL(14)
1,CPV(2),ALT(4,2),DARF,ITIT1,ITIT2
    COMMON/EA/ILES,IDRU,ISTA,NNESE
    DIMENSION INX(4),INY(4),INA(4)
    EQUIVALENCE(P(1),INX(1)),(P(5),INY(1)),(P(9),INA(1))
    STZ=100000.
    IF(X(1).GT.80.)STZ=1000.

```

APPENDIX

```

DO 6 N=1,NQ,4
NZE=MINO(4,NQ-N+1)
DO 4 M=1,NZE
L=N+M-1
INX(M)=INT(STZ*X(L)+SIGN(.5,X(L)))
INY(M)=INT(STZ*Y(L)+SIGN(.5,Y(L)))
ARGS=ARG(L)*BOGEN
IF(ITP.EQ.2)ARGS=XP(L)
4 INA(M)=INT(10000.*ARGS+SIGN(.5,ARGS))
6 WRITE(ISTA,8)(NAMP(M),M=3,7),N,(INX(M),INY(M),INA(M),M=1,NZE)
8 FORMAT(5A1,I3,12I6)
RETURN
END
SUBROUTINE STRAAK(TE,RUA,YB,MAXZ,ISTIFT)
DIMENSION XA(2),YA(2)
COMMON P1(121),P(121),XP(121),YP(121),PUFF(14),AGAM(14),X(121),
1Y(121),DS(122),VF(121),ARG(121),ANI(28),ALFA(29),IZZ,KFU,NQ
COMMON/PLTM/MPL,MGC,XZEH,YZEH,MSPLI
EXTERNAL TRUE
LOGICAL TRUE,PL
DATA YV,YXA/10000.,10000./
T = ABS(TE)
PL=ISTIFT.GE.5
YBL=ABS(YB)
QUER=T+10.
ORM=QUER-5.
HOCH=YBL
YMX = 0.
YMN = 0.
DO 2 I = 1,NQ
IF(Y(I).GT.YMX) YMX = Y(I)
IF(Y(I).LT.YMN) YMN = Y(I)
2 CONTINUE
IF(MGC)3,6,3
3 IF(TE)5,16,8
5 CALL GCLOSE
IF(T-1.)16,6,6
6 CALL DEFINE(PL,QUER,HOCH,-5.,ORM,0.,HOCH)
YXA = 5.-T*YMN
MSTR=0
GO TO 10
8 YXA = YXA + RUA
IF(YB.GT.0.) YXA = RUND(YV-T*YMN+1.5*RUA,1./RUA)
10 YV = YXA + T*YMX
IF(YV.GE.YBL.AND.MSTR.NE.0)GO TO 5
MSTR=1
DO 12 I=1,NQ
P1(I) = T*X(I)
12 P(I) = T*Y(I)+YXA

```

APPENDIX

```

CALL SMOOTH(ISTIFT,P1,P,XP,NQ,TRUE)
IF(MXZ)16,16,14
14 XA(1) = T
   XA(2) = 0.
   YA(1) = YXA
   YA(2) = YXA
CALL POLZUG(ISTIFT,XA,YA,2,.TRUE.,TRUE)
16 RETURN
   END
   SUBROUTINE STRDR(T,NT)
   DIMENSION XDR(8), YDR(8),T(42)
   COMMON/EA/ILES,IDRU,ISTA,NNESE
   COMMON P1(121),P(121),XP(121),YP(121),PUFF(14),AGAM(14),X(121),
1Y(121),DS(122),VF(121),ARG(121),ANI(28),ALFA(29),IZZ,KFU,NQ,NUPRO,
2JAB,JST,CM,ETA,ABFA,PI,BOGEN,DARG
   EQUIVALENCE (XDR(1),YP(106)),(YDR(1),YP(114))
   DO 8 N = 1,NT,8
   I = 0
   NZL = N+7
   IF(NZL.GT.NT)NZL=NT
   NZT=NZPZ(3,0)
1 WRITE(IDRU,2)NZT,NUPRO,DARG,CM
2 FORMAT (A1,8HAIRFOIL ,I4,3X,8HALPHAO =,F5.2,8H DEGREES,3X,
*5HCHO =,F7.4)
   NZT=NZPZ(1,0)
   WRITE(IDRU,4)NZT,(M,M=N,NZL)
4 FORMAT (A1,7(I2,17H X(CM) Y(CM) ))
5 NZT=NZPZ(1,0)
   IF(NZT.EQ.NNESE)GO TO 1
   I=I+1
   DO 6 K = N,NZL
   L = K-N+1
   XDR(L) = X(I)*.1*ABS(T(K))
6 YDR(L) = Y(I)*.1*ABS(T(K))
   WRITE(IDRU,10) NZT,(XDR(M),YDR(M),M=1,L)
10 FORMAT (A1,7(F9.2,F8.2,2X))
   IF(I.LT.NQ) GOTO 5
8 CONTINUE
   RETURN
   END

```

REFERENCES

1. Abbott, Ira H.; Von Doenhoff, Albert E.; and Stivers, Louis S., Jr.: Summary of Airfoil Data. NACA Rep. 824, 1945. (Supersedes NACA WR L-560.)
2. Abbott, Ira H.; and Von Doenhoff, Albert E.: Theory of Wing Sections. Dover Publ., Inc., c.1959.
3. Schlichting, Hermann (J. Kestin, transl.): Boundary Layer Theory. McGraw-Hill Book Co., Inc., 1955.
4. Smetana, Frederick O.; Summey, Delbert C.; Smith, Neill S.; and Carden, Ronald K.: Light Aircraft Lift, Drag, and Moment Prediction - A Review and Analysis. NASA CR-2523, 1975.
5. Eppler, R.: Direct Calculation of Airfoils From Pressure Distribution. NASA TT F-15,417, 1974. (Translated from Ingenieur-Archiv, vol. 25, no. 1, 1957, pp. 32-57.)
6. Lighthill, M. J.: A New Method of Two-Dimensional Aerodynamic Design. R. & M. No. 2112, British A.R.C., 1945.
7. Hess, John L.: The Use of Higher-Order Surface Singularity Distributions To Obtain Improved Potential Flow Solutions for Two-Dimensional Lifting Airfoils. Comput. Methods Appl. Mech. & Eng., vol. 5, no. 1, Jan. 1975, pp. 11-35.
8. Eppler, R.: Practical Calculation of Laminar and Turbulent Bled-Off Boundary Layers. NASA TM-75328, 1978. (Translated from Ingenieur-Archiv, vol. 32, 1963, pp. 221-245.)
9. Eppler, Richard; and Somers, Dan M.: Low Speed Airfoil Design and Analysis. Advanced Technology Airfoil Research - Volume I, NASA CP-2045, Part 1, 1979, pp. 73-99.
10. Kowalik, J.; and Osborne, M. R.: Methods for Unconstrained Optimization Problems. American Elsevier Pub. Co., Inc., 1968.
