NASA TECHNICAL NOTE

NASA TN D-7718

NASA TN D-7718

(NASA-TN-D-7718) AN ANALYSIS METHOD FOR TWO-DIMENSIONAL TRANSONIC VISCOUS FLOW (NASA) 119 p HC $5.25 CSCL 01A

N75-18179

Unclas

H1/02 12351

AN ANALYSIS METHOD FOR TWO-DIMENSIONAL TRANSONIC VISCOUS FLOW

Paul C. Bavitz

Langley Research Center
Hampton, Va. 23665

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JANUARY 1975

1. Report No.
NASA TN D-7718

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
AN ANALYSIS METHOD FOR TWO-DIMENSIONAL TRANSONIC VISCOUS FLOW

5. Report Date
January 1975

6. Performing Organization Code

7. Author(s)
Paul C. Bavitz

8. Performing Organization Report No.
L-9708

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, Va. 23665

10. Work Unit No.
501-06-05-07

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D.C. 20546

13. Type of Report and Period Covered
Technical Note

14. Sponsoring Agency Code

15. Supplementary Notes
Author is employed at Grumman Aerospace Corporation; work done as Industry Research Associate at Langley Research Center.

16. Abstract

A method for the approximate calculation of transonic flow over airfoils, including shock waves and viscous effects, is described. Numerical solutions are obtained by use of a computer program which is discussed in the appendix. The importance of including the boundary layer in the analysis is clearly demonstrated, as well as the need to improve on existing procedures near the trailing edge. Comparisons between calculations and experimental data are presented for both conventional and supercritical airfoils, emphasis being on the surface pressure distribution, and good agreement is indicated.

17. Key Words (Suggested by Author(s))
Two-dimensional transonic flow
Viscous flow analysis
Compressibility effects

18. Distribution Statement
Unclassified – Unlimited

STAR Category 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	117	$5.25

*For sale by the National Technical Information Service, Springfield, Virginia 22151

AN ANALYSIS METHOD FOR TWO-DIMENSIONAL TRANSONIC VISCOUS FLOW

By Paul C. Bavitz[1]
Langley Research Center

SUMMARY

A method for the approximate calculation of transonic flow over airfoils, including shock waves and viscous effects, is described. Numerical solutions are obtained by use of a computer program which is discussed in the appendix. The importance of including the boundary layer in the analysis is clearly demonstrated, as well as the need to improve on existing procedures near the trailing edge. Comparisons between calculations and experimental data are presented for both conventional and supercritical airfoils, emphasis being on the surface pressure distribution, and good agreement is indicated.

INTRODUCTION

Higher cruise speeds with improved transonic performance are currently being demanded of new aircraft designs. A fundamental requirement in achieving these goals is the development of an effective method for the aerodynamic analysis of transonic airfoil sections, including viscous effects. The approach most widely used to account for viscous effects in an analysis was first suggested by Prandtl. The boundary-layer displacement thickness is added to the original geometry and produces an equivalent inviscid shape which represents the displacement of the inviscid-flow streamlines by the boundary layer. This procedure has been successfully applied to compute incompressible viscous flows with compressibility corrections (ref. 1, for example), but has never been incorporated with a fully compressible inviscid analysis, including embedded shock waves, as is herein attempted. (Similar efforts are, however, currently underway at the Courant Institute of New York University.)

The requirement for a viscous transonic analysis stems from the dramatic effects that the boundary layer has on shock location and lift in transonic flow, as compared with the innocuous results it produces in incompressible flow. Unfortunately, the region controlling most of these effects is near the trailing edge, where no theory yet exists to model the correct behavior of the flow. Because Prandtl's method is not valid, in general, near

[1]Grumman Aerospace Corporation; work done as Industry Research Associate at Langley Research Center.

a separation point, empirical procedures are required to generate the effective inviscid shape near these regions. Recent advances in both inviscid and boundary-layer analysis methods for compressible flow (ref. 2) however prompted this somewhat crude attempt at a combined solution. Specifically, this report defines an engineering technique, in the form of a computer program, to predict airfoil performance at transonic speeds; the method is a combination of existing analytical methods which have been modified by empirical formulations where necessary.

SYMBOLS

a	speed of sound
a_1,G,L	empirical functions in boundary-layer analysis
C_p	pressure coefficient
c	airfoil chord
c_d	section drag coefficient
c_l	section lift coefficient
c_n	section normal-force coefficient
H	shape factor
K	factor defined by equation (7)
M	Mach number
M_{dd}	drag-divergence Mach number
p	pressure
$q^2 = u^2 + v^2$	
R	Reynolds number
r,θ	polar coordinates

r_1	recovery factor
t_1	constant such that $\frac{t_1 r_1}{a_1} \approx 7$
$U, V, \tilde{\tau}$	variables (see eq. (5))
u, v	velocity components in x,y directions
x, y	Cartesian coordinates
x_s	shock location
α	theoretical angle of attack
γ	ratio of specific heats
δ^*	displacement thickness
ϵ	rate of dissipation of turbulent kinetic energy per unit mass
ρ	density
τ	shear stress
Φ	modified velocity potential
ϕ	velocity potential
ω	modulus of derivative of map function

Partial differentiation is implied when x, y, r, or θ appear as subscripts. Overbars and primes denote mean and fluctuating quantities, respectively, in the boundary-layer analysis, and their absence indicates instantaneous (mean plus fluctuating) quantities.

THEORETICAL FORMULATION

General Discussion

The overall problem of defining the viscous aerodynamic characteristics of transonic flow past airfoils is divided into three broad areas: inviscid solution, boundary-layer solution, and combined iterated solution. The inviscid method consists of an iterative, finite-difference, numerical solution of the exact potential equation

$$\left(a^2 - \phi_x^{\,2}\right)\phi_{xx} - 2\phi_x\phi_y\phi_{xy} + \left(a^2 - \phi_y^{\,2}\right)\phi_{yy} = 0 \tag{1}$$

for two-dimensional, steady, irrotational flow. The boundary-layer analysis models only turbulent flow and numerically solves a system of equations consisting of the mean momentum equation, the mean continuity equation, and the turbulent energy equation. This system, after simplification, can be expressed as

$$\left.\begin{aligned}
&(\bar{\rho}\bar{u})_x + \left(\bar{\rho}\bar{v} + \overline{\rho' v'}\right)_y = 0 \\
&(\bar{\rho}\bar{u})\bar{u}_x + \left(\bar{\rho}\bar{v} + \overline{\rho' v'}\right)\bar{u}_y = -\bar{p}_x + \tau_y \\
&(\bar{\rho}\bar{u})\left(\frac{1}{2}\,\overline{q^2}\right)_x + \left(\rho v + \overline{\rho' v'}\right)\left(\frac{1}{2}\,\overline{q^2}\right)_y = \tau\bar{u}_y - \left(\overline{p' v'} + \frac{1}{2}\,\bar{\rho}\,\overline{q^2 v'} + \frac{1}{2}\,\overline{\rho' q^2 v'}\right)_y \\
&\qquad\qquad - \rho\epsilon - \frac{t_1 r_1(\gamma - 1)M^2\tau(\tau/\bar{\rho})_y}{\bar{u}a_1}
\end{aligned}\right\} \tag{2}$$

The combined solution employs an iteration scheme to link these two analyses. This procedure is initiated by computing the inviscid pressure distribution about the physical airfoil and then describing a boundary layer based on this distribution. The boundary-layer displacement thickness, adjusted by the empiricism mentioned in the introduction, is added to the airfoil to produce an equivalent inviscid shape. Then a new inviscid calculation is performed. The resulting pressure distribution is used to redefine the boundary layer, and so forth, until the iterations converge. An illustration of the fundamental features is shown in figure 1.

The individual analysis methods were selected from several available computational schemes not only by examining their capability to correlate with data but also by examining any specific characteristics which would be desirable in the combined solution. For example, the boundary-layer analysis method provides excellent correlation with the experimental displacement thickness on the upper surface of an airfoil forward of about 90 percent chord and poor correlation near the trailing edge. Although it underpredicts the displacement thickness in this region, it is consistent in doing so. By recognizing the inherent failings of any boundary-layer analysis near the trailing edge, particularly on a supercritical airfoil, and by realizing that some modification will be necessary, it becomes obvious that consistent results can only be derived by starting with a method which is, in some sense, consistent. Only brief descriptions of the inviscid and boundary-layer analytical methods are included in this paper; a more detailed explanation can be found in the appropriate references. However, the combined solution, the empiricism, and the requirements for convergence are discussed in detail in the next section.

Inviscid and Boundary-Layer Analyses

The method used to analyze the inviscid flow was developed by Garabedian and Korn (ref. 3) and it implements a rapidly convergent transonic finite-difference scheme defined in reference 4. A similar analysis was also developed by Jameson (ref. 5). The coordinate system, suggested by Sells in reference 6, consists of mapping the interior of the unit circle conformally on to the exterior of the airfoil with the point at infinity corresponding to the origin in the circle plane. The modulus of the derivative of the map function ω is calculated by using a finite-difference approximation based on uniform mesh sizes in r and θ. As an initial guess of the conformal mapping, a Fourier series is computed at equal intervals on the unit circle.

In this coordinate system, equation (1) becomes

$$\left(a^2 - r^2\omega^{-2}\phi_\theta^{\,2}\right)\Phi_{\theta\theta} - 2r^4\omega^{-2}\phi_\theta\phi_r\Phi_{\theta r} + r^2\left(a^2 - r^4\omega^{-2}\phi_r^{\,2}\right)\Phi_{rr} + \left[r^2\omega_\theta\omega^{-3}\left(\phi_\theta^{\,2} + r^2\phi_r^{\,2}\right) - 2r^3\omega^{-2}\phi_\theta\phi_r\right]\Phi_\theta + \left[r^4\omega_r\omega^{-3}\left(\phi_\theta^{\,2} + r^2\phi_r^{\,2}\right) + r\left(a^2 + r^2\omega^{-2}\phi_\theta^{\,2} - 2r^4\omega^{-2}\phi_r^{\,2}\right)\right]\Phi_r = r\omega^{-3}\left(\phi_\theta^{\,2} + r^2\phi_r^{\,2}\right)\left[\omega_\theta \sin(\theta + \alpha) + r\omega_r \cos(\theta + \alpha)\right] \tag{3}$$

where the singularity at the origin is removed by the substitution

$$\Phi = \frac{\phi - \cos(\theta + \alpha)}{r} \tag{4}$$

The circulation is determined at $\theta = 0$ by using the Kutta-Joukowski condition; that is, the velocity at the trailing edge is required to be continuous. A finite-difference scheme related to that of reference 4 is used to solve equation (3) in a uniform grid with mesh sizes Δr and $\Delta\theta$ over the range $0 \leqq \theta \leqq 2\pi$ and $0 \leqq r \leqq 1$. Successively refined grids are employed because (1) the major features of the flow, especially the circulation and shock location, are well approximated on a coarse grid so that the coarse solution provides good initial conditions for the next refinement, and (2) the asymptotic convergence rate slows down tremendously with decreasing mesh size so that improved initial estimates for the potential field are extremely desirable. Also, an artificial viscosity parameter is introduced in an attempt to decrease the truncation errors, which are otherwise first order at supersonic flow field points. (It should be noted that this parameter applies only to the inviscid scheme, and it has nothing to do with the boundary-layer effects in the overall solution.) A sufficient amount of artificial viscosity guarantees that the correct entropy inequality is imposed on the solution. Shock waves arise naturally, and the process is continued until the desired level of convergence is attained.

The boundary-layer analysis method was developed by Bradshaw and others. (See refs. 7 and 8.) It is based on the turbulent energy equation which is transformed into a differential equation for the turbulent shear stress by defining three empirical functions which relate the turbulent intensity, diffusion, and dissipation to the shear stress profile. The major hypotheses are (1) the usual boundary-layer approximation, which implies no static-pressure difference across the layer; and (2) the turbulence structure is essentially unaltered by compressibility, as suggested in reference 9. Furthermore, equations (2) are derived from the exact equations for compressible flow by using typical order-of-magnitude arguments, as described in reference 8. The accuracy of the method thus depends almost entirely on the definition of the empirical functions a_1, L, and G, and the final form of the equations is

$$\left.\begin{aligned}
&U_x\left[1 + r_1(\gamma - 1)M^2\right] + \frac{U_y V r_1(\gamma - 1)M^2}{U} + V_y + \frac{U\bar{p}_x}{\bar{p}} = 0 \\
&UU_x + \left[V - \frac{r_1(\gamma - 1)M^2\tilde{\tau}}{U}\right]U_y = -\frac{U^2\bar{p}_x}{\gamma M^2\bar{p}} + \tilde{\tau}_y \\
&\frac{U\tilde{\tau}_x}{2a_1} + \left[\frac{V}{2a_1} + \frac{t_1 r_1(\gamma - 1)M^2\tilde{\tau}}{Ua_1}\right]\tilde{\tau}_y = \tilde{\tau}\left[1 + \frac{G\tilde{\tau}_{max}^{1/2}\, r_1(\gamma - 1)M^2}{U}\right]U_y - \frac{\tilde{\tau}^{3/2}}{L} - \tilde{\tau}_{max}^{1/2}(G\tilde{\tau})_y
\end{aligned}\right\} \qquad (5)$$

for the variables U, V, and $\tilde{\tau}$ where $U \equiv \bar{u}$, $V \equiv \bar{v} + \frac{\overline{\rho' v'}}{\bar{\rho}}$, and $\tilde{\tau} \equiv \frac{\tilde{\tau}}{\bar{\rho}}$ just for this set of equations. The method of characteristics is used to obtain the solution, and the empirical data inputs are those suggested in reference 8. The empirical functions mentioned previously are solely related to the boundary-layer analysis and should not be confused with any empiricism necessary to combine the inviscid and viscous analysis methods.

CALCULATION PROCEDURE

The inviscid-flow and boundary-layer analysis methods discussed in the previous section use computer programs to determine numerical solutions. The combined analysis is also accomplished by using a computer program whose basic structure is composed from these two separate programs. Specific details of the individual computational schemes can be found in references 10 and 11, whereas the important highlights of the overall calculation procedure follow.

First, the airfoil coordinates are processed by computing and smoothing the slopes and curvatures. The mesh size is set for a fine grid (160 points by 32 points) and the airfoil is conformally mapped onto the unit circle. Then the mesh size is adjusted to a crude grid (40 points by 8 points) and the inviscid-flow—boundary-layer iteration process is initiated by computing the inviscid flow about the physical airfoil. This solution normally requires 50 to 250 cycles to converge (throughout this discussion, "cycle" refers to a single sweep of the computational grid within the inviscid solution and "iteration" refers to one pair of completed inviscid-flow and boundary-layer solutions).

The next step is to define the boundary-layer characteristics. As input to the boundary-layer routine, pressure coefficients at 41 equally spaced points are specified on both the upper and lower surface; this number is independent of the grid size used for the inviscid computations. A spacing of 2.5 percent chord was selected because it seemed to best simulate the behavior of the boundary layer near a shock wave. Since there is no model in the analysis for the interaction of the shock wave and the boundary layer, the effects of the shock wave are only accounted for in the purely classical sense. A denser definition of the pressure distribution, for example, every 1 percent chord, would yield shock-induced separation for conditions which do not warrant it; a sparser definition, typically every 5 percent chord, would not accurately produce the expected hump in the boundary-layer displacement thickness near the shock. Based on results to date, this procedure seems to be adequate and indicates that an interaction model might not be necessary.

Then, before the boundary layer is computed, the pressure distribution is modified by an empirical formulation. As previously mentioned, the boundary-layer displacement thickness is not accurately predicted near the trailing edge, probably because of the neglect of normal pressure gradients and near wake effects. In fact, it is not even clear that a displacement-type effect will produce the appropriate streamline curvature or that the displacement thickness is the correct parameter to use in defining the effective inviscid shape near the trailing edge (based upon discussions with R. E. Melnik of Grumman Aerospace Corporation). Therefore, since some adjustment is necessary, modifying the pressure distribution is easier than working with the displacement thickness itself. This procedure is implemented only to generate the input data to the boundary-layer routine, and the actual surface pressure distribution is still defined as the one computed by the inviscid analysis. Hence, the empiricism is primarily concerned with producing the appropriate effective inviscid shape near the trailing edge; and the resulting thickness distribution in this region, which is computed from the modified pressure distribution, may not precisely agree with a true definition of the boundary-layer displacement thickness although its general character is still preserved.

The modifications to the pressure distribution were initially developed for supercritical airfoils but work equally well for conventional airfoils. First, the maximum pressure coefficient on the aft portion of the lower surface is held constant from its chord location to the trailing edge; forward of this point, the surface pressure distribution is unmodified. On the upper surface, the most aft point at which the surface pressure distribution is used is determined by an empirical equation based on trailing-edge slope. Then, a second-degree polynomial variation for the pressure distribution is described by using the pressure at this last point, the point before it, and the new trailing-edge pressure value from the lower surface. The trailing-edge slope used to compute the deviation point from the upper surface pressure distribution is a weighted average of the upper and lower surface trailing-edge slopes, two-thirds and one-third, respectively. This deviation point is designated IFIX in the computer program, and the variation of IFIX with slope is shown in figure 2. Approximately 12 airfoils have been analyzed in order to generate this variation. The integer value is used; that is, if 39.7 is computed, the last usable surface pressure distribution point is the 39th (95 percent chord). Figure 3 illustrates typical modifications to the pressure distribution for airfoils with small, medium, and large amounts of aft camber. As the aft camber increases, the deviation point moves forward. In a crude sense, this movement is consistent with an intuitive picture of the region in which there are near-wake influences and in which the basic assumptions about the boundary-layer break down; for example, normal pressure gradients might be introduced as a result of the increased curvature. Whenever a rigorous solution is obtained for the problem of the interaction of inviscid flow, boundary layer, and the near wake, this empirical model can be replaced or modified.

The boundary-layer characteristics are therefore calculated by using a modified pressure distribution; and, obviously, the problem of separation must be confronted. At transonic speeds, most airfoils have a small separated zone on the upper surface near the trailing edge and, for supercritical airfoils, occasionally on the lower surface in the cove region. Empirical definitions for the displacement thickness in these areas have therefore been included in the calculation procedure. No attempt has been made to reckon with shock-induced separation. However, the analytical method apparently produces a reasonable definition of the boundary for incipient shock-induced separation, as described in the next section.

On the upper surface, the slope of the equivalent airfoil at the separation point is maintained constant to the trailing edge. This requirement is imposed before the equivalent airfoil is processed so that anomalies in the curvature are resolved by the smoothing routine. For conventional airfoils, the same is done on the lower surface. Supercritical airfoils, however, have a favorable pressure gradient near the trailing edge on the lower surface; and if separation occurs, it usually is located slightly forward of the cove. Furthermore, separation is often predicted there only on the first boundary-layer calculation, and subsequent iterations are unseparated. This feature is a result of the relieving effect that the boundary layer has on the pressure distribution; therefore, pressure gradients are higher when the boundary layer is not yet taken into account. Therefore, a model was developed simply to allow the iteration process to continue without intending to cope with cases where the cove separation was still present in the final analysis. Of course, a representative shape for the displacement thickness was assumed in order to minimize the number of iterations. However, cases analyzed to date indicate data correlation is good even when the cove separation persists. A third-degree polynomial variation defines the displacement thickness by using stations at (1) 10 percent chord before the separation point, (2) 8 percent chord before the separation point, (3) midway between the trailing edge and the separation point, and (4) the trailing edge. The calculated values of displacement thickness are used at the first two stations. At the third station, an increment to the first value is derived from an empirical equation based on the difference in the pressure coefficients between the stations. The average of the first and third values is used at the trailing edge. When the lack of sensitivity of the pressure distribution to the geometry in this region is considered, the correlation actually is not surprising. The most arbitrary values in this model, those near the third station, are defining the y-coordinates of the equivalent inviscid shape in the region of the cove where the slope is approximately zero. The variations in the pressure distribution with small geometric perturbations in this area are insignificant. Therefore, as long as the magnitudes are reasonably appropriate, qualitatively reproducing the general shape seems to be sufficient. This fact is further substantiated when the effects of smoothing the airfoils are examined. Although the coordinates are accurately maintained over most of the airfoil, there is a tendency for the cove region to be filled in slightly; yet, correlation with experimental results is not affected by these changes. In view of the arbitrary nature of this procedure, however, results with extensive separation on the aft part of the lower surface should still be treated with caution. Occasionally, separation near the leading edge is predicted for early iterations, and the boundary-layer displacement thickness is temporarily defined as zero over the entire corresponding surface.

Once the boundary-layer displacement thickness is completely defined on both surfaces, it is added to the physical airfoil normal to the surface and thereby produces an equivalent inviscid airfoil. The resulting shape is processed as was the original airfoil.

The mesh size is adjusted to the fine grid for the mapping process and then changed back to the crude grid for the next inviscid-flow computation. Smoothing the airfoil at this stage is very important because the inviscid computation is more sensitive than the boundary-layer analysis. Minute wiggles in the displacement thickness, which are therefore present in the effective inviscid airfoil, could produce oscillations in the pressure distribution which would hinder the convergence of the overall analysis.

A new inviscid solution is then computed by using the new geometry. The resulting surface pressure distribution is again modified, a boundary-layer displacement thickness is calculated, and the effective inviscid shape is redefined. This process is repeated until a stabilized result is obtained. In order to make the procedure as automatic as possible and to determine solutions quickly and inexpensively, some convergence criteria must be included, for it can hardly be expected that both the inviscid-flow and boundary-layer equations will be exactly satisfied simultaneously. Two types of criteria are employed, one physically and one computationally oriented, and either one or the other must be satisfied. The most stringent criterion would be to require the coefficient of pressure at each calculation point to remain approximately the same from one iteration to the next. However, simply imposing a small increment within which this difference must remain is not adequate since the pressure distribution is apt to contain a shock. A shift in shock location of only 2 percent chord from one iteration to the next might be tolerable; the pressure coefficients at a station between these shock locations however could differ by 0.5. The criterion must therefore consider the local gradient in the pressure distributions in establishing an acceptable tolerance, and the form currently used is

$$\left| \left(C_p\right)_{new} - \left(C_p\right)_{old} \right| \leq 0.025 + 0.0005K \tag{6}$$

at each calculation station where

$$K = \mathrm{Max}\left\{ \left[\frac{\left(\Delta C_p\right)_{new}}{\Delta(x/c)}\right]_+^2 ; \left[\frac{\left(\Delta C_p\right)_{new}}{\Delta(x/c)}\right]_-^2 ; \left[\frac{\left(\Delta C_p\right)_{old}}{\Delta(x/c)}\right]_+^2 ; \left[\frac{\left(\Delta C_p\right)_{old}}{\Delta(x/c)}\right]_-^2 \right\} \tag{7}$$

"New" refers to the current pressure distribution and "old" refers to the pressure distribution of the previous iteration. The local pressure gradients at a particular point are simply calculated from the pressure coefficient and x-coordinate at that point and the corresponding values at the computational grid points immediately on either side, as indicated by the plus and minus subscripts. At most stations the local gradient is less than 4.0 and so the pressure distribution is basically required to repeat within a level of 0.025 everywhere. Additionally, the lift coefficient must repeat within 0.02. This condition on the lift helps to limit the acceptable shock travel from one iteration to the next, for a shift in shock location is manifested in a change in lift coefficient. Hence, the convergence criteria associated with physical parameters are comprised of these two requirements.

Alternately, a second type of test is included to measure the convergence of the overall process, and it is associated with the computational cycles of the inviscid analysis itself. The initial inviscid calculation starts from incompressible flow conditions. The remaining inviscid calculations always start from the last flow-field definition of the previous inviscid solution; that is, they are starting from a converged inviscid solution but with a different geometry. As the inviscid-flow—boundary-layer iteration process converges, the changes to the effective airfoil shape diminish; and eventually, the starting point for a given set of inviscid calculations will be very close to the end point. Therefore, the number of computational cycles for the inviscid analysis has been used to measure the overall convergence level. By excluding the initial inviscid solution, therefore, a solution which requires 20 computational cycles or less implies convergence. This formulation emerged because the other set of criteria occasionally did not indicate convergence for cases where consecutive pressure distributions appeared to be almost identical and correlated well with experimental data. This condition occurred when the difference in pressure coefficients from one iteration to the next did not remain within the specified tolerance at only one or two of the many computation points. Obviously, any set of convergence criteria must be somewhat arbitrary, and these have been chosen because they provide a good balance between quickness and accuracy. This subject is also discussed in the appendix as it applies to the use of the computer program.

Once the iteration process converges, the last inviscid solution is refined; that is, the inviscid flow-field definition with the crude grid (40 points by 8 points) is used as a starting solution for calculations employing a medium size grid (80 points by 16 points). There is no boundary-layer calculation between these two inviscid analyses, but the pressure distribution will change slightly. For example, if it contains a shock wave which is spread over two mesh points, the width in x/c for the medium grid will be half the width for the crude grid; thereby, a stronger shock jump is produced. Therefore, the iteration process is reinitiated by using this new inviscid solution as the starting point. All other facets of the analysis procedure remain the same. Once these iterations converge, a fine grid (160 points by 32 points) is introduced in a similar manner. When this final set of calculations converges, the process is terminated and the last pressure distribution represents the overall solution. These grid refinements are necessary because a solution with a coarse grid is not accurate enough (although it is desirable to start with a coarse grid for the reasons mentioned in the previous section). Limits are placed on the number of iterations performed with each of the grids in case convergence is not attained. They are set at six, four, and three iterations with the crude, medium, and fine grids, respectively.

This procedure allows a maximum of seven, five, and four inviscid analyses and six, four, and three boundary-layer analyses because each set always starts and ends with an inviscid solution. The number of computational cycles for the inviscid analysis itself is also limited to 800, 500, and 300 on the crude, medium, and fine grids, respectively. If any limit is reached for an intermediate calculation step, that particular step is terminated, but the overall analysis still continues with the next step in a normal fashion. If convergence is not indicated after six iterations by using the crude grid, for example, the seventh inviscid solution is still refined and iterations using the medium grid are started. However, if the limit is reached when iterating with the fine grid or if the intermediate inviscid calculations diverge, the process is terminated. Interpretation of results for cases such as these is discussed in the appendix.

The introduction of the crude and medium grids for the inviscid analysis, combined with the requirement that these intermediate inviscid-flow—boundary-layer iterations converge, helps to reduce total computer time. Initially, when the pressure distribution is not accurate because the boundary-layer effects are not fully sensed, only 320 computation points are used rather than 5 120. When the fine grid is finally employed, the inviscid solution is easier to attain because the boundary layer has driven the shock wave forward on the airfoil and reduced the amount of lift. Thus, not only are the inviscid-flow—boundary-layer iterations with the fine grid minimized because the intermediate iterations have converged, but also the number of computational cycles for a particular inviscid analysis with the fine grid is small. Most solutions are attained in 5 to 10 minutes of computer time on the Control Data 6600 series computer.

To illustrate the effects of the boundary layer on the pressure distribution, two sets of sample calculations have been made. First, the result of inviscid and viscous analyses are compared in figure 4 for both a subcritical and supercritical case. At the low Mach number, the effects are minimal. This fact is consistent with the past practice of ignoring the boundary layer in evaluating airfoils. However, at supercritical speeds with embedded shock waves, the effects of the boundary layer are dramatic. The inviscid prediction overestimates the lift coefficient by 40 percent (75 percent is not uncommon) and the characteristics of the pressure distribution are very different. These trends primarily stem from the reduction in effective aft camber which the boundary layer introduces by thickening the upper surface and filling the concave region on the lower surface of a typical supercritical airfoil. The second set of calculations illustrate the effects produced by varying the Reynolds number over a range from 2×10^6 to 2×10^8. (Obviously, any inviscid analysis can show trends with variations in Mach number and angle of attack, but the new parameter introduced in this combined analysis is the Reynolds number.) The case presented is for off-design conditions on a supercritical airfoil where substantial changes in the important parameters might be anticipated. Figure 5 shows the pressure distributions for eight different Reynolds numbers and figure 6 summarizes the trends in both the boundary-layer and external-flow characteristics by presenting the displacement thickness, shape factor, shock location, and lift coefficient. The corresponding inviscid solution is shown in figure 4(b). These trends, therefore, account for the interplay between the external flow and the boundary layer and show large variations, particularly at the low Reynolds numbers, as would be expected. They also demonstrate that the inviscid solution (corresponding to an infinite Reynolds number), at least for cases such as this, can be far removed even from a Reynolds number as high as 2×10^8. This condition is also discussed in the next section. The slight scatter in the calculation points arises because each individual analysis is only accurate within a band associated with the tolerances of the convergence criteria. Thus, trends must often be faired, as experimental data would be.

EXPERIMENTAL VERIFICATION

To discuss the results of the analytical method and to obtain an indication of its accuracy and capability, comparisons with experimental data are presented for several airfoil shapes, both supercritical and conventional. Primarily, the ability of the theory to predict the surface pressure distribution is evaluated. None of the airfoils herein analyzed was used in developing the empiricism in the method. Most of the computations for data correlation are made at one nominal Mach number for a series of angles of attack, although the actual tunnel Mach number may vary by ±0.001. The order of this deviation is only significant near a design point, and special note is made for these cases. Similarly, the nominal Reynolds number is used, but variations of this magnitude are insignificant. Since the aerodynamic angle of attack for two-dimensional wind-tunnel tests is rarely known, the comparisons are made at approximately the same lift. In order to accomplish the comparisons, computations at several arbitrarily selected angles were initially made, and an approximate lift curve was constructed. Then the angles necessary to produce solutions at lift values where data existed were defined by interpolation. Since there is some scatter in the analysis resulting from the tolerances on the convergence criteria, as discussed in the previous section, exactly identical lift values were rarely obtained. In general, comparisons between theory and experiment at nominally the same lift could have differences in the coefficients of up to 0.02. Thus, both the experimental normal-force coefficient and the theoretical lift coefficient are indicated in the comparisons because this small difference sometimes accounts for minor discrepancies between the pressure distributions. However, these differences are never substantial enough to warrant repeating the calculation at a new angle of attack.

The first airfoil evaluated is one designed by Korn using his inviscid complex-characteristics hodograph method. It was designed to be shockless at $M = 0.75$ with $c_l = 0.63$. It is approximately an 11.5-percent-thick supercritical airfoil whose characteristics are well documented in reference 12. Calculations are made over a wide range of Mach number and angle of attack, through drag divergence, with a Reynolds number of about 21×10^6. All these computations used the theoretical design coordinates (rather than the measured coordinates). Calibration studies of the facility which generated this data have produced an estimate for the effects of wall interference so that the geometric angle of attack can be corrected to the aerodynamic value. However, to be consistent with other comparisons, the theoretical results are still evaluated at the same lift as the data. An extensive analysis is presented in reference 13, this same data being used, in an attempt to account theoretically for these wall effects. Since that work employs the same type of inviscid solution as the method herein presented (except that the Kutta condition is not satisfied) and since it describes some details pertinent to data correlation in great depth, it is referred to throughout the following discussion.

Theoretical and experimental pressure distributions are compared in figure 7 for increasing angles of attack at $M = 0.512$. As indicated by the critical pressure coefficient (the long tick on the vertical axis), all these cases are for subcritical flow. The entire pressure distribution is accurately predicted, starting from the typical figure-eight shape at near-zero lift (fig. 7(a)) to the more evenly distributed lift variation at $c_l = 0.63$ (fig. 7(d)). The leading-edge peak, the central plateau, and the trailing-edge recovery are all reasonably defined for the four cases. Figure 8 compares pressure distributions at $M = 0.700$ for lift coefficients that range from slightly negative to almost 1.0. At the lowest lift, a very weak shock is present on the lower surface near the leading edge, and it is detected by the theory. The next case (fig. 8(b)) has a very slight peak near the nose on the upper surface and the correlation in this region is excellent. Although this case is for subcritical flow, minor irregularities in the pressure distribution such as this can adversely affect the overall transonic performance of an airfoil. As the angle of attack increases, the flow over the upper surface becomes supercritical and a well-defined shock wave develops. This trend is depicted in figures 8(c) to 8(e). The predicted shock location and the overall pressure distribution agree well with the data. However, some discrepancy occurs immediately behind the shock wave, as figure 8(e) aptly illustrates. This characteristic is present in all the computations with shocks. The difference in pressure levels in this region may be due, at least partly, to the failure of the inviscid solution to satisfy the correct jump condition rather than any interaction problem with the boundary layer. This is discussed in reference 13 which shows that neither the irrotational jump condition, appropriate to the potential flow equation, nor the Rankine-Hugoniot jump condition are matched. Thus, the pressure coefficient at the foot of the shock is not positive enough and this difference slowly diminishes over the next 20 percent to 30 percent chord.[2]

[2]After completing this note, a relaxation analysis routine for the full potential equation in conservation form was developed by Jameson, and his preliminary calculations indicate this formulation apparently accounts for the discrepancy just downstream of the shock.

The magnitude of the shock jump is also discussed in reference 14 by use of small-disturbance theory. This shortcoming of the inviscid calculation complicates the comparisons because they are made at constant lift. An angle of attack slightly smaller than otherwise expected is therefore required to compensate for the additional local lift behind the shock. This smaller angle results in a corresponding reduction in lift locally over the front of the airfoil, as figure 8(e) also depicts. A separation bubble behind the shock is just beginning to form for this case and it is obviously present in the next case shown in figure 8(f). Any discrepancies in the correlation in the region of the bubble cannot be explained by this interpretation; and probably, only an interaction method can detail that part. Still, the overall correlation is very good, particularly in the region near the trailing edge. Both sets of calculations just described were made at the nominal indicated tunnel Mach number which was not corrected for blockage effects.

A series of comparisons near the design Mach number are shown in figure 9 over a wide range in lift. Since the pressure distribution is very sensitive to Mach number at these conditions, a blockage correction of $\Delta M = -0.005$ suggested in reference 13 has been applied; that is, the indicated test Mach number for these angles is 0.757 ± 0.001. Therefore, the computations were made at $M = 0.752$ for all angles. Again, the correlation with the data is good. The characteristics of the upper surface pressure distributions are particularly interesting; as α increases, the shock waves start near the leading edge with a weak jump, evolve through a multiple system, and finally locate in an aft position with a large strength. (See figs. 9(c) to 9(f).) To validate the blockage correction, the calculations in figure 9(c) have been repeated at $M = 0.757$, and both results are shown in figure 10 with the corresponding results from reference 13. The effects produced by the shift in Mach number are almost identical for either analysis method. (Since the calculations in this report, for a series of angles, are made at one average Mach number, fig. 10(a) compares results at $M = 0.757$ and $M = 0.752$. For this particular data point, the indicated test Mach number is 0.758; therefore fig. 10(b) compares $M = 0.758$ and $M = 0.753$ because the precise Mach number at each data point was used. If the computations in figs. 10(a) and 9(c) were made at $M = 0.753$, the data correlation could only be enhanced.) The procedure outlined in reference 13 is solely aimed at providing the actual Mach number and angle of attack for wind-tunnel data so that indicated values can be corrected, and it requires the experimental pressure distribution to accomplish this. Since the data correlation for the analysis method herein presented is good, it too can be used, to some extent, to provide similar answers although this was not the original intent. The discrepancy behind the shock wave, and its corresponding effect on the angle of attack, does however presently limit this application. The Kutta condition at the trailing edge is not imposed in the method of reference 13, so the upper and lower surface pressure distributions cross near the trailing edge and produce a local region of decreased lift. This condition somewhat compensates for the increased local lift behind the shock and thus minimizes its effect on the angle of attack.

A final set of calculations has been made at $M = 0.782$, and these are presented in figure 11. Since the sensitivity to Mach number is greatly reduced because these conditions are not at the design point and since the precise blockage correction is not known, the indicated Mach number is again used. In general, the correlation with the data is good. However, the theory does fail to predict a very weak shock in the pressure distribution forward of the main shock. This inadequacy can be attributed to the mesh size; that is, the "fine" grid is not fine enough and an additional refinement would be necessary. Since this type of pressure distribution does not often occur, and the main shock and other characteristics are well defined, and since the cost of a computer run with an extra fine mesh would be prohibitive, it is not worth including further grid refinements in the analysis. Also, an examination of the sonic line will usually indicate the potential for such a wave because, as discussed in reference 13, an indentation in the sonic line appears in the general location of this weak shock and forecasts the division into two or more distinct supersonic zones.

Calculations at high transonic Mach numbers are presently limited by the ability of the initial inviscid solution to converge. Cases where the shock would be located about 10 percent to 20 percent chord forward of the trailing edge in the final analysis cannot be obtained because, without the boundary-layer displacement effect included, the initial solution would have the shock back at the trailing edge; and the present inviscid analysis method is unstable for this condition. Modifications are being made to the inviscid method, however, at the Courant Institute. Results however can currently be computed through drag divergence. In an attempt to provide temporarily a wider calculation envelope, the present analysis method automatically defines an arbitrary boundary-layer displacement thickness on the first cycle if the initial inviscid solution diverges. An effective inviscid shape is then prescribed, all other quantities are reinitialized, and the first inviscid solution repeated since it should now be less likely to diverge. However, the increase in attainable computation points has been disappointing. A minor error in this phase of the computer program is suspected, where everything is not properly reinitialized. This matter is also discussed in the appendix.