11. Eppler, R. (Francesca Neffgen, transl.): Laminar Airfoils for Reynolds Numbers Greater Than 4×10^6 . B-819-35, Apr. 1969. (Available from NTIS as N69-28178.) (Translated from Ingenieur-Archiv, vol. 38, 1969, pp. 232-240.)
12. Eppler, Richard: Turbulent Airfoils for General Aviation. J. Aircr., vol. 15, no. 2, Feb. 1978, pp. 93-99.
13. Squire, H. B.; and Young, A. D.: The Calculation of the Profile Drag of Aerofoils. R. & M. No. 1838, British A.R.C., 1938.
14. Woods, L. C.: The Theory of Subsonic Plane Flow. Cambridge Univ. Press, c.1961.

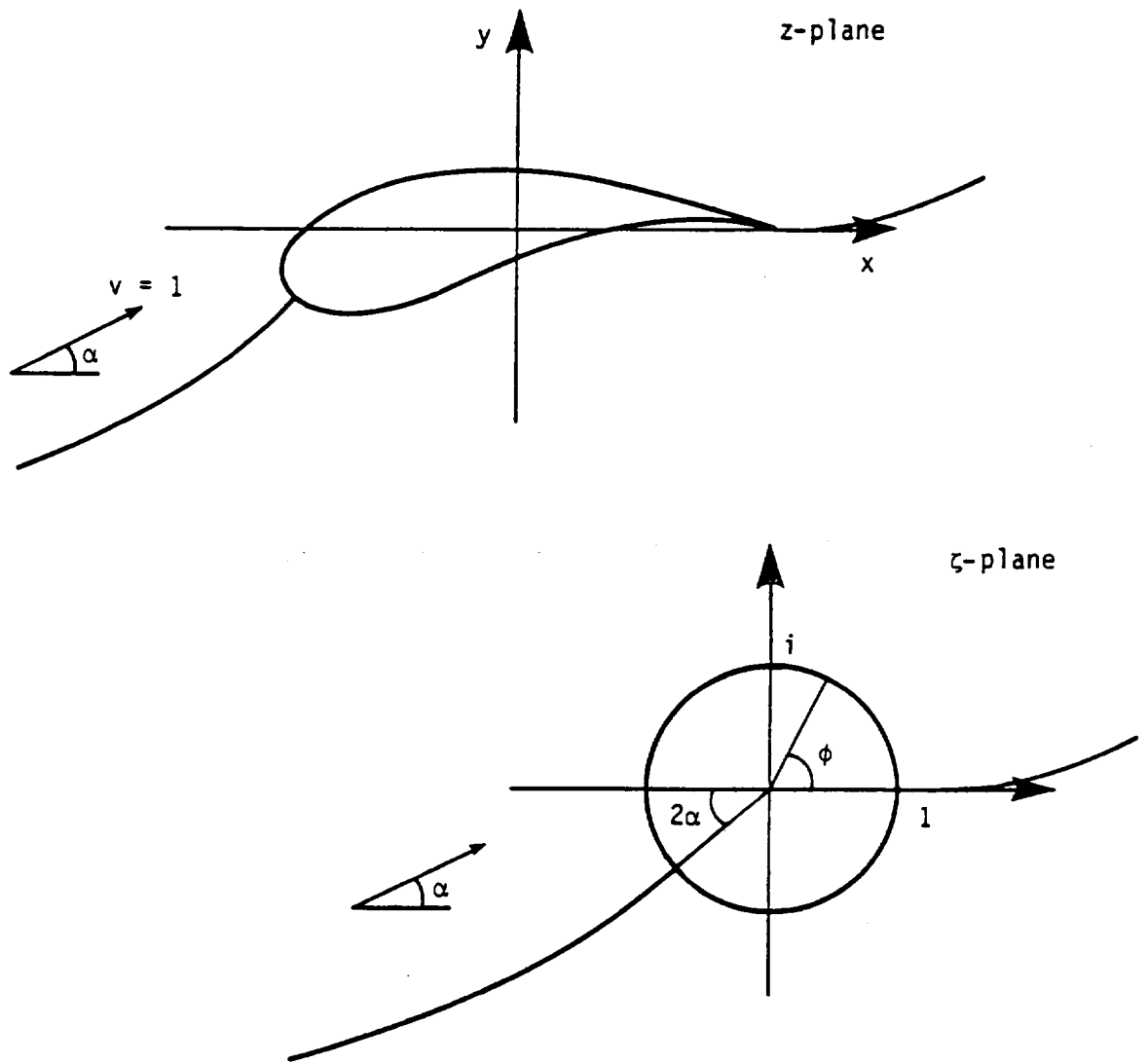


Figure 1.- Mathematical planes.

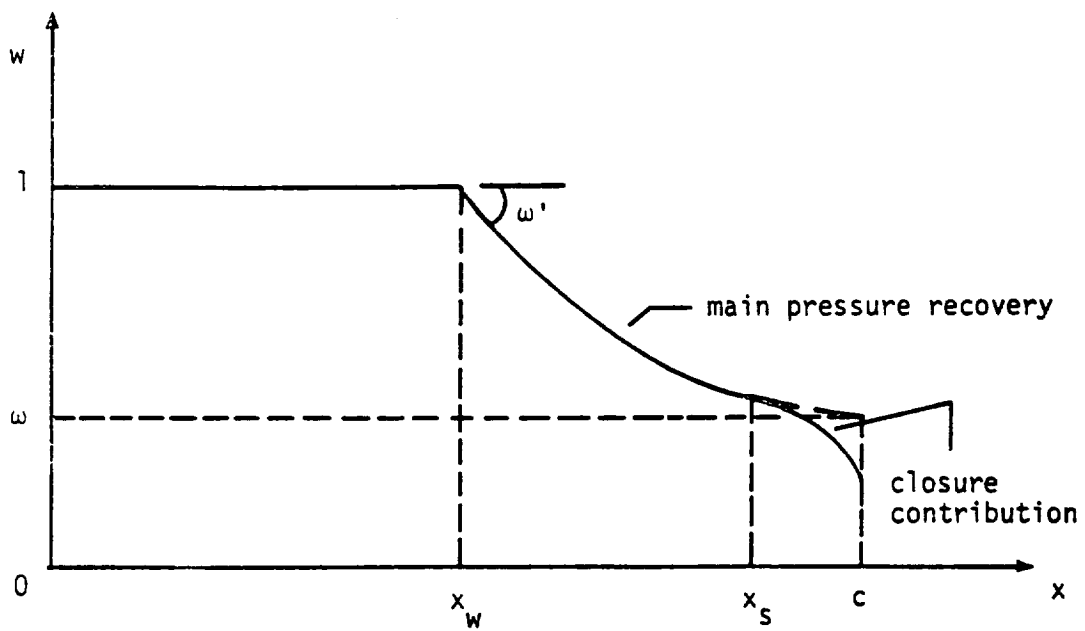


Figure 2.- Typical pressure-recovery distribution.

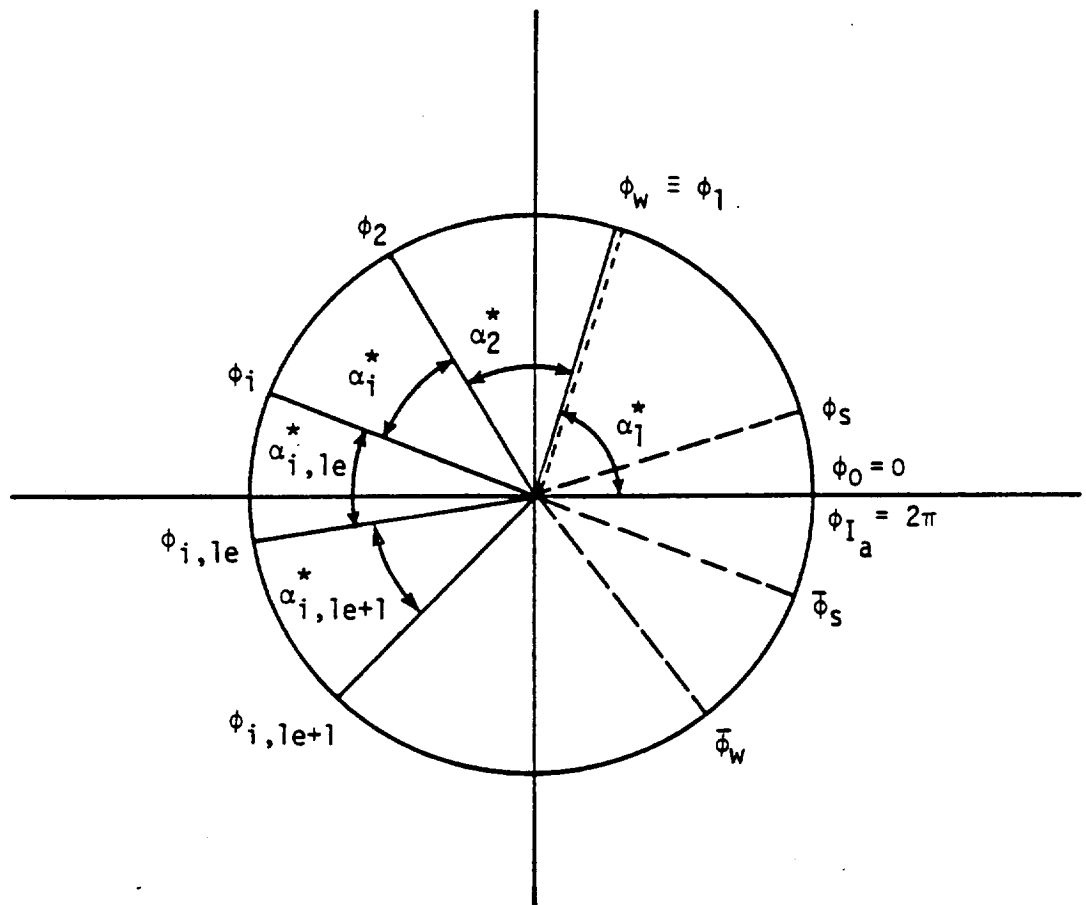


Figure 3.- Arc limits on ζ unit circle.

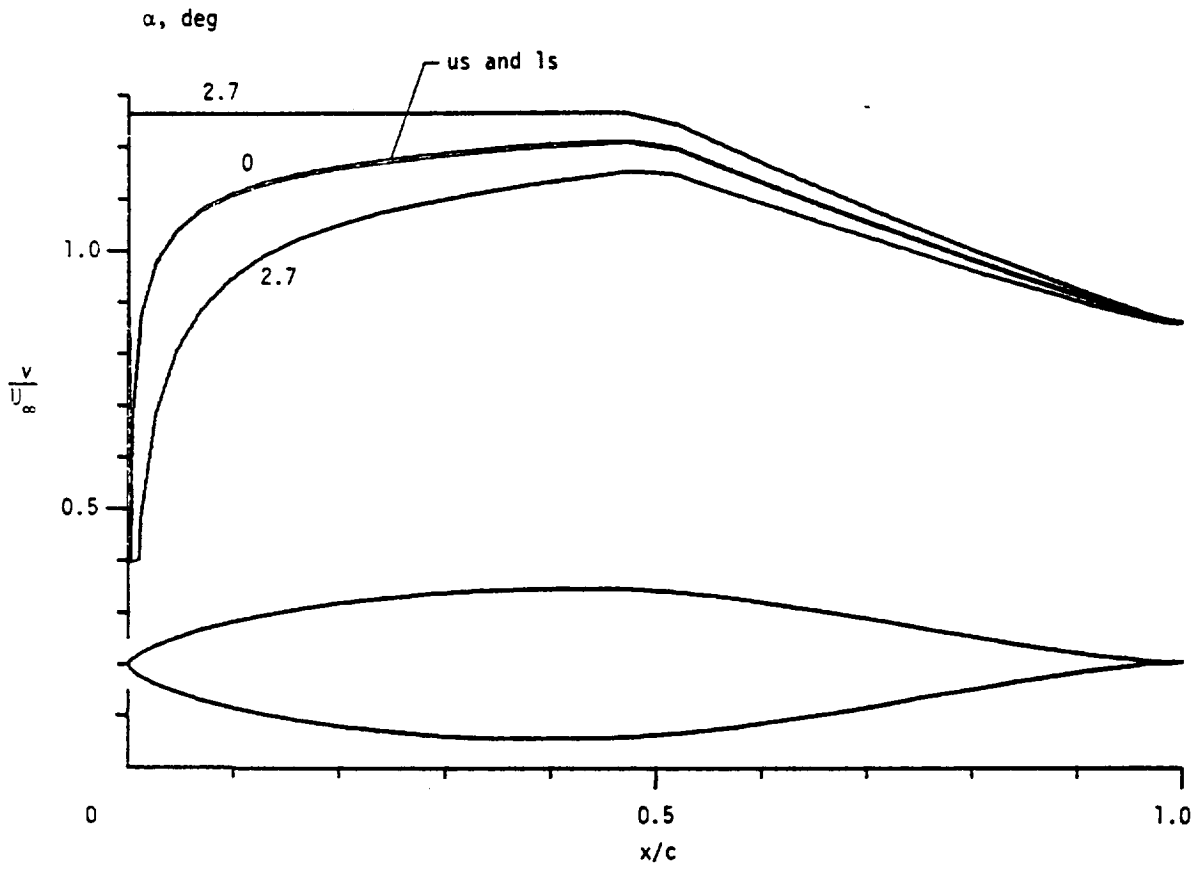


Figure 4.- Airfoil 1095. (α relative to zero-lift line.)