Although the correlation between theoretical and experimental pressure distributions is of prime importance, the prediction of the integrated forces and moments should also be examined. Normal-force values are relatively insensitive to the integration procedure and are routinely obtained. Rotation into the lift direction does not require precise knowledge of the angle of attack. However, comparisons between theoretical and experimental lift curves are usually meaningless. Generally, only the geometric angle is tabulated in the data without any correction for wind-tunnel wall interference, and the value for the aerodynamic angle from the theory is slightly tainted by the local overprediction of lift just behind a shock wave, as previously mentioned. Accurate pitching-moment information is usually insured by good correlation between pressure distributions because the location of the shock wave is the major contributing factor. The prediction of axial force and rotation into the drag direction has been and still is a problem. The absolute theoretical drag levels, estimated by adding the friction and pressure drags, are not very accurate. In fact, the precise origin of the wave drag in these types of analyses is not very clear. (An interpretation of the wave drag from isentropic shock waves is discussed in ref. 15.) Presently, the surface pressure distribution is integrated in the normal fashion to predict this drag.[3] Reference 15 suggests a shock integration method which has not yet been tried in this analysis because of the difficulty in accurately locating the shock wave in the flow field. Theoretical and experimental drag polars for the Korn airfoil are shown in figure 12 at several Mach numbers. The data points lie within a narrow band in Mach number for each curve and are labeled at the average value. Since reference 12 only tabulates this data, these polars were faired in order to examine drag creep and drag divergence. The variation in incremental drag with Mach number, from the subsonic level at $M = 0.512$, is therefore shown in figure 13 for several lift and normal-force coefficients. The correlation between the estimated and actual creep is only fair, in that the amount of creep near the divergence point is not known within the accuracy usually associated with drag. Of course, since this is only an incremental trend, and since the absolute level is not really known, precise definition of the amount of creep alone is not extremely useful. As the buildup in drag with increased Mach number becomes large, however, the theoretical and experimental trends agree very well and this agreement provides for an accurate definition of the drag divergence boundary. By using a slope criterion of $\partial c_d / \partial M = 0.1$, this boundary is summarized in figure 14, where the correlation between theory and data is excellent.

To evaluate the analysis method on a conventional airfoil, results for an NACA 64A410 are compared with data from reference 16 in figures 15 to 17. The coordinates tabulated in the data report are used in the computations. Unfortunately, the test Reynolds number is near 10^6, and a lambda shock pattern is present instead of a normal shock. Additionally, the transition is natural rather than fixed, whereas the analysis method only models a turbulent boundary layer. Figure 15(f) typically illustrates the discrepancy in the shock region resulting from the two-part compression associated with the lambda wave. At higher Reynolds numbers with turbulent boundary layers this situation would not occur. The general characteristics of the pressure distribution are accurately represented in all the cases. Even though the empiricism in the analysis method was initially introduced to cope with supercritical airfoil shapes, good correlation is achieved in the aft region of the conventional airfoil, where the geometry of the two differs the most.

---

[3]The integration of the pressure distribution obtained by using Jameson's new conservative difference scheme produces drag levels which better approximate the experimentally measured values.

The theoretical and experimental drag polars are again used to examine the drag creep and the drag divergence boundary. Figure 16 shows that the creep is defined slightly worse than it is for the Korn supercritical airfoil, and figure 17 shows the divergence boundary is again well represented.

The development of the analysis procedure made considerable use of data on the recent series of NASA supercritical airfoils. Although the ability to predict the pressure distributions for these airfoils would probably generate the most interest, these data are unpublished and so these correlations are not shown. However, a few pressure distributions on one of the early NASA supercritical airfoils have recently been presented in reference 17. Therefore, two cases have been computed by use of the measured model coordinates for this airfoil, and the resulting comparisons with the data are presented in figure 18. The subcritical computation illustrates two regions where the correlations are not quite in line, and this is typical of most of the comparisons for this class of airfoils. At the entrance to the cove region on the lower surface, the theoretical pressure coefficients are slightly more positive than the experimental values, most likely because of the presence of a small separation bubble in the actual flow which modifies the effective shape of the airfoil and thereby delays the start of the compression. On the upper surface, just forward of the trailing edge, the experimental pressures are inexplicably more negative, but this difference seems to be consistent. If any of these supercritical airfoils are to be modified for a new design point, the original shape should be analyzed, the results correlated with the data, and the predictions for the new geometry adjusted accordingly. This adjustment can be undertaken safely because the general characteristics should be closely estimated so that the adjustments are minor, and because the slight differences will normally remain consistent. For the supercritical flow case in figure 18(b), separation is predicted at 72 percent chord on the lower surface; yet the calculations proceeded by use of the empiricism discussed in the previous section. The predicted separation point was compared with the value from the Stratford criterion (ref. 18), and both locations are in perfect agreement.

To substantiate the Reynolds number variations shown in the previous section, calculations are compared with some unpublished data on a NACA 65_1-213 (a = 0.5) airfoil. Wind-tunnel data are rarely obtained at very high Reynolds numbers, but these measurements were made by using a model with a chord of 0.914 meter in the NAE facility in Ottowa, Canada (under NASA Contract NAS1-10632) which is the same facility as was used in reference 12. Figure 19 therefore shows the resulting pressure distributions over a range in Reynolds number from 25.0×10^6 to 52.6×10^6 for the nominal conditions of $M = 0.75$ and $\alpha = 0.0^\circ$. Although the Mach number and lift are not exactly the same at each point, the Reynolds number is the principal variable. The Mach number is corrected for blockage effects, and the theoretical computations use this value. The correlation in all cases is very good, particularly the shock location. For these conditions, there is not a high degree of sensitivity to Reynolds number. This result is further substantiated by examining the results of an inviscid analysis at the nominal conditions, where the shock wave is located at about 59 percent chord with $c_l = 0.29$. The computations in figure 19(c), closest to the nominal conditions, indicate a shock location of about 57 percent chord. The analysis method did, however, predict this insensitivity; and in view of the wide variations shown on the supercritical airfoil in the previous section, these correlations indicate that predicted trends are valid. This method should therefore be capable of providing answers for conditions which more closely resemble flight and which most wind tunnels cannot attain. In addition, the comparisons between the inviscid and viscous calculations for both airfoils indicate that a positive conclusion about the character of the flow at high Reynolds numbers (on the order of flight values) cannot be drawn simply from an inviscid analysis.

Finally, the ability to predict the shock-induced separation boundary has yet to be thoroughly examined. A few cases arose for the NASA supercritical airfoils where the inviscid-flow—boundary-layer iterations using the crude and medium grids for the inviscid computations would converge, but the refinement to the fine grid immediately yielded a pressure distribution with an adverse pressure gradient at the shock wave strong enough to separate the boundary layer. (As discussed in the previous section, the pressure gradient at the shock increases as the grid becomes finer because the shock jump is contained over a narrower range in x/c.) The angle of attack was reduced in small increments in order to define the angle below which this no longer occurred. When an empirical curve, based on data for a variety of airfoils, was used to check this point for these cases, remarkably close agreement with the present results was observed. (This curve typically shows the variation with Mach number of the pressure coefficient just prior to the shock wave for incipient separation, and it appears as a narrow data band.) Although any correlation between the present results and the incipient separation boundary would appear to be fortuitous, since a true shock—boundary-layer interaction is not modeled, it might be plausible. The spacing of the input points for the boundary-layer routine within the analysis is chosen solely to provide reasonable characteristics in the vicinity of the shock as described by the fine grid, but for cases where shock-induced separation is not present. The results just up to the demarcation point should therefore be valid, and it might be expected to reasonably indicate when the shock strength is such that separation is imminent. Of course, this still has to be validated over a wide range of conditions.

CONCLUDING REMARKS

The present analysis method, in the form of a computer program, has the capability to calculate surface pressure distributions at transonic speeds for both conventional and supercritical airfoils with viscous effects included. An automated iteration scheme was used to link separate, existing inviscid and boundary-layer analyses. Empirical formulations were included primarily to allow the computations to proceed when separation is present and to modify the boundary-layer displacement thickness near the trailing edge so that the effective inviscid shape in this region is more correctly defined. In an engineering sense, these formulations appear to be adequate.

The approximations to the surface pressure distribution have been shown to correlate reasonably well with experimental data for a variety of airfoil shapes over a wide range in Mach number, angle of attack, and Reynolds number. The shock was located within a few percent chord for almost every case analyzed, and good estimates of the pressure coefficients were obtained in the region of the trailing edge. Although the drag is not accurately predicted, the drag divergence boundary is well defined.

Several refinements are necessary to improve upon the existing version of the analysis method. The major factor requiring attention is the difference in theoretical and experimental pressure levels consistently observed immediately downstream of a shock wave. However, since improved estimates of the transonic performance of airfoils appear to result from almost any attempt to include the boundary layer in an analysis, the technique herein described is appreciably better than the current practice of using inviscid methods alone.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., December 3, 1974.

APPENDIX

A FORTRAN COMPUTER PROGRAM

The FORTRAN program which follows computes the pressure distribution about an airfoil by using the procedure outlined in the previous sections. It is written for the Control Data 6000 series computers. This program is basically composed of two other programs (refs. 10 and 11) which are separately used to analyze the inviscid flow and the boundary layer. These programs have been combined in a very crude manner by using the general format of the inviscid computational scheme; thus, the result probably contains some loose ends. The version tabulated in this report requires a field length of approximately 134 000_8 words (octal), of which 121 000_8 are required for execution. An overlaid version exists, which requires only 75 000_8 words. In the absolute binary mode, this field length is reduced to 66 500_8 words, and most of the computations are made with this version for obvious reasons. It is not shown, however, because it is difficult to follow.

Input and Output Data

The input consists of an identifying title, the airfoil coordinates, and the calculation points in terms of Mach number, angle of attack, and Reynolds number. The title and coordinates are copied, by means of the control cards, on to a disk file labeled TAPE3 to be read later by the main program. An end-of-record card is therefore necessary to separate this part of the input from the remainder. Then follows the aerodynamic information, with one card for each calculation point; multiple cases are automatically executed. All the input data, except the title card, use an F10.0 format. A maximum of 160 total coordinates can be specified, and the amount on one surface is limited to 100. The scaling for the x/c values is from 0.0 to 1.0. A summary of this input mode follows:

Title card – columns 2 to 59

Input control card – 3 fields
- Number of upper surface coordinates
- Number of lower surface coordinates
- Number of cases – automatically set to 1 if blank

Blank card or any desired label

Table of upper surface x/c values – 8 fields per card

Table of upper surface y/c values – 8 fields per card

Blank card or any desired label

Table of lower surface x/c values – 8 fields per card

Table of lower surface y/c values – 8 fields per card

End-of-record card

Run cards – 5 fields per card – as many cards as number of cases

Run number – does not have to be sequenced

Mach number

Angle of attack, degrees

Reynolds number times 10^{-6}

IFIX – automatically selected if blank

The output starts by tabulating the information pertinent to the geometry of the airfoil and the conformal mapping. The information relative to the flow solution is temporarily written on TAPE3 until the iteration process is terminated. Then either one of two forms is used. If the overall process converges, only the last set of boundary-layer characteristics and the last inviscid-flow definition are retrieved from TAPE3. Additionally, a CalComp plot of the pressure distribution is generated. If convergence is not attained, and the iteration process is automatically stopped at the limit of four inviscid calculations by using the fine grid, the boundary-layer and inviscid-flow characteristics are presented for all the iterations. Four CalComp plots are generated for the four pressure distributions computed by using the fine grid. Then, TAPE3 is rewound, and the output data for the next case is again temporarily written on it and again retrieved after the convergence is evaluated.

For the converged case, the specifics of the output are as follows. The title card and input data are printed for identification purposes along with a statement regarding the convergence of the iteration process. Next, the information used for the boundary-layer calculation is tabulated as well as the boundary-layer characteristics on the upper and lower surfaces. The displacement thickness, momentum thickness, shape factor, and local skin-friction coefficient are listed, and the presence of separation is noted. Then, a page-size plot of the displacement thickness on the upper and lower surfaces from 1 percent to 100 percent chord is shown. The coordinates for the equivalent inviscid airfoil are listed, followed by the pressure distribution and the lift, drag, and pitching moment in coefficient form. A page-size plot of the pressure distribution is also shown from 1 percent to 100 percent chord. Finally, a summary chart of all the iterations is presented which lists the lift coefficient, the number of cycles for an inviscid computation, and the location of separation, if any, on both the upper and lower surfaces. This chart should be examined for any peculiar behavior of the intermediate iterations. The output for cases which are not converged consists of this data for all the iterations rather than just the last one. It should be noted that the overlaid version does not produce a CalComp plotting tape until all cases are analyzed. If an abnormal exit is encountered, the plotting information for the cases thus far completed will be lost unless saved on a tape by means of the control cards. Then this save tape can be used to generate a plotting tape by combining it with the appropriate part of the program and resubmitting it.

A listing of the input data and the program output is given on the following pages for a sample case which required 200 seconds of central processor time. The trailing-edge adjustment parameter IFIX was not defined in the input, and so the empirical relation within the analysis selected the appropriate value. The resulting CalComp plot of the pressure distribution is shown in figure 20 to illustrate its format. The input data appear across the top. The lift, pitching-moment, and drag coefficients are summarized at the bottom along with the value for IFIX, the number of cycles for the inviscid computation NCY, the artificial viscosity level EP, and the number of grid points for the inviscid solution $M \times N$. The long tick on the vertical axis indicates the critical pressure coefficient level.

INPUT DATA – SAMPLE CASE

```
                               **********

                            COLUMN   NUMBER
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

                               **********

KORN AIRFOIL (THEOR COORD WITH AXIS ROTATED 0.12 DEG)
      76.       74.        1.
UPPER SURFACE COORDINATES (TABLE OF X/C THEN TABLE OF Y/C)
.0000000  .0001707  .0002774  .0004024  .0008207  .0014433  .0027634  .0043530
.0069575  .0087138  .0111412  .0146477  .0196968  .0235426  .0283907  .0339274
.0378338  .0409463  .0511148  .0540662  .0570022  .0600323  .0629989  .0688572
.0740271  .0766130  .0791126  .0818048  .0848120  .0887955  .0943301  .1029820
.1088650  .1160702  .1252516  .1364073  .1497806  .1655668  .1843911  .2060384
.2304857  .2575924  .2870966  .3187641  .3521519  .3867489  .4211690  .4558077
.4901447  .5236700  .5546379  .5841017  .6116746  .6369992  .6587513  .6780408
.7065550  .7280635  .7383184  .7610531  .7765781  .7918669  .8087590  .8246416
.8366122  .8568614  .8752980  .8979475  .9133739  .9273796  .9395038  .9517002
.9715175  .9888925  .9976367 1.0000000
.0000000  .0031239  .0038896  .0045925  .0062157  .0078003  .0100386  .0119356
.0143825  .0157936  .0175490  .0198103  .0226760  .0246304  .0268818  .0292199
.0307471  .0318990  .0352243  .0361355  .0370040  .0378621  .0386661  .0401522
.0413517  .0419114  .0424275  .0429563  .0435175  .0442192  .0451322  .0464543
.0472947  .0482710  .0494409  .0507635  .0522269  .0538026  .0554932  .0572142
.0589021  .0604900  .0619121  .0631136  .0640399  .0646494  .0649128  .0648330
.0644033  .0636268  .0625651  .0612071  .0595843  .0577608  .0559207  .0540710
.0508592  .0481866  .0468213  .0435875  .0412078  .0387602  .0359227  .0331232
.0309434  .0271255  .0235324  .0190136  .0159090  .0131128  .0107233  .0083711
.0047122  .0017650  .0004593  .0001234
LOWER SURFACE COORDINATES (TABLE OF X/C THEN TABLE OF Y/C)
.0000000  .0001020  .0001677  .0002520  .0003544  .0004675  .0006943  .0010316
.0015058  .0019019  .0023409  .0030747  .0042774  .0048290  .0058715  .0071353
.0083710  .0100814  .0124032  .0154701  .0194148  .0254882  .0295958  .0393139
.0527423  .0627017  .0735680  .0855366  .1000756  .1188486  .1413490  .1675828
.2026627  .2283692  .2479330  .2641452  .2822689  .2973241  .3105001  .3245101
.3454288  .3641594  .3884023  .4053111  .4271864  .4509051  .4762792  .5219445
.5499170  .5786508  .6053171  .6234647  .6421117  .6574869  .6711358  .6902354
.7087596  .7265269  .7464950  .7682830  .7846519  .8008943  .8189163  .8382702
```

```
.8585386  .8791581  .9034650  .9262652  .9434243  .9583859  .9733971  .9859140
.9962660 1.0000000
.0000000 -.0020190 -.0026897 -.0032561 -.0037992 -.0042930 -.0050788 -.0059921
-.0070163 -.0077434 -.0084678 -.0095382 -.0109573 -.0115718 -.0126227 -.0137472
-.0147253 -.0159378 -.0173967 -.0190951 -.0210218 -.0236028 -.0251638 -.0284252
-.0322162 -.0346358 -.0369776 -.0392487 -.0416518 -.0442745 -.0468145 -.0490945
-.0512022 -.0521781 -.0526385 -.0528460 -.0529007 -.0528072 -.0526265 -.0523362
-.0517160 -.0509744 -.0497571 -.0487396 -.0472198 -.0453149 -.0429923 -.0381648
-.0347510 -.0309447 -.0271778 -.0245072 -.0216993 -.0193718 -.0172837 -.0143739
-.0116115 -.0090595 -.0063567 -.0036779 -.0018870 -.0003194  .0011607  .0024410
.0034392  .0040999  .0044234  .0042731  .0038563  .0032664  .0024303  .0015165
.0005652  .0001234
END OF RECORD
    10.       0.702      1.10      21.18
```

OUTPUT DATA – SAMPLE CASE

ANALYSIS OF TWO-DIMENSIONAL, TRANSONIC, VISCOUS FLOW

KORN AIRFOIL (THEOR COORD WITH AXIS RCTATED 0.12 DEG)

THERE ARE 10 SMOOTHING ITERATIONS USED

AIRFOIL COORDINATES AND CURVATURES

X	Y	ARC LENGTH	THETA	KAPPA	KP	KPP
1.000000	.000123	0.000000	-5.862	-76944.0904	4.378E+00	0.
.998020	.000323	.001990	-5.653	3.0663	4.378E+00	-3.330E+02
.995648	.000549	.004373	-5.289	1.8639	-5.666E+00	5.298E+02
.992955	.000794	.007077	-5.056	1.6810	7.707E+00	-4.271E+02
.989775	.001065	.010268	-4.706	1.8232	-2.610E+00	1.625E+02
.986028	.001362	.014027	-4.365	1.4677	1.295E+00	-6.041E+01
.981663	.001681	.018404	-4.009	1.3535	-1.685E-01	2.374E+01
.976659	.002016	.023419	-3.642	1.2150	4.122E-01	-4.037E+00
.971020	.002356	.029068	-3.264	1.1291	3.128E-01	3.395E+00
.964762	.002691	.035336	-2.872	1.0586	3.966E-01	1.721E-01
.957909	.003010	.042196	-2.467	1.0042	4.009E-01	7.746E-01
.950490	.003302	.049620	-2.049	.9599	4.199E-01	7.210E-01
.942536	.003557	.057578	-1.620	.9239	4.375E-01	1.142E+00
.934077	.003764	.066040	-1.180	.8948	4.652E-01	1.496E+00
.925143	.003912	.074975	-.728	.8719	5.010E-01	1.788E+00
.915764	.003993	.084355	-.264	.8545	5.435E-01	1.915E+00
.905969	.003998	.094150	.212	.8420	5.885E-01	1.820E+00
.895787	.003917	.104332	.700	.8334	6.308E-01	1.416E+00
.885246	.003742	.114874	1.202	.8280	6.633E-01	5.945E-01
.874374	.003465	.125750	1.717	.8246	6.767E-01	-7.595E-01
.863197	.003078	.136934	2.244	.8218	6.597E-01	-2.728E+00
.851741	.002575	.148401	2.783	.8180	5.996E-01	-5.323E+00
.840032	.001950	.160127	3.330	.8114	4.838E-01	-8.452E+00
.828094	.001197	.172088	3.883	.8001	3.025E-01	-1.190E+01
.815952	.000314	.184263	4.435	.7821	5.071E-02	-1.536E+01
.803628	-.000701	.196629	4.981	.7559	-2.694E-01	-1.841E+01
.791144	-.001848	.209165	5.511	.7202	-6.477E-01	-2.066E+01
.778522	-.003123	.221852	6.019	.6744	-1.066E+00	-2.176E+01
.765780	-.004520	.234670	6.494	.6187	-1.500E+00	-2.147E+01
.752938	-.006032	.247600	6.929	.5541	-1.922E+00	-1.971E+01
.740014	-.007648	.260625	7.316	.4820	-2.303E+00	-1.655E+01
.727023	-.009356	.273728	7.649	.4046	-2.618E+00	-1.224E+01
.713981	-.011140	.286892	7.924	.3244	-2.848E+00	-7.219E+00
.700902	-.012987	.300100	8.139	.2439	-2.981E+00	-2.074E+00
.687799	-.014880	.313339	8.294	.1655	-3.019E+00	2.520E+00
.674686	-.016804	.326593	8.391	.0911	-2.974E+00	6.009E+00
.661573	-.018744	.339849	8.434	.0221	-2.868E+00	8.170E+00
.648472	-.020687	.353093	8.426	-.0410	-2.727E+00	9.169E+00
.635394	-.022619	.366312	8.373	-.0980	-2.571E+00	9.428E+00
.622349	-.024529	.379497	8.279	-.1492	-2.413E+00	9.381E+00
.609347	-.026407	.392634	8.150	-.1950	-2.259E+00	9.330E+00
.596397	-.028243	.405713	7.988	-.2360	-2.108E+00	9.507E+00
.583509	-.030030	.418725	7.798	-.2725	-1.957E+00	1.018E+01

.570690	-.031762	.431659	7.584	-.3046	-1.798E+00	1.156E+01
.557950	-.033432	.444508	7.349	-.3326	-1.620E+00	1.351E+01
.545297	-.035036	.457263	7.097	-.3561	-1.417E+00	1.520E+01
.532738	-.036570	.469916	6.832	-.3752	-1.192E+00	1.533E+01
.520280	-.038033	.482459	6.557	-.3899	-9.688E-01	1.283E+01
.507931	-.039421	.494886	6.275	-.4009	-7.855E-01	7.716E+00
.495697	-.040736	.507191	5.989	-.4092	-6.773E-01	1.431E+00
.483584	-.041976	.519367	5.701	-.4163	-6.577E-01	-3.744E+00
.471598	-.043142	.531409	5.412	-.4235	-7.082E-01	-5.911E+00
.459745	-.044235	.543312	5.120	-.4315	-7.866E-01	-4.611E+00
.448030	-.045254	.555071	4.826	-.4403	-8.465E-01	-9.911E-01
.436458	-.046201	.566682	4.530	-.4494	-8.592E-01	2.952E+00
.425033	-.047077	.578141	4.232	-.4581	-8.222E-01	5.502E+00
.413759	-.047882	.589444	3.933	-.4659	-7.548E-01	5.976E+00
.402639	-.048617	.600588	3.633	-.4727	-6.829E-01	4.745E+00
.391678	-.049284	.611570	3.334	-.4785	-6.270E-01	2.696E+00
.380877	-.049885	.622387	3.036	-.4836	-5.958E-01	6.135E-01
.370241	-.050422	.633037	2.739	-.4885	-5.889E-01	-1.175E+00
.359771	-.050896	.643518	2.445	-.4933	-6.019E-01	-2.765E+00
.349469	-.051310	.653827	2.152	-.4982	-6.319E-01	-4.419E+00
.339338	-.051664	.663965	1.861	-.5035	-6.789E-01	-6.278E+00
.329379	-.051963	.673928	1.572	-.5093	-7.443E-01	-8.222E+00
.319594	-.052207	.683716	1.284	-.5158	-8.280E-01	-9.922E+00
.309984	-.052399	.693329	.998	-.5231	-9.270E-01	-1.107E+01
.300549	-.052540	.702764	.713	-.5313	-1.035E+00	-1.159E+01
.291291	-.052632	.712023	.429	-.5404	-1.146E+00	-1.181E+01
.282210	-.052678	.721104	.146	-.5503	-1.256E+00	-1.232E+01
.273306	-.052678	.730008	-.138	-.5610	-1.368E+00	-1.374E+01
.264579	-.052636	.738735	-.421	-.5726	-1.491E+00	-1.653E+01
.256030	-.052552	.747285	-.705	-.5851	-1.635E+00	-2.089E+01
.247658	-.052428	.755657	-.989	-.5986	-1.813E+00	-2.673E+01
.239463	-.052266	.763854	-1.273	-.6135	-2.035E+00	-3.371E+01
.231444	-.052068	.771875	-1.559	-.6301	-2.309E+00	-4.122E+01
.223602	-.051835	.779721	-1.846	-.6486	-2.635E+00	-4.837E+01
.215934	-.051569	.787394	-2.136	-.6694	-3.010E+00	-5.419E+01
.208440	-.051270	.794894	-2.428	-.6927	-3.419E+00	-5.792E+01
.201119	-.050941	.802222	-2.724	-.7184	-3.845E+00	-5.942E+01
.193970	-.050582	.809380	-3.025	-.7465	-4.272E+00	-5.935E+01
.186992	-.050195	.816369	-3.330	-.7769	-4.688E+00	-5.910E+01
.180183	-.049780	.823190	-3.640	-.8092	-5.091E+00	-6.046E+01
.173543	-.049340	.829845	-3.954	-.8433	-5.493E+00	-6.500E+01
.167069	-.048874	.836336	-4.275	-.8791	-5.914E+00	-7.347E+01
.160760	-.048385	.842664	-4.600	-.9167	-6.377E+00	-8.534E+01
.154615	-.047873	.848830	-4.931	-.9563	-6.901E+00	-9.873E+01
.148631	-.047339	.854838	-5.267	-.9981	-7.491E+00	-1.109E+02
.142807	-.046785	.860688	-5.609	-1.0423	-8.136E+00	-1.194E+02
.137141	-.046211	.866383	-5.957	-1.0889	-8.811E+00	-1.235E+02
.131631	-.045619	.871924	-6.310	-1.1379	-9.490E+00	-1.258E+02
.126276	-.045010	.877314	-6.669	-1.1890	-1.016E+01	-1.319E+02
.121073	-.044385	.882555	-7.034	-1.2421	-1.085E+01	-1.494E+02
.116020	-.043745	.887648	-7.405	-1.2973	-1.160E+01	-1.854E+02
.111115	-.043091	.892597	-7.780	-1.3548	-1.251E+01	-2.432E+02
.106356	-.042425	.897402	-8.162	-1.4154	-1.366E+01	-3.200E+02
.101740	-.041747	.902067	-8.549	-1.4802	-1.514E+01	-4.060E+02
.097266	-.041059	.906594	-8.941	-1.5503	-1.695E+01	-4.859E+02
.092931	-.040362	.910985	-9.341	-1.6265	-1.906E+01	-5.437E+02
.088733	-.039656	.915242	-9.747	-1.7095	-2.134E+01	-5.693E+02
.084669	-.038943	.919368	-10.162	-1.7990	-2.366E+01	-5.671E+02
.080737	-.038223	.923365	-10.585	-1.8945	-2.589E+01	-5.604E+02
.076936	-.037498	.927235	-11.016	-1.9951	-2.803E+01	-5.875E+02
.073261	-.036768	.930981	-11.455	-2.1003	-3.020E+01	-6.896E+02
.069711	-.036035	.934606	-11.903	-2.2100	-3.266E+01	-8.929E+02
.066284	-.035298	.938112	-12.358	-2.3256	-3.574E+01	-1.196E+03
.062977	-.034560	.941500	-12.821	-2.4488	-3.973E+01	-1.572E+03
.059787	-.033820	.944775	-13.293	-2.5821	-4.479E+01	-1.981E+03
.056712	-.033080	.947938	-13.774	-2.7281	-5.096E+01	-2.401E+03
.053749	-.032340	.950992	-14.265	-2.8886	-5.817E+01	-2.839E+03
.050895	-.031601	.953939	-14.767	-3.0655	-6.639E+01	-3.335E+03
.048150	-.030865	.956782	-15.282	-3.2602	-7.571E+01	-3.942E+03
.045508	-.030130	.959524	-15.811	-3.4742	-8.634E+01	-4.694E+03
.042969	-.029398	.962166	-16.354	-3.7095	-9.852E+01	-5.601E+03
.040529	-.028669	.964712	-16.914	-3.9684	-1.125E+02	-6.681E+03
.038187	-.027944	.967165	-17.491	-4.2533	-1.286E+02	-8.023E+03
.035939	-.027223	.969526	-18.087	-4.5671	-1.472E+02	-9.837E+03
.033763	-.026506	.971798	-18.704	-4.9134	-1.692E+02	-1.245E+04
.031716	-.025794	.973984	-19.342	-5.2978	-1.959E+02	-1.617E+04
.029737	-.025086	.976086	-20.006	-5.7284	-2.293E+02	-2.117E+04
.027841	-.024383	.978107	-20.697	-6.2157	-2.713E+02	-2.740E+04

.026028	-.023685	.980050	-21.419	-6.7724	-3.236E+02	-3.472E+04
.024294	-.022992	.981918	-22.177	-7.4116	-3.872E+02	-4.317E+04
.022638	-.022304	.983712	-22.975	-8.1464	-4.633E+02	-5.325E+04
.021056	-.021619	.985435	-23.820	-8.9900	-5.533E+02	-6.600E+04
.019547	-.020939	.987090	-24.717	-9.9577	-6.606E+02	-8.262E+04
.018108	-.020263	.988680	-25.673	-11.0679	-7.895E+02	-1.039E+05
.016738	-.019589	.990207	-26.696	-12.3435	-9.453E+02	-1.298E+05
.015433	-.018918	.991674	-27.793	-13.8107	-1.132E+03	-1.583E+05
.014194	-.018248	.993083	-28.974	-15.4954	-1.351E+03	-1.852E+05
.013016	-.017580	.994437	-30.248	-17.4171	-1.597E+03	-2.036E+05
.011900	-.016911	.995739	-31.626	-19.5814	-1.857E+03	-2.037E+05
.010843	-.016241	.996990	-33.113	-21.9698	-2.107E+03	-1.770E+05
.009844	-.015570	.998194	-34.716	-24.5339	-2.317E+03	-1.244E+05
.008901	-.014896	.999353	-36.433	-27.1984	-2.458E+03	-6.620E+04
.008012	-.014218	1.000470	-38.260	-29.8812	-2.531E+03	-4.432E+04
.007177	-.013537	1.001548	-40.187	-32.5313	-2.578E+03	-1.058E+05
.006394	-.012851	1.002589	-42.205	-35.1686	-2.686E+03	-2.683E+05
.005660	-.012161	1.003596	-44.312	-37.9004	-2.951E+03	-4.904E+05
.004975	-.011467	1.004572	-46.512	-40.8917	-3.420E+03	-6.773E+05
.004338	-.010767	1.005518	-48.819	-44.2945	-4.049E+03	-7.278E+05
.003747	-.010062	1.006438	-51.254	-48.1715	-4.706E+03	-5.885E+05
.003202	-.009351	1.007334	-53.835	-52.4537	-5.224E+03	-2.654E+05
.002702	-.008633	1.008209	-56.575	-56.9468	-5.451E+03	2.163E+05
.002249	-.007908	1.009065	-59.475	-61.3620	-5.270E+03	8.528E+05
.001842	-.007174	1.009904	-62.525	-65.3363	-4.567E+03	1.630E+06
.001481	-.006431	1.010730	-65.695	-68.4426	-3.246E+03	2.431E+06
.001166	-.005678	1.011546	-68.941	-70.2272	-1.301E+03	2.983E+06
.000898	-.004916	1.012354	-72.202	-70.3190	1.064E+03	2.964E+06
.000674	-.004144	1.013158	-75.408	-68.5908	3.402E+03	2.217E+06
.000492	-.003363	1.013960	-78.492	-65.2881	5.147E+03	8.991E+05
.000352	-.002571	1.014765	-81.404	-61.0283	5.857E+03	-5.861E+05
.000251	-.001768	1.015573	-84.129	-56.6503	5.392E+03	-1.809E+06
.000186	-.000956	1.016389	-86.686	-52.9861	3.944E+03	-2.508E+06
.000156	-.000132	1.017213	-89.128	-50.6608	1.915E+03	-2.638E+06
.000161	.000704	1.018049	-91.531	-49.9903	-2.482E+02	-2.302E+06
.000201	.001552	1.018898	-93.980	-50.9768	-2.166E+03	-1.648E+06
.000281	.002412	1.019762	-96.559	-53.3601	-3.564E+03	-8.059E+05
.000402	.003285	1.020644	-99.335	-56.6797	-4.261E+03	1.150E+05
.000571	.004170	1.021545	-102.355	-60.3303	-4.159E+03	9.951E+05
.000794	.005065	1.022467	-105.634	-63.6236	-3.259E+03	1.694E+06
.001077	.005968	1.023414	-109.153	-65.8763	-1.685E+03	2.076E+06
.001426	.006877	1.024388	-112.855	-66.5266	2.984E+02	2.066E+06
.001846	.007789	1.025392	-116.656	-65.2545	2.334E+03	1.694E+06
.002343	.008701	1.026430	-120.451	-62.0597	4.060E+03	1.092E+06
.002918	.009610	1.027506	-124.135	-57.2582	5.213E+03	4.444E+05
.003573	.010515	1.028623	-127.615	-51.3899	5.700E+03	-9.497E+04
.004308	.011414	1.029785	-130.826	-45.0747	5.592E+03	-4.457E+05
.005123	.012309	1.030995	-133.732	-38.8710	5.063E+03	-6.049E+05
.006015	.013199	1.032255	-136.328	-33.1833	4.315E+03	-6.182E+05
.006983	.014086	1.033568	-138.632	-28.2331	3.518E+03	-5.460E+05
.008026	.014971	1.034937	-140.676	-24.0821	2.785E+03	-4.406E+05
.009143	.015856	1.036362	-142.498	-20.6828	2.169E+03	-3.356E+05
.010334	.016743	1.037847	-144.136	-17.9313	1.680E+03	-2.471E+05
.011600	.017632	1.039394	-145.622	-15.7086	1.305E+03	-1.788E+05
.012941	.018526	1.041005	-146.986	-13.9035	1.022E+03	-1.286E+05
.014358	.019424	1.042683	-148.249	-12.4232	8.104E+02	-9.256E+04
.015853	.020328	1.044431	-149.429	-11.1935	6.517E+02	-6.705E+04
.017423	.021238	1.046250	-150.539	-10.1570	5.320E+02	-4.941E+04
.019086	.022154	1.048143	-151.592	-9.2709	4.402E+02	-3.757E+04
.020827	.023076	1.050113	-152.594	-8.5049	3.675E+02	-2.984E+04
.022654	.024004	1.052163	-153.552	-7.8389	3.074E+02	-2.468E+04
.024571	.024938	1.054295	-154.473	-7.2610	2.558E+02	-2.067E+04
.026578	.025877	1.056511	-155.363	-6.7631	2.108E+02	-1.662E+04
.028679	.026821	1.058815	-156.226	-6.3373	1.732E+02	-1.195E+04
.030877	.027770	1.061208	-157.070	-5.9705	1.452E+02	-6.897E+03
.033173	.028721	1.063694	-157.896	-5.6434	1.283E+02	-2.529E+03
.035571	.029675	1.066274	-158.707	-5.3326	1.219E+02	-1.444E+02
.038073	.030630	1.068952	-159.502	-5.0187	1.215E+02	-4.425E+02
.040681	.031585	1.071731	-160.275	-4.6952	1.203E+02	-2.942E+03
.043400	.032540	1.074611	-161.023	-4.3729	1.120E+02	-6.133E+03
.046229	.033493	1.077597	-161.745	-4.0766	9.400E+01	-8.295E+03
.049174	.034444	1.080691	-162.445	-3.8341	6.878E+01	-8.393E+03
.052234	.035392	1.083896	-163.132	-3.6628	4.234E+01	-6.482E+03
.055414	.036335	1.087213	-163.817	-3.5617	2.120E+01	-3.431E+03
.058716	.037272	1.090645	-164.512	-3.5113	9.624E+00	-3.157E+02
.062143	.038193	1.094194	-165.223	-3.4810	8.522E+00	2.082E+03
.065697	.039112	1.097864	-165.951	-3.4384	1.604E+01	3.435E+03
.069381	.040008	1.101656	-166.690	-3.3569	2.886E+01	3.759E+03