ORIGINAL PAGE IS
OF POOR QUALITY

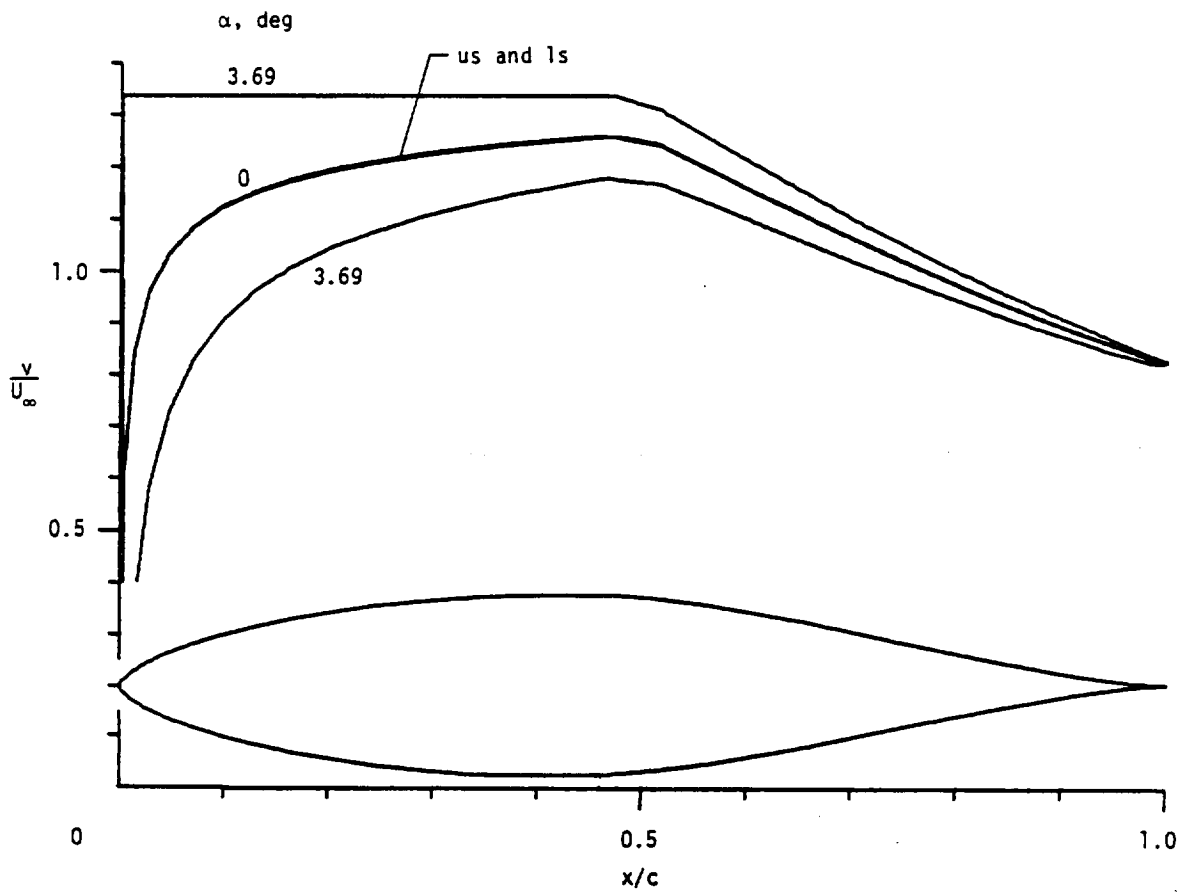


Figure 5.- Airfoil 1096. (α relative to zero-lift line.)

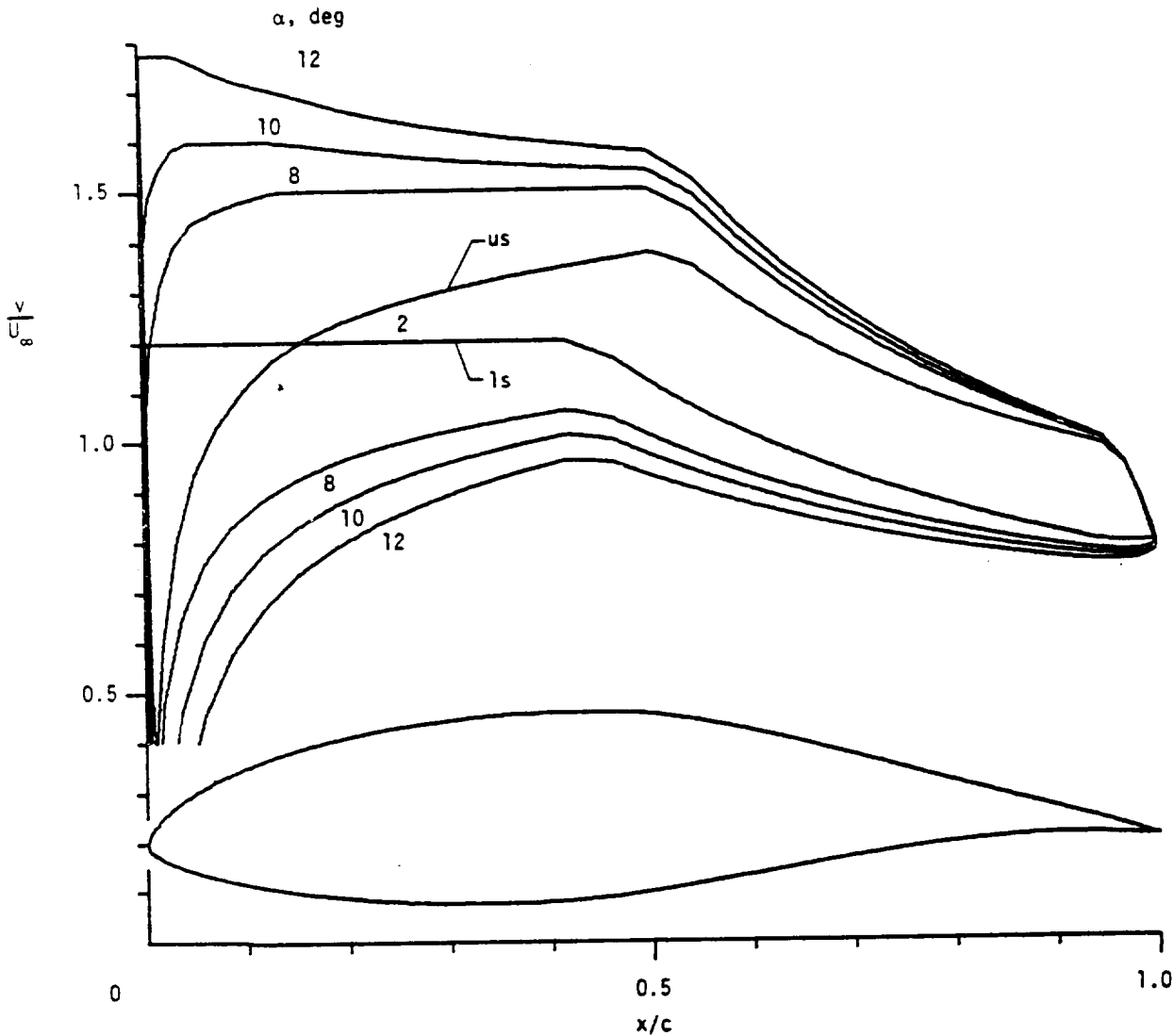


Figure 6.- Airfoil 1098. (α relative to zero-lift line.)