.073199	.040885	1.105574	-167.429	-3.2208	4.336E+01	3.252E+03
.077153	.041741	1.109619	-168.154	-3.0264	5.631E+01	2.185E+03
.081244	.042573	1.113794	-168.849	-2.7811	6.530E+01	8.666E+02
.085476	.043381	1.118102	-169.502	-2.5012	6.898E+01	-3.963E+02
.089849	.044167	1.122545	-170.101	-2.2080	6.724E+01	-1.360E+03
.094365	.044933	1.127126	-170.642	-1.9230	6.110E+01	-1.902E+03
.099026	.045680	1.131846	-171.127	-1.6634	5.224E+01	-2.030E+03
.103834	.046412	1.136709	-171.558	-1.4397	4.249E+01	-1.854E+03
.108739	.047130	1.141716	-171.943	-1.2552	3.332E+01	-1.519E+03
.113894	.047837	1.146870	-172.291	-1.1076	2.558E+01	-1.150E+03
.119150	.048533	1.152172	-172.609	-.9912	1.955E+01	-8.251E+02
.124560	.049220	1.157625	-172.904	-.8994	1.510E+01	-5.766E+02
.130124	.049899	1.163231	-173.180	-.8258	1.189E+01	-4.021E+02
.135845	.050570	1.168991	-173.443	-.7656	9.598E+00	-2.852E+02
.141725	.051233	1.174908	-173.693	-.7152	7.923E+00	-2.076E+02
.147766	.051887	1.180985	-173.935	-.6722	6.670E+00	-1.554E+02
.153969	.052533	1.187221	-174.168	-.6347	5.706E+00	-1.194E+02
.160336	.053171	1.193620	-174.395	-.6016	4.946E+00	-9.420E+01
.166870	.053799	1.200184	-174.615	-.5720	4.330E+00	-7.613E+01
.173570	.054418	1.206913	-174.830	-.5455	3.819E+00	-6.265E+01
.180440	.055027	1.213810	-175.041	-.5214	3.387E+00	-5.206E+01
.187481	.055625	1.220876	-175.248	-.4994	3.019E+00	-4.339E+01
.194693	.056212	1.228112	-175.451	-.4794	2.705E+00	-3.623E+01
.202079	.056786	1.235520	-175.650	-.4610	2.435E+00	-3.045E+01
.209639	.057348	1.243101	-175.847	-.4440	2.203E+00	-2.590E+01
.217376	.057897	1.250857	-176.040	-.4282	2.001E+00	-2.239E+01
.225288	.058431	1.258788	-176.231	-.4136	1.822E+00	-1.969E+01
.233379	.058951	1.266895	-176.420	-.4000	1.661E+00	-1.756E+01
.241648	.059454	1.275179	-176.607	-.3874	1.514E+00	-1.584E+01
.250096	.059941	1.283641	-176.792	-.3757	1.378E+00	-1.440E+01
.258723	.060411	1.292281	-176.975	-.3648	1.251E+00	-1.315E+01
.267531	.060862	1.301100	-177.157	-.3548	1.133E+00	-1.202E+01
.276518	.061294	1.310098	-177.338	-.3456	1.022E+00	-1.091E+01
.285686	.061706	1.319275	-177.517	-.3372	9.196E-01	-9.817E+00
.295034	.062097	1.328631	-177.696	-.3296	8.252E-01	-8.746E+00
.304561	.062465	1.338166	-177.874	-.3227	7.392E-01	-7.762E+00
.314268	.062810	1.347878	-178.052	-.3165	6.612E-01	-6.938E+00
.324153	.063131	1.357769	-178.230	-.3109	5.899E-01	-6.325E+00
.334215	.063427	1.367836	-178.407	-.3060	5.235E-01	-5.931E+00
.344454	.063695	1.378078	-178.586	-.3017	4.600E-01	-5.719E+00
.354868	.063936	1.388494	-178.765	-.2980	3.974E-01	-5.625E+00
.365455	.064148	1.399084	-178.945	-.2950	3.346E-01	-5.583E+00
.376213	.064329	1.409844	-179.126	-.2927	2.709E-01	-5.546E+00
.387141	.064478	1.420772	-179.308	-.2912	2.064E-01	-5.498E+00
.398235	.064594	1.431867	-179.493	-.2905	1.411E-01	-5.459E+00
.409493	.064676	1.443125	-179.681	-.2907	7.499E-02	-5.468E+00
.420912	.064720	1.454544	-179.871	-.2918	7.436E-03	-5.562E+00
.432488	.064727	1.466121	-180.065	-.2940	-6.264E-02	-5.761E+00
.444218	.064693	1.477850	-180.264	-.2972	-1.366E-01	-6.053E+00
.456097	.064617	1.489729	-180.468	-.3018	-2.159E-01	-6.393E+00
.468121	.064497	1.501754	-180.678	-.3078	-3.012E-01	-6.699E+00
.480284	.064330	1.513919	-180.895	-.3154	-3.923E-01	-6.867E+00
.492582	.064114	1.526219	-181.120	-.3247	-4.875E-01	-6.773E+00
.505009	.063846	1.538648	-181.355	-.3360	-5.831E-01	-6.300E+00
.517558	.063522	1.551201	-181.602	-.3493	-6.736E-01	-5.376E+00
.530222	.063139	1.563871	-181.861	-.3646	-7.523E-01	-4.018E+00
.542995	.062694	1.576652	-182.134	-.3818	-8.122E-01	-2.351E+00
.555867	.062182	1.589534	-182.423	-.4006	-8.479E-01	-5.629E-01
.568832	.061599	1.602512	-182.728	-.4206	-8.565E-01	1.213E+00
.581879	.060941	1.615576	-183.050	-.4414	-8.375E-01	3.031E+00
.595000	.060203	1.628717	-183.391	-.4626	-7.890E-01	5.142E+00
.608183	.059381	1.641927	-183.749	-.4834	-7.054E-01	7.789E+00
.621419	.058471	1.655194	-184.124	-.5030	-5.765E-01	1.085E+01
.634695	.057468	1.668508	-184.514	-.5203	-3.939E-01	1.362E+01
.648000	.056371	1.681858	-184.918	-.5340	-1.608E-01	1.500E+01
.661321	.055176	1.695232	-185.331	-.5428	1.002E-01	1.414E+01
.674644	.053884	1.708617	-185.749	-.5462	3.502E-01	1.108E+01
.687955	.052495	1.722001	-186.167	-.5445	5.495E-01	6.923E+00
.701242	.051011	1.735370	-186.582	-.5388	6.759E-01	3.189E+00
.714487	.049434	1.748710	-186.991	-.5306	7.351E-01	1.004E+00
.727677	.047770	1.762004	-187.392	-.5210	7.540E-01	5.396E-01
.740795	.046022	1.775237	-187.783	-.5107	7.644E-01	1.178E+00
.753824	.044197	1.788394	-188.164	-.4995	7.873E-01	2.103E+00
.766747	.042301	1.801455	-188.533	-.4868	8.288E-01	2.812E+00
.779546	.040339	1.814403	-188.889	-.4719	8.852E-01	3.225E+00
.792203	.038321	1.827220	-189.229	-.4542	9.509E-01	3.468E+00
.804698	.036255	1.839885	-189.551	-.4331	1.023E+00	3.627E+00
.817013	.034149	1.852379	-189.852	-.4080	1.099E+00	3.668E+00

.829126	.032015	1.864679	-190.130	-.3783	1.177E+00	3.500E+00
.841018	.029863	1.876764	-190.380	-.3434	1.252E+00	3.048E+00
.852667	.027705	1.888611	-190.600	-.3027	1.319E+00	2.282E+00
.864051	.025555	1.900196	-190.786	-.2560	1.369E+00	1.210E+00
.875148	.023426	1.911496	-190.935	-.2033	1.397E+00	-1.162E-01
.885935	.021331	1.922484	-191.045	-.1448	1.394E+00	-1.584E+00
.896387	.019283	1.933135	-191.114	-.0813	1.357E+00	-3.015E+00
.906480	.017297	1.943421	-191.142	-.0135	1.287E+00	-4.205E+00
.916189	.015386	1.953317	-191.130	.0574	1.188E+00	-4.995E+00
.925486	.013560	1.962792	-191.079	.1306	1.068E+00	-5.334E+00
.934345	.011832	1.971818	-190.992	.2056	9.397E-01	-5.270E+00
.942737	.010210	1.980365	-190.873	.2826	8.114E-01	-5.030E+00
.950633	.008704	1.988403	-190.725	.3621	6.880E-01	-4.567E+00
.958003	.007319	1.995902	-190.552	.4454	5.752E-01	-4.789E+00
.964816	.006061	2.002830	-190.358	.5337	4.565E-01	-3.126E+00
.971045	.004935	2.009160	-190.148	.6302	3.789E-01	-8.150E+00
.976663	.003940	2.014865	-189.925	.7327	1.772E-01	7.580E+00
.981653	.003077	2.019930	-189.696	.8597	3.636E-01	-4.310E+01
.986011	.002341	2.024350	-189.463	.9606	-6.858E-01	8.842E+01
.989756	.001724	2.028145	-189.238	1.1852	1.449E+00	-2.575E+02
.992938	.001213	2.031368	-189.009	1.0887	-4.803E+00	3.100E+02
.995635	.000790	2.034098	-188.859	1.1510	3.061E+00	-1.868E+02
.998013	.000423	2.036504	-188.639	1.8143	-2.600E+00	-2.263E-13
1.000000	.000123	2.038513	-188.515	-46340.9029	-2.600E+00	-2.263E-13

ERR	DA	DB
2.946951E-02	5.758071E-03	4.372224E-02
6.458831E-03	-1.805529E-03	1.327163E-02
2.615811E-03	1.652716E-04	3.823044E-03
7.410768E-04	-1.796608E-04	7.819629E-04
2.471375E-04	3.686719E-05	2.027501E-04
8.302199E-05	-3.058255E-05	2.283669E-05
3.098620E-05	-1.211597E-06	-8.316692E-06
1.063870E-05	-1.197288E-05	-1.825942E-05
4.389199E-06	-7.662224E-06	-1.977655E-05
1.605039E-06	-9.308708E-06	-2.033855E-05
6.490394E-07	-8.662946E-06	-2.040102E-05

MAPPING TO THE INSIDE OF A CIRCLE

DZ/DSIGMA = -(1/SIGMA**2)*(1-SIGMA)**(1-EPSIL)*(EXP(W(SIGMA))

W(SIGMA) = SUM((A(N)+I*B(N))*SIGMA**(N-1))

EPSIL = .015   300 POINTS AROUND THE CIRCLE

A(N)	B(N)
2.038513E+00	4.634364E-02
9.852617E-01	-2.040102E-05
-3.198867E-01	9.236633E-02
2.080350E-01	-1.811353E-02
-1.350887E-01	2.670647E-02
7.248768E-02	-3.248779E-02
-5.571157E-02	1.979889E-02
3.730862E-02	-2.156461E-02
-2.510498E-02	1.896828E-02
1.706313E-02	-1.428820E-02
-1.381418E-02	1.261280E-02
7.774508E-03	-8.407565E-03
-6.378555E-03	5.599031E-03
3.361523E-03	-3.276317E-03
-2.261805E-03	1.245588E-03
4.536691E-04	1.056626E-04

DEL S	RES	S/L	W(0)
2.855E-02	0.	1.00006	-1.29843
1.356E-04	1.333E-06	.99999	-1.29843

ANGLE OF ZERO LIFT = -2.65350   OUTER MAPPING RADIUS = .27296
THE THICKNESS TO CHORD RATIO IS .1173

APPENDIX

INPUT FOR BOUNDARY LAYER CALCULATION FROM PREVIOUS INVISCID SOLUTION

	***** UPPER SURFACE *****		***** LOWER SURFACE *****	
X/C	CP	CURV	CP	CURV
0.000	.800000	50.325570	.800000	50.325570
.025	-.644574	7.154498	.247582	7.151358
.050	-.904418	3.787818	.034812	3.128971
.075	-1.132435	3.132292	-.087394	2.050502
.100	-1.233545	1.618051	-.163480	1.507463
.125	-1.228268	.893546	-.213747	1.202042
.150	-1.144247	.658656	-.246087	.988527
.175	-.766439	.540442	-.266166	.835771
.200	-.822957	.466162	-.277978	.722812
.225	-.833460	.414162	-.283975	.645317
.250	-.787089	.375801	-.286101	.594810
.275	-.763934	.347159	-.284648	.558995
.300	-.746957	.326000	-.279691	.531835
.325	-.733023	.310527	-.271465	.512231
.350	-.721137	.299751	-.260100	.497967
.375	-.710333	.292998	-.245148	.486316
.400	-.700528	.290539	-.226944	.474087
.425	-.691813	.292568	-.205043	.458154
.450	-.683925	.299458	-.179807	.438851
.475	-.676698	.312066	-.151904	.421502
.500	-.669793	.331473	-.121146	.406279
.525	-.662460	.358332	-.087135	.384354
.550	-.653692	.392034	-.049945	.347374
.575	-.641293	.430447	-.009794	.293820
.600	-.624116	.470444	.032810	.224617
.625	-.600530	.507663	.076672	.138782
.650	-.568350	.535304	.120984	.033644
.675	-.528569	.546198	.164639	-.092892
.700	-.481069	.539380	.206403	-.238516
.725	-.428190	.522984	.245058	-.392171
.750	-.371084	.502770	.278807	-.537666
.775	-.310435	.477160	.306605	-.659005
.800	-.246164	.441030	.328546	-.745513
.825	-.178736	.388435	.343755	-.795522
.850	-.108469	.312030	.353884	-.817025
.875	-.036827	.204012	.358944	-.824780
.900	.034019	.057019	.359874	-.836970
.925	.100582	-.126779	.359874	-.871639
.950	.153476	-.355721	.359874	-.957667
.975	.178075	-.702334	.359874	-1.189744
1.000	.359874	-2.368634	.359874	-4.070168

UPPER SURFACE BOUNDARY LAYER

X/C	DELSTAR/C	MOMNTM/C	H	CF
.025	.00007	.0000	1.669	3.858E-03
.051	.00008	.0000	1.874	3.238E-03
.075	.00014	.0001	1.873	2.948E-03
.101	.00020	.0001	1.909	2.677E-03
.125	.00026	.0001	1.924	2.471E-03
.152	.00039	.0002	1.906	2.046E-03
.175	.00037	.0002	1.799	2.145E-03
.204	.00055	.0003	1.765	2.040E-03
.231	.00058	.0003	1.732	2.159E-03
.251	.00062	.0004	1.730	2.152E-03
.275	.00068	.0004	1.734	2.026E-03
.307	.00076	.0004	1.741	1.949E-03
.328	.00080	.0005	1.740	1.919E-03
.359	.00086	.0005	1.734	1.889E-03
.380	.00090	.0005	1.727	1.877E-03
.401	.00093	.0005	1.720	1.872E-03
.428	.00098	.0006	1.703	1.859E-03
.461	.00103	.0006	.693	1.872E-03
.477	.00106	.0006	1.689	1.873E-03
.511	.00112	.0007	1.682	1.865E-03
.527	.00115	.0007	1.680	1.859E-03
.560	.00121	.0007	1.676	1.840E-03
.577	.00124	.0007	1.675	1.824E-03

.609	.00132	.0008	1.675	1.781E-03
.625	.00137	.0008	1.670	1.751E-03
.650	.00145	.0009	1.671	1.685E-03
.697	.00165	.0010	1.684	1.563E-03
.720	.00177	.0010	1.693	1.489E-03
.742	.00190	.0011	1.704	1.411E-03
.763	.00206	.0012	1.717	1.330E-03
.784	.00223	.0013	1.733	1.245E-03
.803	.00242	.0014	1.752	1.158E-03
.838	.00284	.0016	1.793	9.807E-04
.861	.00322	.0018	1.841	8.378E-04
.881	.00363	.0019	1.891	7.152E-04
.899	.00407	.0021	1.948	6.184E-04
.914	.00453	.0023	2.009	5.247E-04
.929	.00498	.0024	2.068	4.445E-04
.942	.00541	.0026	2.114	3.913E-04
.956	.00577	.0027	2.141	3.646E-04
.968	.00655	.0029	2.228	2.140E-04
.975	.00749	.0032	2.364	9.454E-05

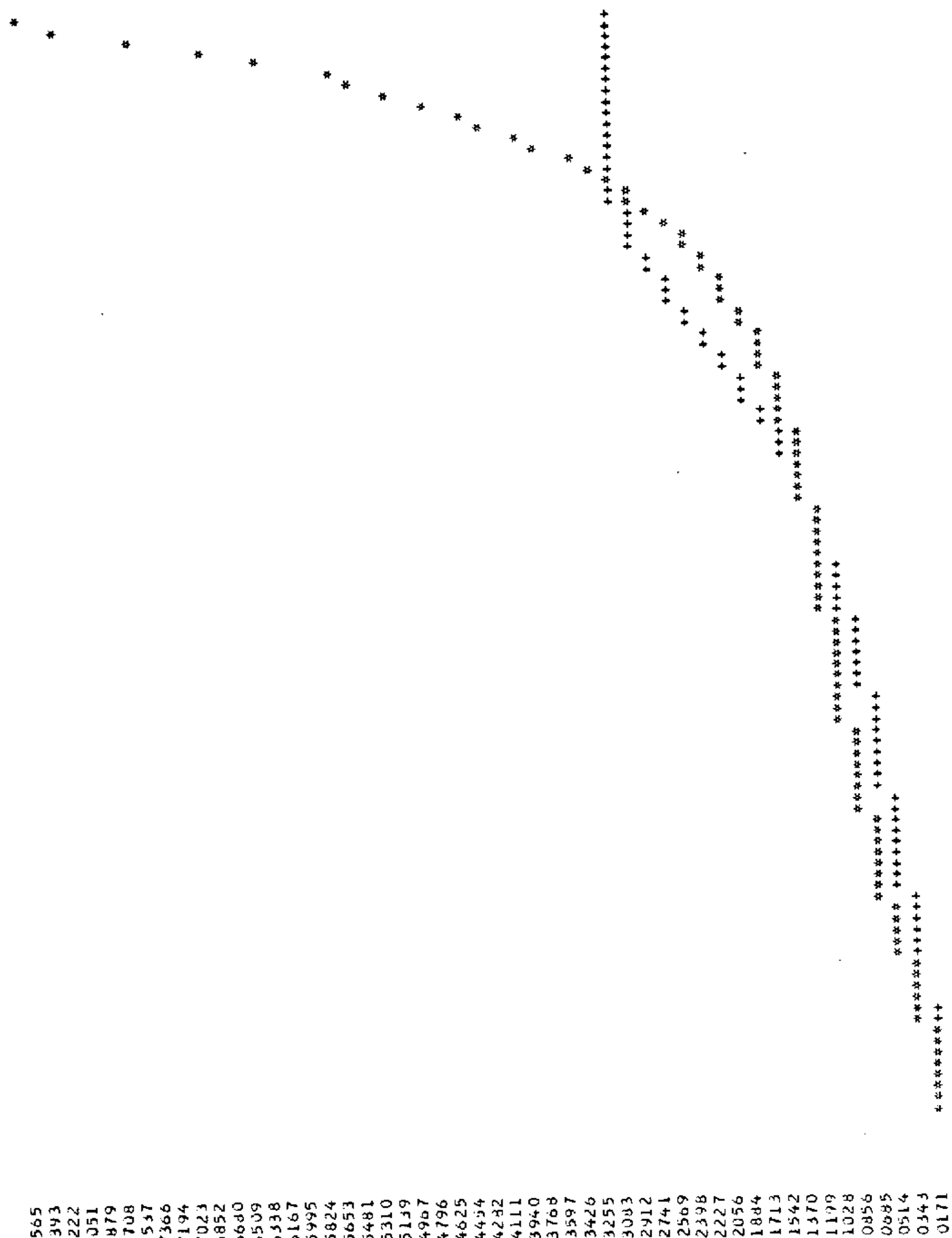
SEPARATION

LOWER SURFACE BOUNDARY LAYER

X/C	DELSTAR/C	MOMNTM/C	H	CF
.026	.00006	.0000	1.526	4.174E-03
.051	.00009	.0001	1.608	3.448E-03
.076	.00013	.0001	1.602	3.119E-03
.101	.00017	.0001	1.594	2.946E-03
.126	.00021	.0001	1.594	2.825E-03
.151	.00025	.0002	1.594	2.668E-03
.177	.00030	.0002	1.597	2.561E-03
.204	.00034	.0002	1.599	2.485E-03
.226	.00038	.0002	1.591	2.422E-03
.253	.00043	.0003	1.589	2.371E-03
.280	.00048	.0003	1.588	2.321E-03
.300	.00052	.0003	1.587	2.281E-03
.335	.00059	.0004	1.581	2.197E-03
.356	.00063	.0004	1.581	2.164E-03
.377	.00067	.0004	1.581	2.124E-03
.408	.00074	.0005	1.581	2.067E-03
.429	.00079	.0005	1.581	2.027E-03
.459	.00088	.0006	1.578	1.963E-03
.475	.00092	.0006	1.578	1.912E-03
.506	.00102	.0006	1.581	1.849E-03
.537	.00113	.0007	1.585	1.773E-03
.552	.00119	.0007	1.588	1.733E-03
.581	.00132	.0008	1.593	1.655E-03
.610	.00147	.0009	1.602	1.572E-03
.645	.00168	.0010	1.610	1.455E-03
.665	.00182	.0011	1.618	1.400E-03
.685	.00197	.0012	1.626	1.350E-03
.703	.00211	.0013	1.634	1.301E-03
.738	.00240	.0015	1.643	1.225E-03
.755	.00253	.0015	1.644	1.204E-03
.798	.00280	.0017	1.627	1.197E-03
.824	.00292	.0018	1.612	1.224E-03
.851	.00300	.0019	1.589	1.273E-03
.879	.00303	.0019	1.561	1.341E-03
.907	.00303	.0020	1.532	1.414E-03
.935	.00302	.0020	1.507	1.481E-03

RUN COMPLETED

APPENDIX


.8565
.8393
.8222
.8051
.7879
.7708
.7537
.7366
.7194
.7023
.6852
.6680
.6509
.6338
.6167
.5995
.5824
.5653
.5481
.5310
.5139
.4967
.4796
.4625
.4454
.4282
.4111
.3940
.3768
.3597
.3426
.3255
.3083
.2912
.2741
.2569
.2398
.2227
.2056
.1884
.1713
.1542
.1370
.1199
.1028
.0856
.0685
.0514
.0343
.0171

ADDITION OF BOUNDARY LAYER DELSTAR TO AIRFOIL COORDINATES

***** UPPER SURFACE *****

X-BODY	Y-BODY	THETA	DELSTAR	X-NEW	Y-NEW
0.000000	0.000000	-89.257	0.000000	0.000000	0.000000
.000171	.003124	-96.944	.000001	.000169	.003124
.000277	.003890	-98.979	.000002	.000275	.003890
.000402	.004592	-101.273	.000003	.000400	.004593
.000821	.006216	-107.818	.000006	.000815	.006217
.001443	.007800	-115.101	.000010	.001434	.007805
.002763	.010039	-125.806	.000018	.002749	.010049
.004353	.011936	-133.375	.000027	.004333	.011954
.006958	.014383	-139.746	.000039	.006932	.014413
.008714	.015794	-142.585	.000046	.008686	.015830
.011141	.017549	-145.540	.000054	.011111	.017593
.014648	.019810	-148.679	.000062	.014616	.019863
.019697	.022676	-152.030	.000068	.019665	.022736
.023543	.024630	-154.028	.000071	.023512	.024694
.028391	.026882	-156.095	.000072	.028362	.026947
.033927	.029220	-158.063	.000072	.033900	.029287
.037834	.030747	-159.186	.000072	.037808	.030815
.040946	.031899	-160.309	.000073	.040922	.031968
.051115	.035224	-162.694	.000082	.051090	.035302
.054066	.036136	-163.133	.000088	.054041	.036220
.057002	.037004	-163.868	.000095	.056976	.037095
.060032	.037862	-164.515	.000103	.060005	.037961
.062999	.038666	-165.152	.000110	.062971	.038773
.068857	.040152	-166.382	.000125	.068828	.040274
.074027	.041352	-167.504	.000139	.073997	.041487
.076613	.041911	-168.066	.000145	.076583	.042053
.079113	.042428	-168.606	.000151	.079083	.042575
.081805	.042956	-169.155	.000157	.081775	.043111
.084812	.043518	-169.695	.000164	.084783	.043679
.088795	.044219	-170.300	.000173	.088766	.044389
.094330	.045132	-170.931	.000184	.094301	.045314
.102982	.046454	-171.663	.000200	.102953	.046652
.108865	.047295	-172.066	.000208	.108836	.047501
.116070	.048271	-172.492	.000224	.116041	.048493
.125252	.049441	-172.976	.000256	.125220	.049695
.136407	.050763	-173.486	.000327	.136370	.051089
.149781	.052227	-174.016	.000382	.149741	.052607
.165567	.053803	-174.572	.000364	.165532	.054165
.184391	.055493	-175.152	.000428	.184355	.055920
.206038	.057214	-175.745	.000552	.205997	.057764
.230486	.058902	-176.344	.000582	.230449	.059482
.257592	.060490	-176.941	.000633	.257559	.061122
.287097	.061912	-177.531	.000714	.287066	.062625
.318764	.063114	-178.117	.000783	.318738	.063897
.352152	.064040	-178.700	.000849	.352133	.064888
.386749	.064649	-179.278	.000909	.386737	.065559
.421169	.064913	-179.845	.000965	.421166	.065878
.455808	.064833	-180.422	.001024	.455815	.065857
.490145	.064403	-181.018	.001081	.490164	.065484
.523670	.063627	-181.646	.001140	.523703	.064767
.554638	.062565	-182.294	.001197	.554686	.063761
.584102	.061207	-183.001	.001260	.584168	.062465
.611675	.059584	-183.746	.001328	.611761	.060910
.636999	.057761	-184.505	.001404	.637109	.059160
.658751	.055921	-185.149	.001477	.658884	.057392
.678041	.054071	-185.875	.001555	.678200	.055618
.706555	.050859	-186.830	.001692	.706756	.052540

***** LOWER SURFACE *****

X-BODY	Y-BODY	THETA	DELSTAR	X-NEW	Y-NEW
0.000000	0.000000	-89.257	0.000000	0.000000	0.000000
.000102	-.002019	-85.354	.000000	.000102	-.002019
.000168	-.002690	-82.959	.000001	.000167	-.002690
.000252	-.003256	-80.336	.000001	.000251	-.003256
.000354	-.003799	-78.234	.000002	.000353	-.003800
.000467	-.004293	-75.898	.000002	.000466	-.004293
.000694	-.005079	-71.955	.000003	.000691	-.005080
.001032	-.005992	-67.536	.000004	.001028	-.005994
.001506	-.007016	-62.903	.000006	.001500	-.007019
.001902	-.007743	-60.014	.000008	.001895	-.007747
.002341	-.008468	-57.684	.000010	.002333	-.008473
.003075	-.009538	-52.914	.000012	.003065	-.009546
.004277	-.010957	-48.390	.000017	.004265	-.010968
.004829	-.011572	-47.289	.000019	.004815	-.011584
.005872	-.012623	-43.429	.000022	.005856	-.012639
.007135	-.013747	-39.894	.000026	.007119	-.013767
.008371	-.014725	-36.949	.000030	.008353	-.014749
.010081	-.015938	-33.827	.000034	.010062	-.015966
.012403	-.017397	-30.601	.000040	.012383	-.017431
.015470	-.019095	-27.521	.000046	.015449	-.019136
.019415	-.021022	-24.670	.000053	.019393	-.021070
.025488	-.023603	-21.590	.000062	.025466	-.023660
.029596	-.025164	-20.058	.000066	.029573	-.025226
.039314	-.028425	-17.214	.000076	.039291	-.028498
.052742	-.032216	-14.459	.000091	.052720	-.032304
.062702	-.034636	-12.900	.000107	.062678	-.034740
.073568	-.036978	-11.449	.000125	.073543	-.037100
.085537	-.039249	-10.087	.000144	.085511	-.039391
.100076	-.041652	-8.722	.000168	.100050	-.041817
.118849	-.044274	-7.221	.000198	.118824	-.044471
.141349	-.046814	-5.712	.000236	.141325	-.047050
.167583	-.049095	-4.263	.000282	.167562	-.049376
.202663	-.051202	-2.673	.000343	.202647	-.051545
.228369	-.052178	-1.694	.000388	.228358	-.052566
.247933	-.052639	-1.007	.000423	.247926	-.053061
.264145	-.052846	-.465	.000452	.264142	-.053298
.282269	-.052901	.119	.000485	.282270	-.053386
.297324	-.052807	.588	.000513	.297329	-.053320
.310500	-.052626	.981	.000538	.310509	-.053164
.324510	-.052336	1.393	.000565	.324524	-.052901
.345429	-.051716	2.000	.000607	.345450	-.052322
.364159	-.050974	2.534	.000646	.364188	-.051620
.388402	-.049757	3.211	.000699	.388441	-.050455
.405311	-.048740	3.676	.000737	.405358	-.049475
.427186	-.047220	4.271	.000791	.427245	-.048008
.450905	-.045315	4.913	.000852	.450978	-.046164
.476279	-.042992	5.517	.000924	.476368	-.043912
.521945	-.038165	6.607	.001073	.522068	-.039231
.549917	-.034751	7.271	.001183	.550067	-.035924
.578651	-.030945	7.815	.001309	.578829	-.032242
.605317	-.027178	8.248	.001446	.605524	-.028609
.623465	-.024507	8.490	.001547	.623693	-.026037
.642112	-.021699	8.592	.001662	.642360	-.023343
.657487	-.019372	8.656	.001767	.657753	-.021118
.671136	-.017284	8.710	.001865	.671418	-.019127
.690235	-.014374	8.593	.002010	.690536	-.016362
.708760	-.011611	8.346	.002158	.709073	-.013747

.728064	.048187	-187.407	.001815	.728298	.049987
.738318	.046821	-187.745	.001880	.738572	.048684
.761053	.043587	-188.473	.002042	.761354	.045607
.776578	.041208	-188.917	.002167	.776914	.043349
.791867	.038760	-189.292	.002308	.792240	.041038
.808759	.035923	-189.782	.002483	.809181	.038369
.824642	.033123	-190.189	.002669	.825114	.035750
.836612	.030943	-190.457	.002831	.837126	.033727
.856861	.027126	-190.878	.003156	.857457	.030224
.875298	.023532	-191.163	.003510	.875978	.026976
.897948	.019014	-191.372	.004055	.898747	.022989
.913374	.015909	-191.352	.004497	.914259	.020318
.927380	.013113	-191.226	.004941	.928341	.017959
.939504	.010723	-191.051	.005339	.940527	.015964
.951700	.008371	-190.767	.005648	.952755	.013920
.971518	.004712	-190.104	.006895	.972727	.011500
.988892	.001765	-188.978	.008061	.990150	.009728
.997637	.000459	-188.088	.008470	.998828	.008845
1.000000	.000123	-188.093	.008565	1.000000	.008725

.726527	-.009059	7.981	.002304	.726847	-.011341
.746495	-.006357	7.403	.002463	.746812	-.008799
.768283	-.003678	6.590	.002618	.768583	-.006279
.784652	-.001887	5.887	.002722	.784931	-.004595
.800894	-.000319	5.131	.002815	.801146	-.003123
.818916	.001161	4.257	.002903	.819132	-.001734
.838270	.002441	3.312	.002967	.838442	-.000521
.858539	.003439	2.329	.003010	.858661	.000431
.879158	.004100	1.343	.003032	.879229	.001069
.903465	.004423	.180	.003031	.903475	.001393
.926265	.004273	-.947	.003018	.926215	.001255
.943424	.003856	-1.841	.003011	.943328	.000846
.958386	.003266	-2.698	.003004	.958244	.000266
.973397	.002430	-3.701	.002997	.973204	-.000560
.985914	.001516	-4.678	.002991	.985670	-.001464
.996266	.000565	-6.081	.002986	.995950	-.002404
1.000000	.000123	-8.078	.002984	1.000000	-.002884

NOTE THAT THE TRAILING EDGE OF THE EQUIVALENT AIRFOIL IS REDEFINED SO THAT X/C = 1.000

ORIGINAL PAGE IS
OF POOR QUALITY

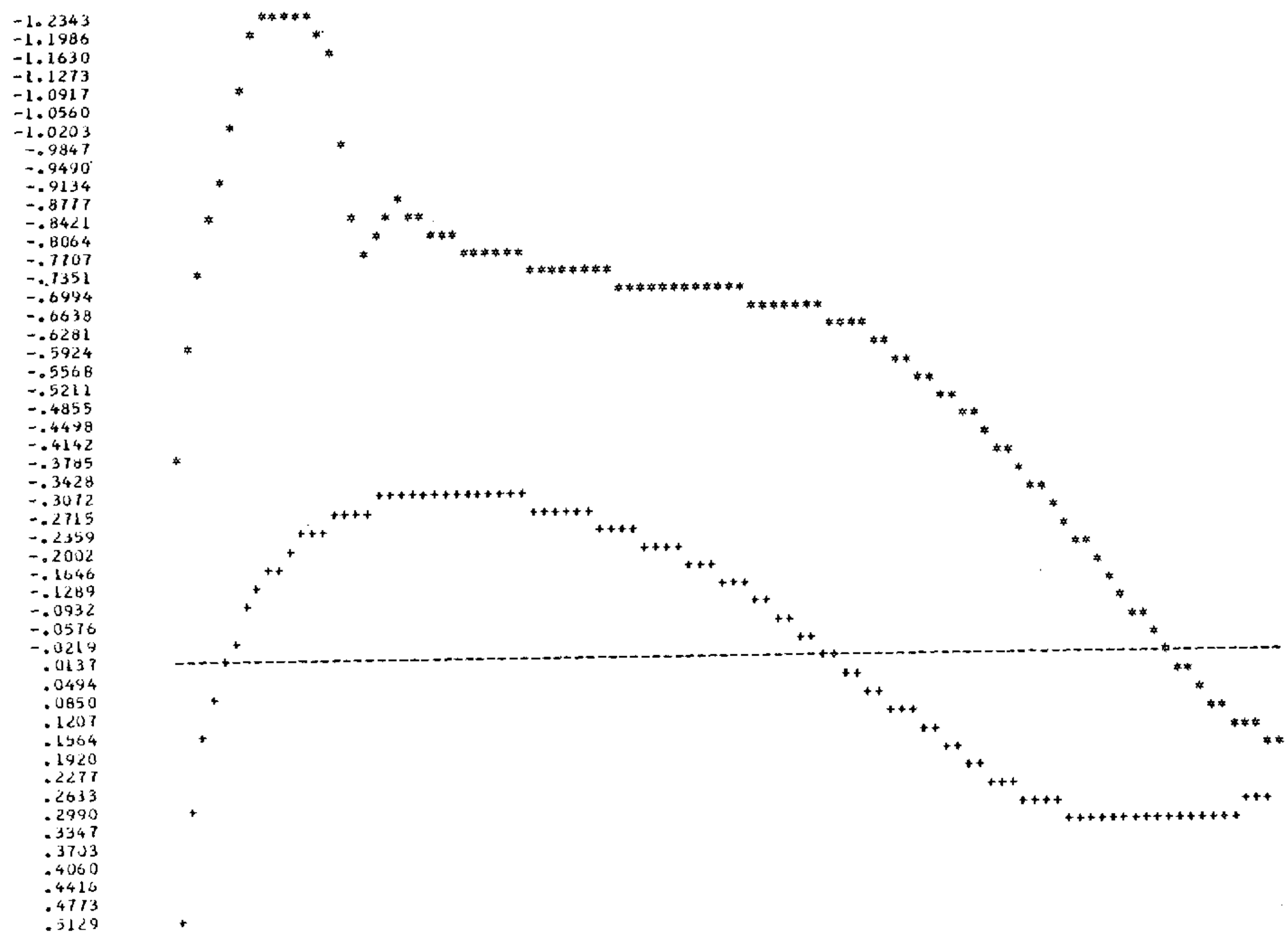
APPENDIX

PRESSURE DISTRIBUTION VS EQUIVALENT AIRFOIL COORDINATES

X/C	Y/C	CP
1.000000	0.000000	.201169
.999543	.000047	.244899
.998164	.000187	.264341
.995850	.000409	.279081
.992594	.000704	.292719
.988389	.001055	.304729
.983235	.001446	.315018
.977132	.001860	.323736
.970088	.002280	.331584
.962109	.002688	.338547
.953205	.003065	.344619
.943390	.003392	.349764
.932681	.003646	.353940
.921100	.003806	.357103
.908670	.003852	.359199
.895420	.003760	.360148
.881383	.003509	.359824
.866595	.003077	.358035
.851101	.002441	.354501
.834948	.001582	.348857
.818194	.000483	.340659
.800901	-.000869	.329462
.783138	-.002476	.314856
.764976	-.004333	.296594
.746487	-.006423	.274644
.72773'	-.008717	.249216
.708793	-.011180	.220741
.689699	-.013771	.189817
.670501	-.016447	.157123
.651231	-.019165	.123340
.631916	-.021889	.089082
.612578	-.024583	.054886
.593234	-.027216	.021231
.573901	-.029764	-.011436
.554592	-.032204	-.042695
.535319	-.034519	-.072169
.516096	-.036695	-.099608
.496933	-.038723	-.124982
.477848	-.040597	-.148463
.458858	-.042313	-.170246
.439980	-.043866	-.190352
.421234	-.045253	-.208617
.402638	-.046470	-.224853
.384211	-.047517	-.238988
.365972	-.048394	-.251069
.347941	-.049101	-.261203
.330139	-.049641	-.269510
.312587	-.050018	-.276103
.295307	-.050232	-.281054
.278322	-.050287	-.284390
.261654	-.050187	-.286123
.245325	-.049934	-.286291
.229357	-.049531	-.284959
.213773	-.048981	-.282161
.198595	-.048288	-.277805
.183844	-.047454	-.271631
.169541	-.046479	-.263268
.155703	-.045367	-.252365
.142350	-.044120	-.238622
.129497	-.042742	-.221733
.117160	-.041236	-.201368
.105354	-.039608	-.177334
.094093	-.037864	-.149608
.083392	-.036011	-.118011
.073265	-.034055	-.081947
.063726	-.032005	-.040925
.054786	-.029870	.004889
.046462	-.027660	.055280
.038769	-.025386	.110784
.031726	-.023056	.172518
.025352	-.020678	.241677
.019667	-.018256	.320237
.014696	-.015789	.413087
.010459	-.013262	.529028
.006963	-.010650	.678021
.004190	-.007937	.850553
.002130	-.005127	1.013099
.000782	-.002205	1.117206
.000104	.000854	1.110560
.000000	.004019	.985257