ORIGINAL PAGE IS
OF POOR QUALITY

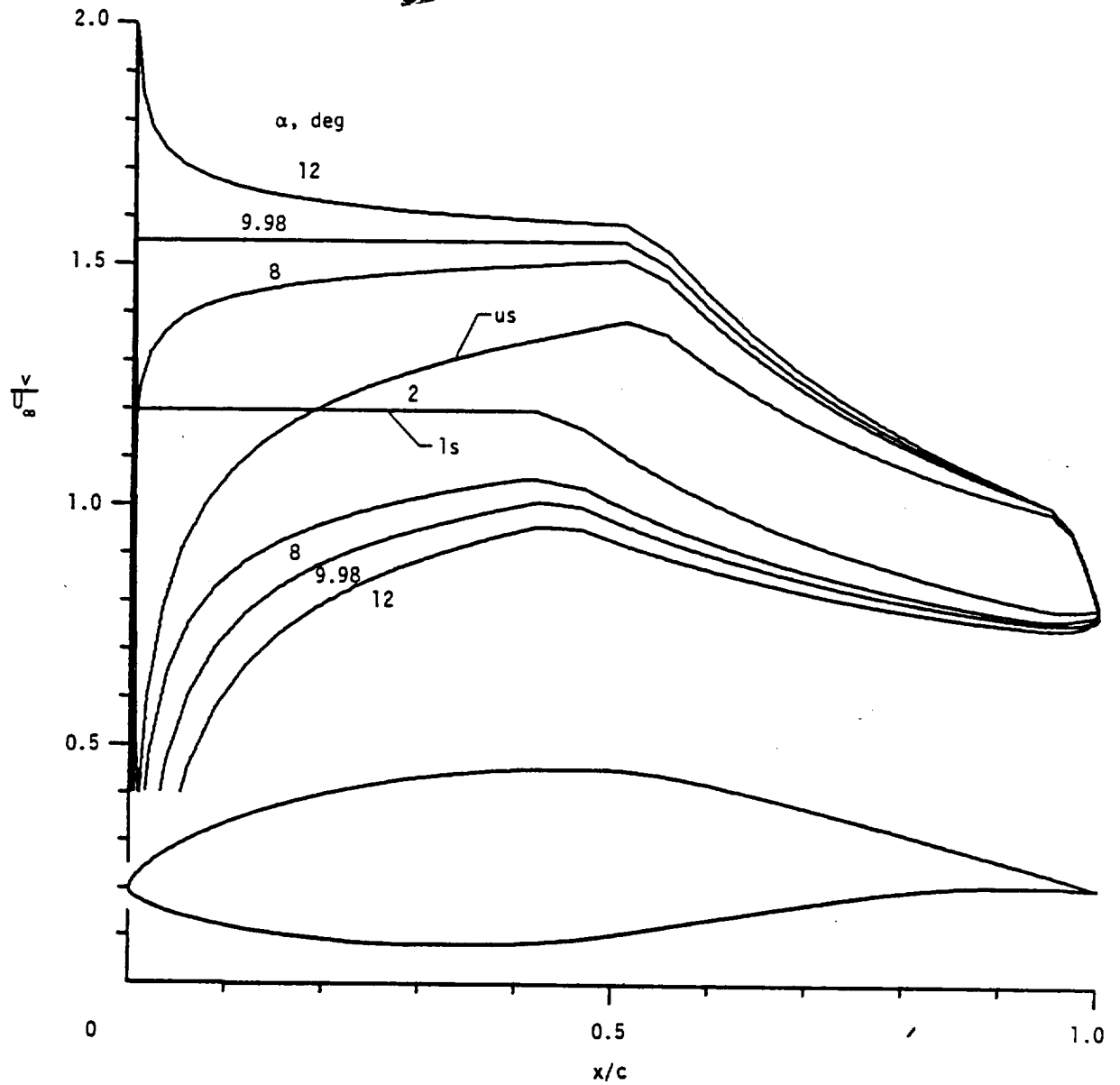
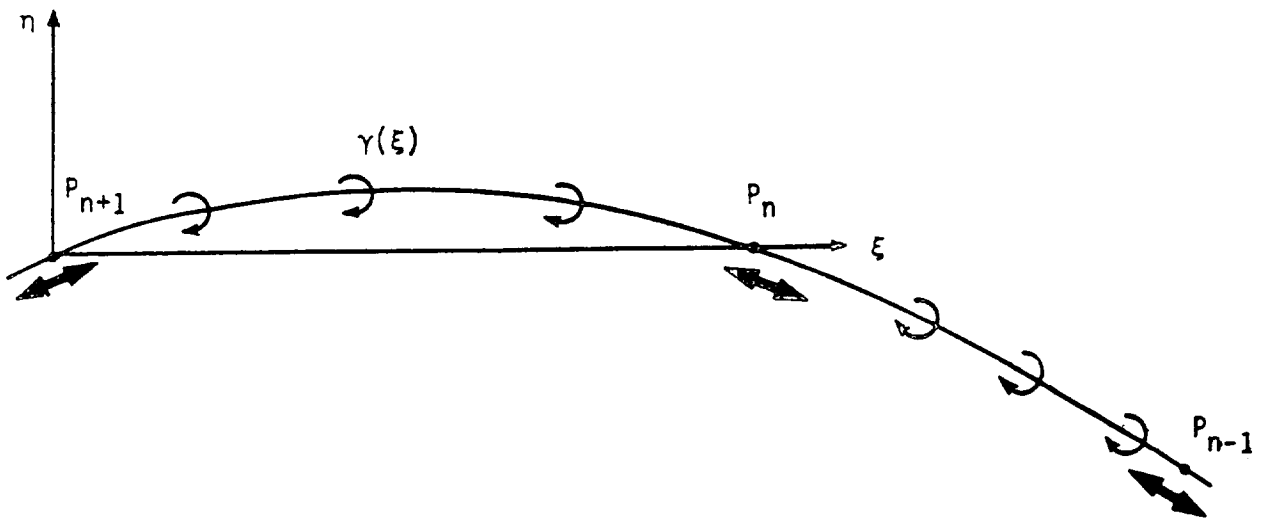


Figure 7.- Airfoil 1097. (α relative to zero-lift line.)



Panel between P_n and P_{n+1}
 Shape: Cubic in $n(\xi)$
 Vorticity: Parabolic
 Flow condition: Inner tangential velocity
 equals zero at P_n and P_{n+1}

Figure 8.- Panel method.