.000431	.007170	.730289
.001468	.010208	.366201
.003212	.013116	.022219
.005718	.015940	-.202190
.008985	.018738	-.337272
.012978	.021540	-.438969
.017659	.024347	-.530189
.023002	.027144	-.616067
.028987	.029914	-.699118
.035600	.032634	-.777359
.042833	.035287	-.844362
.050678	.037859	-.908697
.059117	.040333	-.986702
.068141	.042674	-1.074912
.077770	.044846	-1.153852
.088038	.046843	-1.207015
.098970	.048691	-1.231194
.110563	.050428	-1.234446
.122788	.052085	-1.227483
.135607	.053671	-1.213954
.148991	.055185	-1.165445
.162917	.056625	-.913638
.177358	.057999	-.736340
.192279	.059308	-.798488
.207651	.060540	-.843860
.223449	.061688	-.839522
.239656	.062750	-.801599
.256245	.063729	-.775589
.273188	.064625	-.765135
.290454	.065435	-.755229
.308015	.066156	-.745581
.325844	.066785	-.735901
.343913	.067318	-.726452
.362195	.067756	-.717485
.380660	.068096	-.709109
.399280	.068336	-.701376
.418024	.068476	-.694318
.436863	.068513	-.687923
.455767	.068446	-.682113
.474705	.068270	-.676746
.493648	.067982	-.671609
.512566	.067580	-.666393
.531430	.067057	-.660660
.550213	.066409	-.653842
.568890	.065630	-.645270
.587437	.064715	-.634250
.605831	.063658	-.620129
.624054	.062457	-.602353
.642088	.061108	-.580543
.659918	.059612	-.554623
.677531	.057973	-.524909
.694912	.056196	-.492069
.712047	.054293	-.456911
.728922	.052274	-.420148
.745518	.050154	-.382255
.761818	.047946	-.343507
.777805	.045665	-.304089
.793459	.043329	-.264199
.808760	.040955	-.224084
.823689	.038562	-.184057
.838223	.036170	-.144476
.852340	.033800	-.105727
.866012	.031475	-.068168
.879212	.029213	-.032054
.891912	.027036	.002475
.904081	.024962	.035402
.915688	.023009	.066669
.926700	.021195	.095977
.937083	.019536	.122626
.946799	.018048	.145593
.955803	.016741	.163877
.964052	.015616	.176958
.971506	.014664	.185139
.978129	.013871	.189539
.983897	.013218	.191801
.988795	.012687	.193567
.992814	.012263	.196294
.995949	.011939	.198421
.998195	.011709	.199945
.999547	.011571	.200863
.999999	.011525	.201169

CL = .583   CM = -.107   CD = .C0338


-1.2343
-1.1986
-1.1630
-1.1273
-1.0917
-1.0560
-1.0203
-.9847
-.9490
-.9134
-.8777
-.8421
-.8064
-.7707
-.7351
-.6994
-.6638
-.6281
-.5924
-.5568
-.5211
-.4855
-.4498
-.4142
-.3785
-.3428
-.3072
-.2715
-.2359
-.2002
-.1646
-.1289
-.0932
-.0576
-.0219
.0137
.0494
.0850
.1207
.1564
.1920
.2277
.2633
.2990
.3347
.3703
.4060
.4416
.4773
.5129

APPENDIX

SUMMARY OF RUN 10 FOR KORN AIRFOIL (THEOR COORD WITH AXIS ROTATED 0.12 DEG)

MACH = .702 ALPHA = 1.10 RN X 10E-6 = 21.18 IFIX = 40

CALCULATIONS USING THE CRUDE GRID (CCNV)

CYCLE	CL	NCY	X/C UPP SEP	X/C LOW SEP
1	.722	144	.957	1.000
2	.591	68	.977	1.000
3	.591	8		

CALCULATIONS USING THE MEDIUM GRID (CONV)

CYCLE	CL	NCY	X/C UPP SEP	X/C LOW SEP
1	.590	19	.971	1.000
2	.585	16		

CALCULATIONS USING THE FINE GRID (CCNV)

CYCLE	CL	NCY	X/C UPP SEP	X/C LOW SEP
1	.584	15	.975	1.000
2	.583	3		

ONE CALCOMP PLOT IS GENERATED

```
03/27/74  LRC ICOPS PRODUCTN 6600C-131K 01/23/74A
09.54.07. ACCT - TOTAL EXCEEDED
09.54.07.TER1901.
09.54.07.JOB,1,0600,066500,5000.          A4314    R
09.54.07.1711         100710         B641         CENT
09.54.07.USER.BAVITZ,PAUL C.                      1724
09.54.07.045161 33100 NAS          GRUM
09.54.07.LINECNT(10000)
09.54.08.FETCH(A4314,SPRZ16,BINARY,,BAVITZ)
09.54.14.TIME         BG ATTACH
09.55.07.TIME         ED ATTACH
09.55.08.END      FETCH
09.55.09.COPYBR(INPUT,TAPE3)
09.55.11.SETINDF.
09.55.12.BAVITZ.
10.08.06.PLOTTING COMMENCED
10.08.27.LAST CALCOMP BLOCK ADDRESS =       1
10.08.27.    DATA PLOTTED =           9060
10.08.28.STOP 0101
10.08.28.SPPRINT(OUTPUT,3)
10.08.30.RFL,10000.
10.08.30.MEMORY  010000 CM
10.08.30.RFL        0000355 O/S CALLS
10.08.30.RFL CPU    200.519090 SEC.
10.08.31.REWIND(CALTPE)
10.11.51.TX102 ASSIGN57,PLT218
10.11.53.REQUEST,TAPE97,HI,X.     CALTP,RIM,PCB,A3
10.11.53.275,B 641
10.11.53. (57 ASSIGNED)
10.11.53.REWIND(TAPE97)
10.11.55.COPYBF(CALTPE,TAPE97)
10.11.57.DROPFIL(TAPE97)
10.11.59.       0000381 O/S CALLS
10.11.59.CPU   200.532222 SEC.
10.11.59.PPU   188.895232 SEC.
10.12.00.COST OF THIS JOB WAS $    17
10.12.00.KWH      4.22      KILOWORD HOURS
11.42.21. TER1901.    1066 LINES PRINTED. LP21
```

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX

User Particulars

Convergence.- In some instances, the inviscid-flow—boundary-layer iterations with the crude and medium grids, or at least the medium grid, will converge, although the iterations with the fine grid will not. The convergence criterion associated with the number of cycles of an inviscid computation was introduced to prevent this convergence, but the current requirement of 20 cycles or less occasionally is not adequate. Sometimes the limit of four inviscid solutions with the fine grid is reached without convergence of the iterations. In almost all these cases, however, the first solution with the fine grid (which is just a refinement of the last converged solution with the medium grid) is acceptable. In fact, some of the data correlations in this paper fall in this category. Therefore, as a rule of thumb, use the first of the four pressure distributions provided they are all reasonably similar and provided the inviscid-flow—boundary-layer iterations using the medium grid have converged. A reduction in the maximum number of fine-grid computations to 1 or 2 was contemplated to save computer time; but in rare instances all four of the aforementioned pressure distributions are not similar and thus the results cannot be used. The program can be modified, if desired, to increase the allowable number of inviscid computational cycles associated with overall convergence (from 20 to about 23 or 24), or to raise the tolerance level on the inviscid solution (to redefine ST from 5×10^{-5} to approximately 10^{-4}).

At high transonic Mach numbers, when the shock wave is very near the trailing edge, the initial inviscid solution currently diverges. As previously mentioned, an arbitrary boundary layer is introduced in an attempt to increase the number of attainable computation points with disappointing results. It should be noted that although a minor error is suspected in the version of the program which is herein listed, an additional error definitely exists in the overlaid version. Instead of starting from an incompressible flow definition, the second inviscid calculation definitely starts from the last cycle of the first inviscid calculation. Thus, it too diverges. This whole process probably should be deleted from the computer program. Additional computation points at the high Mach numbers can be attained, however, by reducing the value of the term associated with the artificial viscosity EP. It is currently defined as 0.7. For cases where the initial inviscid solution would diverge, a value of 0.0 has successfully been used. (EP can vary from 0.0, which corresponds to second-order accuracy at those points.) Also, the program can be modified to start the calculations at a lower Mach number or angle of attack where the first solution will not diverge, and then to increase to the desired level after the boundary layer has been incorporated, for example, on the second or third iteration.

Effect of IFIX.- This empirical factor can be input or automatically selected. To illustrate the typical effects it has on the pressure distribution, results on a supercritical airfoil are shown in figure 21 for the automatically selected value (IFIX = 37) and for a specified value corresponding to no internal adjustment of the aft surface pressure distribution (IFIX = 41). Initially, IFIX was intended to vary with thickness ratio as well as trailing-edge slope, but consistent data over a range of thickness ratios were not available. Recent calculations for thin supercritical airfoils have indicated that the present empirical formula apparently produces values of IFIX which are slightly low. This fact can be deduced by examining the behavior of the upper surface pressure distribution near the trailing edge, where an atypical reversal from an unfavorable pressure gradient to a favorable one occurs.

Insufficient coordinate definition.- Inadequate definition of the airfoil, particularly near the nose, or incorrect definition of a particular coordinate can cause the program to terminate with no message. Thus, special note is being made here. After the first line of output, the program exits from subroutine AIRFOL. Additionally, there should be no coordinate input aft of the 99-percent chord, except for the trailing-edge coordinate, because an error can arise in the definition of the equivalent inviscid airfoil in this region.