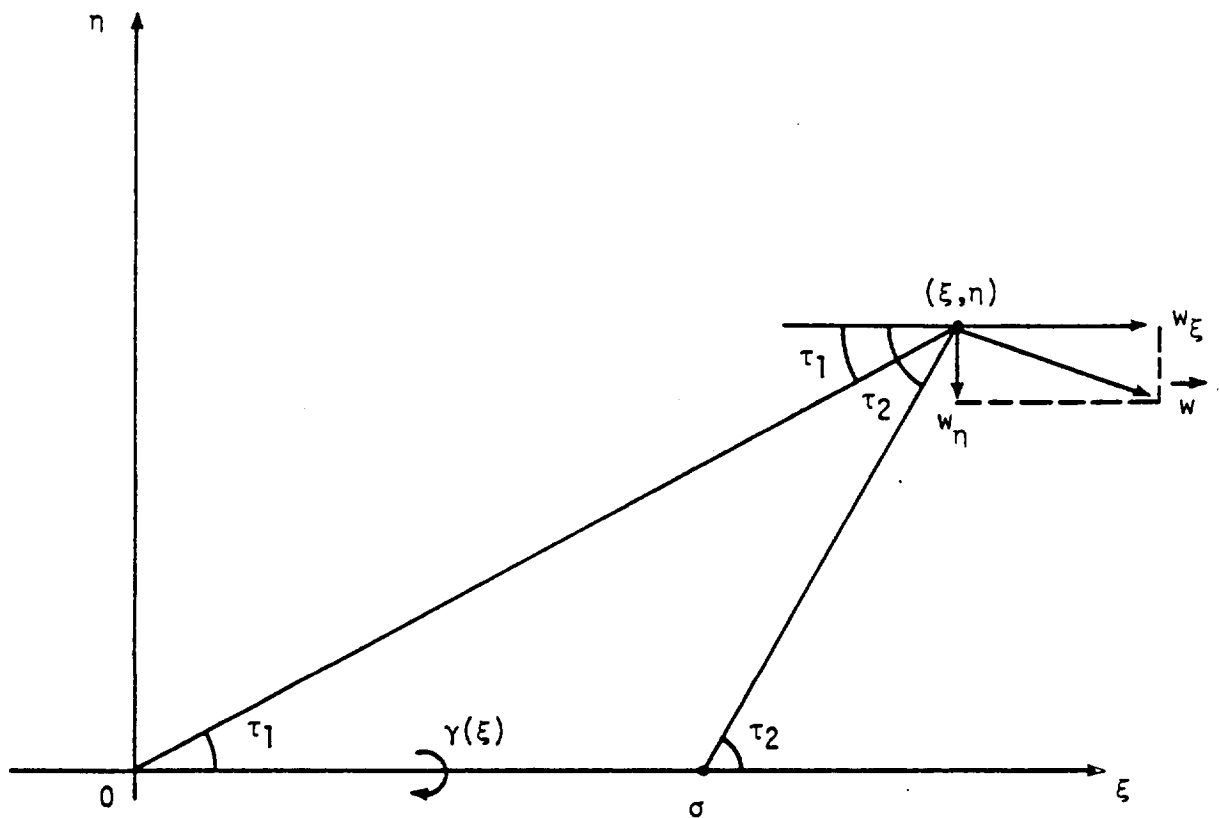


Figure 9.- Induced-velocity vector.

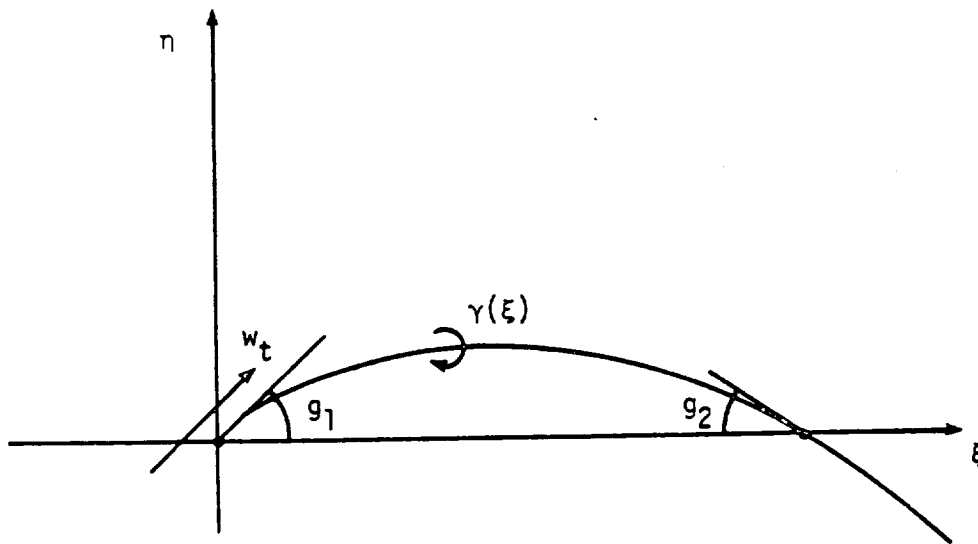
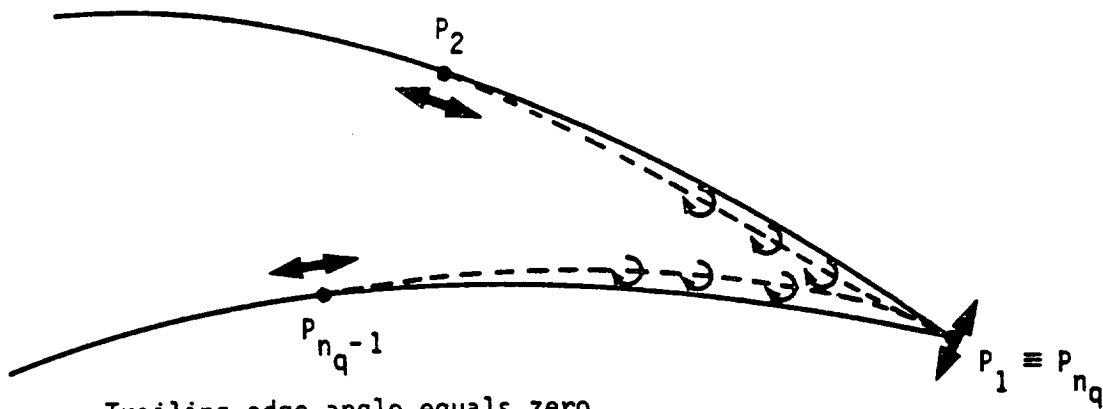


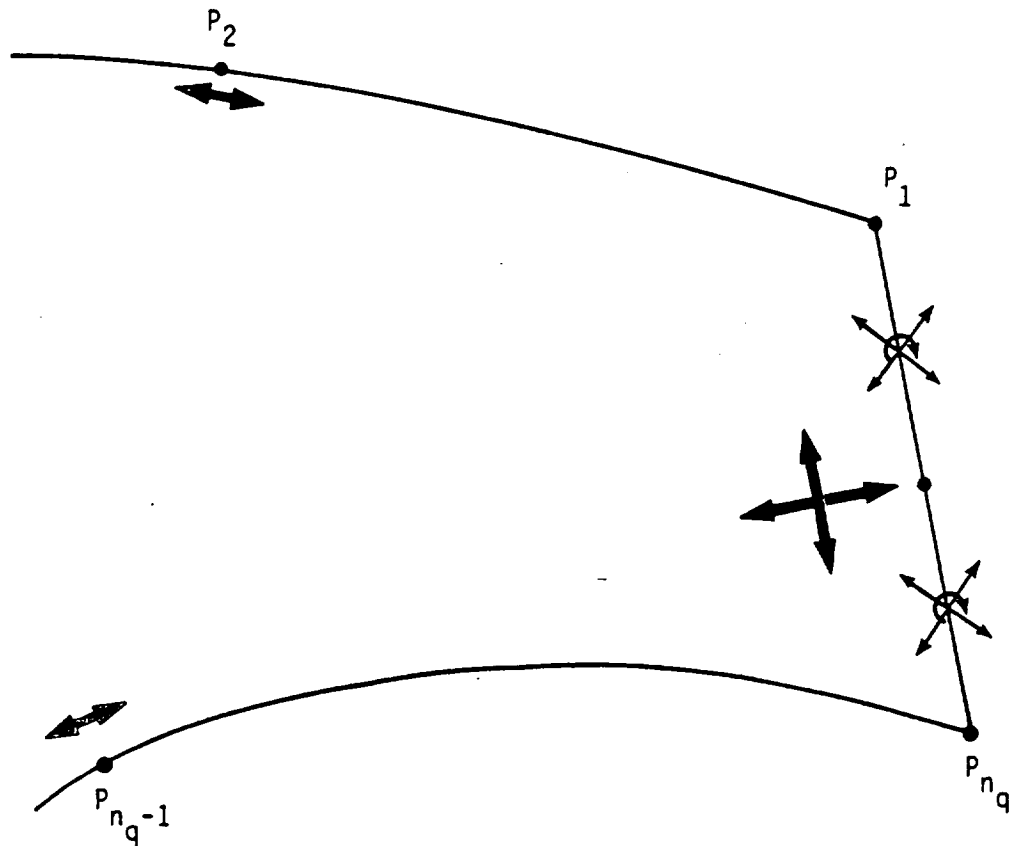
Figure 10.- Induced tangential velocity on panel.



Trailing-edge angle equals zero
 Flow conditions: Equal velocity on both sides of
 trailing edge; normal velocity equals
 zero at trailing edge

Figure 11.- Sharp trailing edge.

ORIGINAL PAGE IS
OF POOR QUALITY



Linear vorticity and source distributions at the base
No flow singularity at P_1 or P_{nq}

Flow conditions: Inner tangential and inner normal
velocities equal zero at the middle
of the base

Figure 12.- Blunt trailing edge.

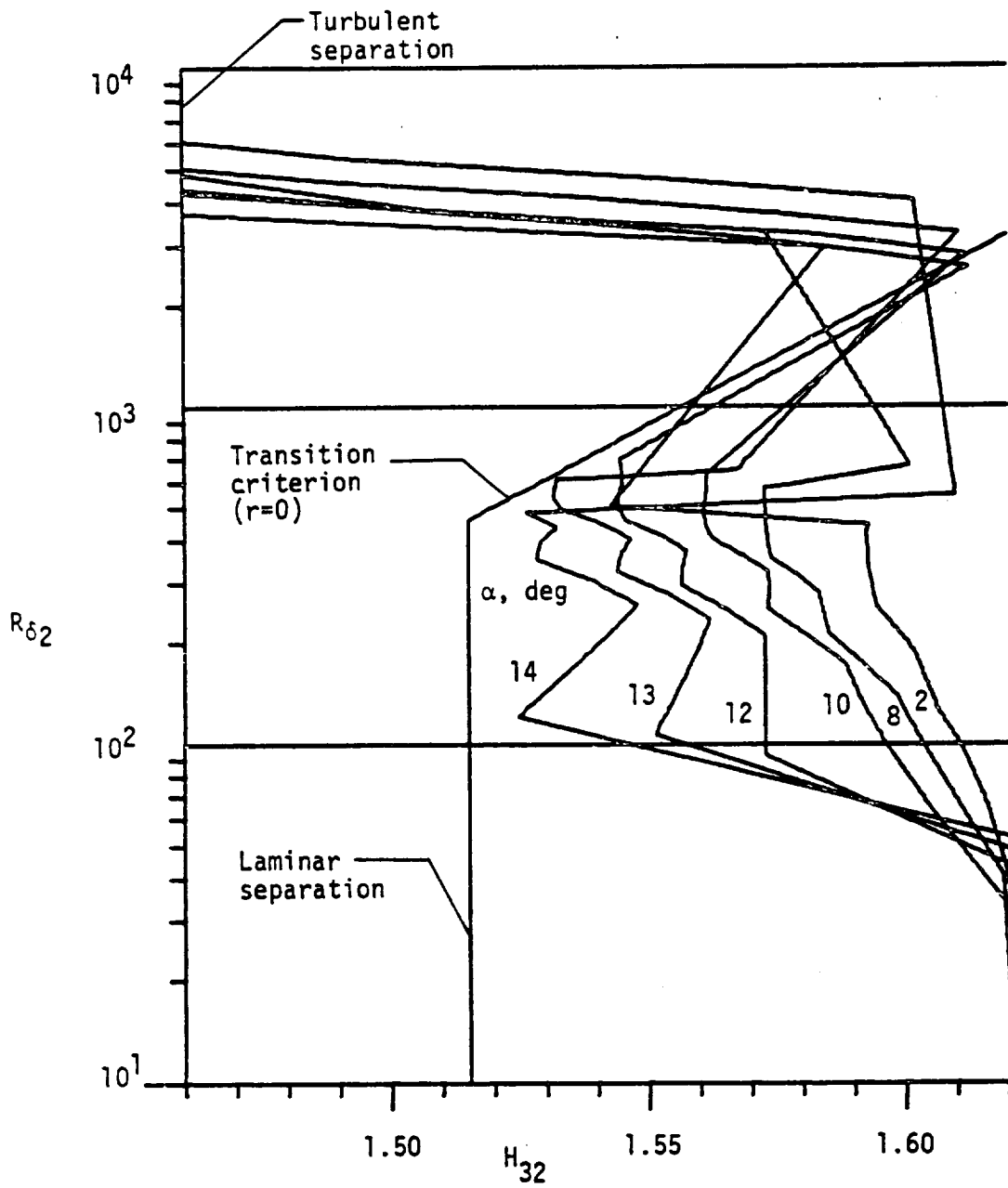


Figure 13.- Boundary-layer development; airfoil 1098, upper surface, $R = 1 \times 10^6$. (α relative to zero-lift line.)

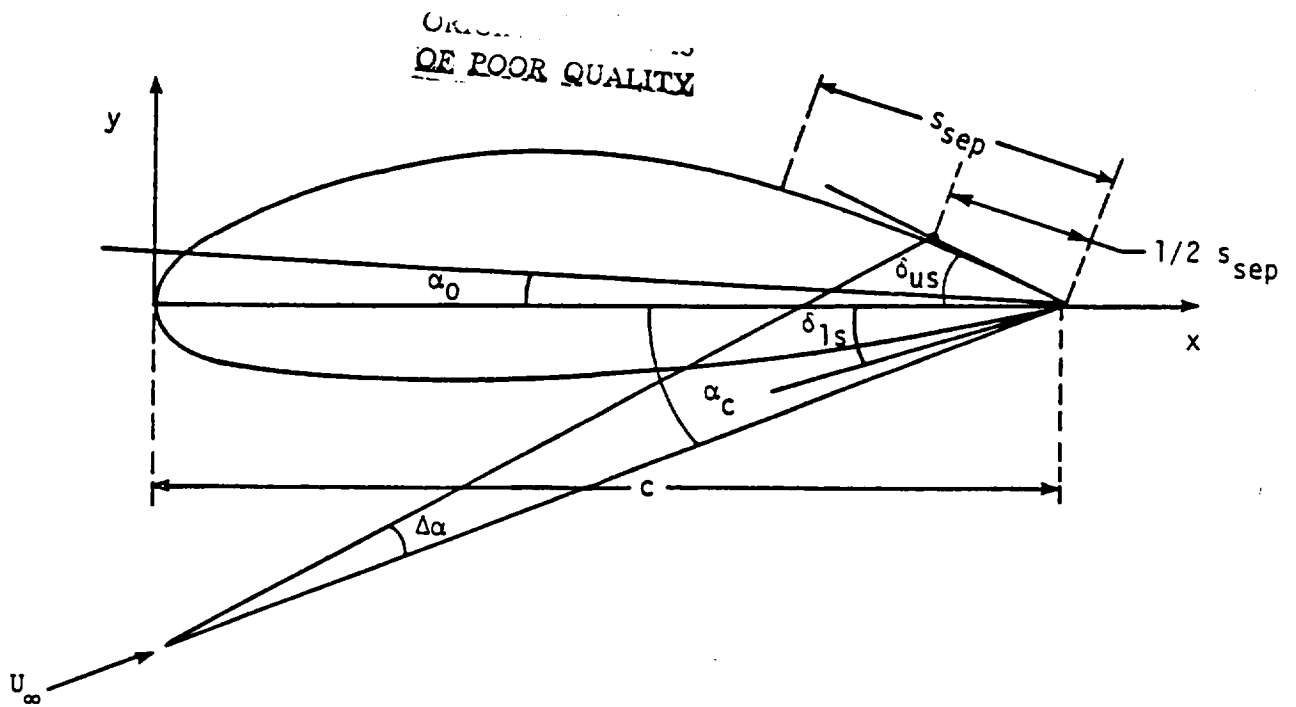


Figure 14.- Lift-coefficient correction due to separation.

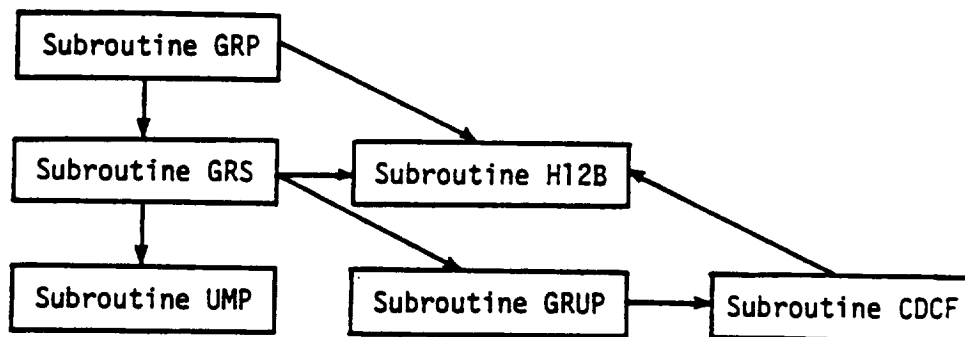


Figure 15.- Block diagram of boundary-layer subroutines.

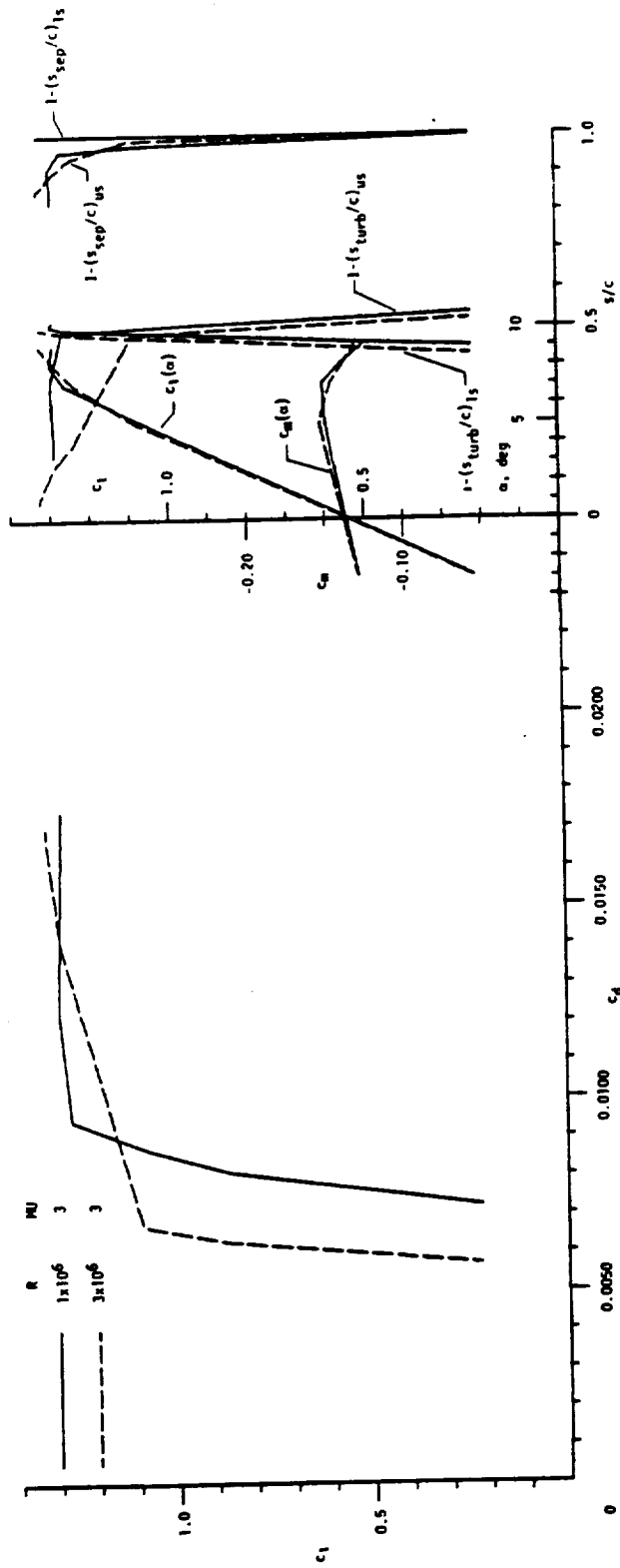


Figure 16.- Section characteristics for airfoil 1098.