Program Listing

```
      PROGRAM FLOW1(INPUT=66,OUTPUT=500,TAPE3=500,TAPE6=OUTPUT,TAPE5=INP
     1UT)
C     ANALYSIS OF TWO-DIMENSIONAL, TRANSONIC, VISCOUS FLOW
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN,NCASE
      COMMON /E/ KCYCLE,FNU,FNL,IBNDLAY(13),ABNDLAY(19),XBODY(162),YBODY
     1(162),BODSLOP(162),BODCURV(82),XNEW(162),YNEW(162),CPSURF(162),CPB
     2DLY(82),IGRID,GRID,XUPLE(20),XUPARC(20),XLOLE(20),XLOARC(20),TITLO
     3UT(15),CLOUT(3,7),SUPOUT(3,7),SLOWOUT(3,7),CMOUT,CPUP(85),CPLO(85)
     4,XTEMUP(85),XTEMLO(85),DELBLX(162),KOUNT,KOUNTUP,LOWGRD,INVDIV
      COMMON /F/ XBL(75),DELBL(75),THETBL(75),HBL(75),CFRBL(75),KTYPE,KK
     1K
      COMMON /G/ SS(310),TH(310),U(310),V(310),W(310),SP(310)
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),R1(41),S1(162),CO(162),Z(162),FM(162),PHIR(162)
      DIMENSION KWRIT(3), CONVOUT(3,2), NCYOUT(3,7), COME(1838), COMF(37
     17), COMC(86), XTM(101), DELTM(101), XPLOT(100), DELBLXF(100,2), CP
     2PLOT(100,2), PYS(2)
      EQUIVALENCE (COME(1),KCYCLE)
      EQUIVALENCE (COMF(1),XBL(1))
      EQUIVALENCE (COMC(1),M)
      KONTROL=0
      IRUN=1
      KPLOT=0
      DATA GAM,IMO/1.4,0/
      DATA (PYS(J),J=1,2)/1H+,1H*/
      N2=6
      N3=3
      N4=6
      N5=5
      M4=N4
      DO 20 K1=1,3
      DO 20 K2=1,7
      SUPOUT(K1,K2)=1.0
```

```
   20 SLOWOUT(K1,K2)=1.0
      KCYCLE=1
      IGRID=0
C     IBNDLAY 1 THRU 11 AND ABNDLAY 1 THRU 17 ARE INPUTS FOR THE
C     BOUNDARY LAYER ROUTINE
      IBNDLAY(1)=1
      IBNDLAY(2)=500
      IBNDLAY(3)=5000
      IBNDLAY(4)=20
      IBNDLAY(5)=41
      IBNDLAY(6)=10
      IBNDLAY(7)=1
      IBNDLAY(8)=4
      IBNDLAY(9)=0
      IBNDLAY(10)=2
      IBNDLAY(11)=1
      IBNDLAY(12)=7
      ABNDLAY(2)=0.025
      ABNDLAY(3)=0.0
      ABNDLAY(5)=0.0015
      ABNDLAY(6)=0.0
      ABNDLAY(7)=0.001
      ABNDLAY(8)=0.15
      ABNDLAY(9)=0.40
      ABNDLAY(10)=2.0
      ABNDLAY(11)=1.0
      ABNDLAY(12)=1.4
      ABNDLAY(13)=0.89
      ABNDLAY(14)=0.76
      ABNDLAY(15)=0.70
      ABNDLAY(16)=0.01
      ABNDLAY(17)=0.0005
      F15=FLOAT(IBNDLAY(5))
   30 KSTOP=3
      ISKIP=0
      IF (IRUN.GT.1) READ (N3) COME,COMC,PHI,FP,AA,BB,A,B,C,D,E,RHO,RP,R
     1,RS,RI,SI,CO,Z,FM,PHIR
      READ (N5,940) FNRN,EM,ALP,ABNDLAY(18),FIFIX
      NRN=FNRN
      IBNDLAY(13)=FIFIX
      KMAXCYC=IBNDLAY(12)
   40 CONTINUE
      KONTROL=KONTROL+1
      GO TO (140,50,60,70,90,100,110,120,130), KONTROL
   50 NS=-1
      ITYP=1
      GO TO 140
   60 NS=-1
      ITYP=1
      GO TO 140
   70 IF (NCASE.GT.1.AND.IRUN.EQ.1) WRITE (N3) COME,COMC,PHI,FP,AA,BB,A,
     1B,C,D,E,RHO,RP,R,RS,RI,SI,CO,Z,FM,PHIR
      IF (FIFIX.GT.1.0) GO TO 80
C     DEFINE PARAMETER FOR T. E. BOUNDARY LAYER ADJUSTMENT IF NOT ON
C     THE INPUT CARD
      IF (TESLOP .GE. -12.0) FIFIX = 41.0 + (TESLOP + 2.0)/20.0
      IF (TESLOP .LE. -15.0) FIFIX = 39.5 + (TESLOP + 15.0)/2.0
      IF (TESLOP .LT. -12.0 .AND. TESLOP .GT. -15.0) FIFIX =
     1 -7.0*TESLOP**3/540.0 - 0.6*TESLOP**2 - 8.75*TESLOP - 0.5
      IBNDLAY(13)=FIFIX
   80 NS=800
      ITYP=4
      GO TO 140
   90 NS=1
      ITYP=-1
      GO TO 140
  100 NS=500
      ITYP=4
      GO TO 140
```

```
  110 NS=1
      ITYP=-1
      GO TO 140
  120 NS=300
      ITYP=4
      GO TO 140
  130 NS=0
      ITYP=0
  140 ALP=ALP/RAD
      IF (NS.EQ.0) GO TO 340
C     COMPUTE CONSTANTS NEEDED IN CALCULATION
      RA7=1.+EP
      RA8=1.+3.*EP
      RA9=1.+RA8
      C3=2.+EP
      EL=2.*RA7
      IF (EM.EQ.EMO) GO TO 150
C     NEW MACH NUMBER. ADJUST CONSTANTS WHICH DEPEND ON MACH NUMBER
      EMO=AMAX1(EM,.1E-40)
      EM=EMO
      C2=.5*(GAM-1.)
      C1=C2+1./(EM*EM)
      C5=1./(.5*GAM*EM*EM)
      C6=C2*EM*EM
      C4=C6+1.
      C7=1./(C5*C6)
      BET=SQRT(1.-EM*EM)
      IMO=1
  150 QCRIT=2.*C1/(GAM+1.)
      IF (NS.GT.0) GO TO 160
      NS=0
      IF (ITYP.GT.0) CALL CRUDER
      GO TO 250
  160 IF (ITYP) 170,180,190
  170 CALL REFINE
      GO TO 250
C     SET UP CONSTANTS AND DO CONFORMAL MAPPING
  180 CALL AIRFOL
      TESLOP = (RAD*(2.0*TH(NMP+1) + TH(1)) + 360.0)/3.0
      CALL RESTRT
      GO TO 250
  190 IF (IMO.LE.0) GO TO 200
      IMO=0
      CALL PHIRR
  200 CONTINUE
C     CHECK TO SEE IF ANGLE OF ATTACK HAS CHANGED
      IF (ABS(ALP-ALPO).GT.1.E-8) CALL SICO
      Y=(XS-XM)/(1.-QCRIT)
      YM=XS-Y
      IF (XPHI.EQ.0.) YA=YA/(2.*CHD)
C     COMPUTE INVISCID FLOW WITH A MAXIMUM OF NS CYCLES
  210 NCY=0
      INVDIV=0
      DO 240 K=1,NS
      CL=2.*DPHI*CHD
      CALL SWEEP
C     CHECK FOR CONVERGENCE OF INVISCID SOLUTION
      IF (AMAX1(YR,ABS(YA)).GE.ST) GO TO 220
      NCY=K
      GO TO 250
C     CHECK FOR DIVERGENCE
  220 IF (AMAX1(YR,ABS(YA)).LT.10.**10) GO TO 230
      NCY=K
      INVDIV=1
      GO TO 250
  230 YA=YA*XPHI
      IF (XPHI.NE.0.) YA=YA/(XPHI*XPHI)
  240 CONTINUE
      NCY=NS
```

```
  250 ALP=RAD*ALP
      ITYP=IABS(ITYP)
      NCYOUT(IGRID+1,KCYCLE)=NCY
      IF (INVDIV.EQ.0) GO TO 260
      IF (KCYCLE.EQ.1.AND.IGRID.EQ.0) GO TO 260
C     TERMINATE CASE IF INVISCID SOLUTION DIVERGES ON ANY CYCLE OTHER
C     THAN THE FIRST WITH THE CRUDE GRID
      CLOUT(IGRID+1,KCYCLE)=999.0
      CONVOUT(IGRID+1,1)=GRID
      CONVOUT(IGRID+1,2)=8HNOT CONV
      WRITE (N3) COME,COMC,Z
      KWRIT(IGRID+1)=KCYCLE-1
      KSTOP=IGRID+1
      GO TO 340
  260 IF (ITYP.GE.2) CALL GETCP
      IF (ITYP.LT.2) GO TO 40
C     CHECK FOR CONVERGENCE OF INVISCID-FLOW/BOUNDARY-LAYER ITERATIONS
      CALL CONVER (ICHECK,KCYCLE,M,CL,Z,CPSURF,NCY)
      IF (INVDIV.EQ.0) CLOUT(IGRID+1,KCYCLE)=CL
      IF (ICHECK.EQ.0) GO TO 270
      KWRITE=KCYCLE-1
      KCYCLE=KMAXCYC
  270 IF (INVDIV.EQ.0) CALL SETCP
      WRITE (N3) COME,COMC,Z
C     IF I-F/B-L ITERATIONS HAVE NOT CONVERGED AND KCYCLE HAS NOT
C     REACHED MAXIMUM VALUE, CONTINUE ITERATING
      IF (KCYCLE.LT.KMAXCYC) GO TO 300
      KCYCLE=1
      IF (ICHECK.EQ.1) GO TO 280
      KWRITE=KMAXCYC-1
      KWRIT(IGRID+1)=KWRITE
      CONVOUT(IGRID+1,1)=GRID
      CONVOUT(IGRID+1,2)=8HNOT CONV
      GO TO 290
  280 KWRIT(IGRID+1)=KWRITE
      CONVOUT(IGRID+1,1)=GRID
      CONVOUT(IGRID+1,2)=8H    CONV
  290 KMAXCYC=KMAXCYC/2+2
      IBNDLAY(12)=KMAXCYC
      IGRID=IGRID+1
C     REFINE GRID OR TERMINATE CASE
      GO TO 40
  300 ALP=ALP/RAD
C     COMPUTE BOUNDARY LAYER AND DEFINE NEW INVISCID AIRFOIL
      CALL BNDLAY
      KCYCLE=KCYCLE+1
      IF (IGRID.GT.0) GO TO 310
      CALL REFINE
  310 IF (IGRID.GT.1) GO TO 320
      CALL REFINE
C     MAP NEW INVISCID AIRFOIL (WITH FINE GRID)
  320 CALL AIRFOL
      CALL RESTRT
      CALL COSI
      IF (IGRID.GT.0) GO TO 330
      CALL CRUDER
  330 IF (IGRID.GT.1) GO TO 210
      CALL CRUDER
C     REPEAT INVISCID SOLUTION WITH NEW AIRFOIL
      GO TO 210
  340 ITYP=4
      IF (INVDIV.EQ.0) ALP=ALP*RAD
C     OUTPUT LAST INVISCID-FLOW/BOUNDARY-LAYER ITERATION IF OVERALL
C     SOLUTION CONVERGED, OTHERWISE OUTPUT ALL ITERATIONS
      WRITE (N2,950)
      WRITE (N4,1390) TITLOUT,NRN
      WRITE (N4,1270) EM,ALP,ABNDLAY(18),IBNDLAY(13)
      K1=FNU
      K2=FNL
```

```
      K12=K1+K2
      K2P1=K2+1
      REWIND N3
      IF (NCASE.GT.1) READ (N3) COME,COMC,PHI,FP,AA,BB,A,B,C,D,E,RHO,RP,
     1R,RS,RI,SI,CO,Z,FM,PHIR
      IF (INVDIV.EQ.1) GO TO 590
      IF (ICHECK.EQ.0) GO TO 590
C     OUTPUT LAST ITERATION
      DO 360 KK=1,2
      KWRTOUT=KWRIT(KK)
      DO 350 K=1,KWRTOUT
      READ (N3) COME,COMC,Z
      READ (N3) COMF
  350 READ (N3) COMF
  360 READ (N3) COME,COMC,Z
      LOOP=1
      KWRTOUT=KWRIT(3)
  370 READ (N3) COME,COMC,Z
      IF (LOOP.EQ.KWRTOUT) GO TO 380
      LOOP=LOOP+1
      READ (N3) COMF
      READ (N3) COMF
      GO TO 370
  380 WRITE (N4,960)
      WRITE (N4,970)
      WRITE (N4,980)
      WRITE (N4,990)
      I5=IBNDLAY(5)
      DO 390 K=1,I5
      XXXX=ABNDLAY(3)+FLOAT(K-1)*ABNDLAY(2)
      KPLUS=K+I5
  390 WRITE (N4,1000) XXXX,CPBDLY(KPLUS),BODCURV(KPLUS),CPBDLY(K),BODCUR
     1V(K)
      CDF=0.0
      DO 480 LOOP=1,2
      IF (LOOP.EQ.1) WRITE (N4,1010)
      IF (LOOP.EQ.2) WRITE (N4,1020)
      WRITE (N4,1030)
      READ (N3) COMF
      DO 400 K=2,KKK
  400 WRITE (N4,1040) XBL(K),DELBL(K),THETBL(K),HBL(K),CFRBL(K)
      XBL(KKK+1)=1.0
      CFRBL(KKK+1)=CFRBL(KKK)
      GO TO (410,420,430,440,450), KTYPE
  410 WRITE (N4,1050)
      GO TO 460
  420 WRITE (N4,1060)
      CFRBL(KKK+1)=0.0
      GO TO 460
  430 WRITE (N4,1070)
      GO TO 460
  440 WRITE (N4,1080)
      GO TO 460
  450 WRITE (N4,1090)
C     COMPUTE FRICTION DRAG (APPROX)
  460 DO 470 K=1,KKK
  470 CDF=CDF+0.5*(CFRBL(K+1)+CFRBL(K))*(XBL(K+1)-XBL(K))
  480 CONTINUE
      READ (N3) COME,COMC,Z
      DO 490 K=K2P1,K12
      KMN=K-K2
      XTM(KMN)=XBODY(K)
  490 DELTM(KMN)=DELBLX(K)
      DO 500 K=1,100
      XPLOT(K)=FLOAT(K)*0.01
      CALL DISCOT (XPLOT(K),XPLOT(K),XTM,DELTM,DELTM,-010,K1,0,DELBLXF(K
     1,2))
```

```
  500 DELBLXF(K,2)=100.0*DELBLXF(K,2)
      DO 510 K=1,K2
      XTM(K)=XBODY(K)
  510 DELTM(K)=DELBLX(K)
      DO 520 K=1,100
      CALL DISCOT (XPLOT(K),XPLOT(K),XTM,DELTM,DELTM,-010,K2,0,DELBLXF(K
     1,1))
  520 DELBLXF(K,1)=100.0*DELBLXF(K,1)
      CALL PLOTN (2,100,50,XPLOT,DELBLXF,PYS,2,100)
      WRITE (N4,1100)
      WRITE (N4,1110)
      WRITE (N4,1120)
      IF (K2.LE.K1) GO TO 530
      NMINIM=K1
      GO TO 540
  530 NMINIM=K2
  540 DO 550 K=1,NMINIM
      KPLS=K+K2
  550 WRITE (N4,1130) XBODY(KPLS),YBODY(KPLS),BODSLOP(KPLS),DELBLX(KPLS)
     1,XNEW(KPLS),YNEW(KPLS),XBODY(K),YBODY(K),BODSLOP(K),DELBLX(K),XNEW
     2(K),YNEW(K)
      NMIN1=NMINIM+1
      IF (K2.EQ.K1) GO TO 570
      IF (K2.LT.K1) GO TO 560
      WRITE (N4,1140) (XBODY(K),YBODY(K),BODSLOP(K),DELBLX(K),XNEW(K),YN
     1EW(K),K=NMIN1,K2)
      GO TO 570
  560 KST=K2+NMIN1
      WRITE (N4,1150) (XBODY(K),YBODY(K),BODSLOP(K),DELBLX(K),XNEW(K),YN
     1EW(K),K=KST,K12)
  570 WRITE (N4,1160)
      WRITE (N4,1170)
      WRITE (N4,1180) (Z(K),CPSURF(K),K=1,MM)
      KCYCLE=KWRIT(IGRID+1)+1
      CALL FORCES (CDF)
C     CL DEFINED FROM CIRCULATION EXCEPT FOR LAST CYCLE, WHICH USES
C     PRESSURE COEFFICIENT INTEGRATION
      CL1=CLOUT(IGRID+1,KCYCLE)
      WRITE (N4,1190) CL1,CMOUT,X
      DO 580 K=1,100
      CALL DISCOT (XPLOT(K),XPLOT(K),XTEMLO,CPLO,CPLO,-010,KOUNT,0,CPPLO
     1T(K,1))
      CPPLOT(K,1)=-CPPLOT(K,1)
      CALL DISCOT (XPLOT(K),XPLOT(K),XTEMUP,CPUP,CPUP,-010,KOUNTUP,0,CPP
     1LOT(K,2))
  580 CPPLOT(K,2)=-CPPLOT(K,2)
      NPLOT=-2
      CALL PLOTN (NPLOT,100,50,XPLOT,CPPLOT,PYS,2,100)
      CALL GRAFIC
      KPLOT=KPLOT+1
      GO TO 880
C     OUTPUT ALL ITERATIONS
  590 WRITE (N4,1200)
      DO 600 K=1,100
  600 XPLOT(K)=FLOAT(K)*0.01
      DO 860 KK=1,3
      IF (KK.EQ.1) GO TO 610
      KKMN=KK-1
      WRITE (N4,1240) CONVOUT(KKMN,1),CONVOUT(KKMN,2)
      WRITE (N4,1250) CONVOUT(KKMN,1),CONVOUT(KK,1)
  610 LOOP=1
  620 READ (N3) COME,COMC,Z
      IF (INVDIV.EQ.0) GO TO 640
      WRITE (N4,1340) LOOP,CONVOUT(KK,1)
      IF (LOOP.EQ.1.AND.IGRID.EQ.0) GO TO 630
      WRITE (N4,1350)
      GO TO 870
```

```
  630 WRITE (N4,1360)
      ISKIP=1
      READ (N3) COMF
      READ (N3) COMF
      LOOP=LOOP+1
      GO TO 620
  640 IF (LOOP.EQ.1) GO TO 740
      IF (ISKIP.EQ.1) GO TO 740
      DO 650 K=K2P1,K12
      KMN=K-K2
      XTM(KMN)=XBODY(K)
  650 DELTM(KMN)=DELBLX(K)
      DO 660 K=1,100
      CALL DISCOT (XPLOT(K),XPLOT(K),XTM,DELTM,DELTM,-010,K1,0,DELBLXF(K
     1,2))
  660 DELBLXF(K,2)=100.0*DELBLXF(K,2)
      DO 670 K=1,K2
      XTM(K)=XBODY(K)
  670 DELTM(K)=DELBLX(K)
      DO 680 K=1,100
      CALL DISCOT (XPLOT(K),XPLOT(K),XTM,DELTM,DELTM,-010,K2,0,DELBLXF(K
     1,1))
  680 DELBLXF(K,1)=100.0*DELBLXF(K,1)
      CALL PLOTN (2,100,50,XPLOT,DELBLXF,PYS,2,100)
      WRITE (N4,1100)
      WRITE (N4,1110)
      WRITE (N4,1120)
      IF (K2.LE.K1) GO TO 690
      NMINIM=K1
      GO TO 700
  690 NMINIM=K2
  700 DO 710 K=1,NMINIM
      KPLS=K+K2
  710 WRITE (N4,1130) XBODY(KPLS),YBODY(KPLS),BODSLOP(KPLS),DELBLX(KPLS)
     1,XNEW(KPLS),YNEW(KPLS),XBODY(K),YBODY(K),BODSLOP(K),DELBLX(K),XNEW
     2(K),YNEW(K)
      NMIN1=NMINIM+1
      IF (K2.EQ.K1) GO TO 730
      IF (K2.LT.K1) GO TO 720
      WRITE (N4,1140) (XBODY(K),YBODY(K),BODSLOP(K),DELBLX(K),XNEW(K),YN
     1EW(K),K=NMIN1,K2)
      GO TO 730
  720 KST=K2+NMIN1
      WRITE (N4,1150) (XBODY(K),YBODY(K),BODSLOP(K),DELBLX(K),XNEW(K),YN
     1EW(K),K=KST,K12)
  730 WRITE (N4,1160)
  740 WRITE (N4,1210) LOOP,CONVOUT(KK,1)
      ISKIP=0
      WRITE (N4,1220)
      WRITE (N4,1180) (Z(K),CPSURF(K),K=1,MM)
      KCYCLE=LOOP
      IF (KK.EQ.3) CALL FORCES (CDF)
C     CL DEFINED FROM CIRCULATION EXCEPT FOR CYCLES WITH FINE GRID,
C     WHICH USE PRESSURE COEFFICIENT INTEGRATION
      CL1=CLOUT(IGRID+1,KCYCLE)
      IF (KK.EQ.3) WRITE (N4,1190) CL1,CMOUT,X
      IF (KK.NE.3) WRITE (N4,1410) CL1
      DO 750 K=1,100
      CALL DISCOT (XPLOT(K),XPLOT(K),XTEMLO,CPLO,CPLO,-010,KOUNT,0,CPPLO
     1T(K,1))
      CPPLOT(K,1)=-CPPLOT(K,1)
      CALL DISCOT (XPLOT(K),XPLOT(K),XTEMUP,CPUP,CPUP,-010,KOUNTUP,0,CPP
     1LOT(K,2))
  750 CPPLOT(K,2)=-CPPLOT(K,2)
      NPLOT=-2
      CALL PLOTN (NPLOT,100,50,XPLOT,CPPLOT,PYS,2,100)
      IF (KK.EQ.3) CALL GRAFIC
      IF (KK.EQ.3) KPLOT=KPLOT+1
      IF (LOOP.GT.KWRIT(KK)) GO TO 860
```

```
      WRITE (N4,1230)
      WRITE (N4,980)
      WRITE (N4,990)
      I5=IBNDLAY(5)
      DO 760 K=1,I5
      XXXX=ABNDLAY(3)+FLOAT(K-1)*ABNDLAY(2)
      KPLUS=K+I5
  760 WRITE (N4,1000) XXXX,CPBDLY(KPLUS),BODCURV(KPLUS),CPBDLY(K),BODCUR
     1V(K)
      CDF=0.0
      DO 850 INLP1=1,2
      IF (INLP1.EQ.1) WRITE (N4,1010)
      IF (INLP1.EQ.2) WRITE (N4,1020)
      WRITE (N4,1030)
      READ (N3) COMF
      DO 770 INLP2=2,KKK
  770 WRITE (N4,1040) XBL(INLP2),DELBL(INLP2),THETBL(INLP2),HBL(INLP2),C
     1FRBL(INLP2)
      XBL(KKK+1)=1.0
      CFRBL(KKK+1)=CFRBL(KKK)
      GO TO (780,790,800,810,820), KTYPE
  780 WRITE (N4,1050)
      GO TO 830
  790 WRITE (N4,1060)
      CFRBL(KKK+1)=0.0
      GO TO 830
  800 WRITE (N4,1070)
      GO TO 830
  810 WRITE (N4,1080)
      GO TO 830
  820 WRITE (N4,1090)
  830 IF (KK.EQ.1) GO TO 850
      IF (KK.EQ.2.AND.LOOP.NE.KWRIT(2)) GO TO 850
C     COMPUTE FRICTION DRAG (APPROX)
      DO 840 K=1,KKK
  840 CDF=CDF+0.5*(CFRBL(K+1)+CFRBL(K))*(XBL(K+1)-XBL(K))
  850 CONTINUE
      LOOP=LOOP+1
      GO TO 620
  860 CONTINUE
      WRITE (N4,1240) CONVOUT(3,1),CONVOUT(3,2)
  870 WRITE (N4,950)
C     OUTPUT SUMMARY OF OVERALL ITERATION (CL DEFINED FROM
C     CIRCULATION EXCEPT FOR LAST CYCLE)
  880 WRITE (N4,1260) NRN,TITLOUT
      WRITE (N4,1270) EM,ALP,ABNDLAY(18),IBNDLAY(13)
      DO 900 K=1,KSTOP
      WRITE (N4,1280) CONVOUT(K,1),CONVOUT(K,2)
      WRITE (N4,1290)
      KWRTOUT=KWRIT(K)
      DO 890 KK=1,KWRTOUT
      WRITE (N4,1300) KK,CLOUT(K,KK),NCYOUT(K,KK)
  890 WRITE (N4,1310) SUPOUT(K,KK),SLOWOUT(K,KK)
      KWRTOUT=KWRTOUT+1
  900 WRITE (N4,1300) KWRTOUT,CLOUT(K,KWRTOUT),NCYOUT(K,KWRTOUT)
      IF (CLOUT(1,1).GT.990.0) WRITE (N4,1360)
      IF (CLOUT(KSTOP,KWRTOUT).LT.990.0) GO TO 910
      WRITE (N4,1370)
      IF (KSTOP.LT.3) KWRTOUT=1
      KWRTOUT=KWRTOUT-1
      WRITE (N4,1380) KWRTOUT
      GO TO 920
  910 IF (ICHECK.EQ.1) WRITE (N4,1320)
      IF (ICHECK.EQ.0) WRITE (N4,1330)
  920 IF (NCASE.EQ.IRUN) GO TO 930
      REWIND N3
      IRUN=IRUN+1
      KONTROL=3
C     GO TO NEXT CASE
```

```
      GO TO 30
C     TERMINATE PLOT
  930 IF (KPLOT.GT.0) CALL CPLOT ((0.,0.),999)
      IF (NCASE.GT.1) WRITE (N4,1400) KPLOT
      CALL EXIT
C
C
  940 FORMAT(5F10.0)
  950 FORMAT(1H1)
  960 FORMAT (//////////////////////35X60H********************************
     1***************************/35X60H*      THE INVISCID-FLOW / BOUND
     2ARY-LAYER ITERATION HAS      */35X60H* CONVERGED AND THE RESULTS FO
     3R THE FINAL ITERATION FOLLOW */35X60H*****************************
     4********************************)
  970 FORMAT(1H168HINPUT FOR BOUNDARY LAYER CALCULATION FROM PREVIOUS IN
     1VISCID SOLUTION)
  980 FORMAT(1H016X25H***** UPPER SURFACE *****10X25H***** LOWER SURFACE
     1 *****)
  990 FORMAT(1H05H  X/C,15X,2HCP,12X4HCURV,17X2HCP,12X4HCURV)
 1000 FORMAT(F6.3,5X,2F15.6,5X,2F15.6)
 1010 FORMAT(1H128HUPPER SURFACE BOUNDARY LAYER)
 1020 FORMAT(1H128HLOWER SURFACE BOUNDARY LAYER)
 1030 FORMAT(1H05H  X/C,8X,9HDELSTAR/C,7X,8HMOMNTM/C,10X,1HH,11X,2HCF)
 1040 FORMAT(F6.3,F15.5,F15.4,F15.3,E15.3)
 1050 FORMAT(1H013HRUN COMPLETED)
 1060 FORMAT(1H010HSEPARATION)
 1070 FORMAT(1H021HAAT IS LESS THAN -1.0)
 1080 FORMAT(1H015HP2 IS IMAGINARY)
 1090 FORMAT(1H015HTOR IS NEGATIVE)
 1100 FORMAT(1H093H                                    ADDITION OF BOUND
     1ARY LAYER DELSTAR TO AIRFOIL COORDINATES)
 1110 FORMAT(1H0112H                      ***** UPPER SURFACE *****
     1                                      ***** LOWER SURFACE *****)
 1120 FORMAT(9H0  X-BODY,10H    Y-BODY,10H     THETA,11H     DELSTAR,9H
     1  X-NEW,10H     Y-NEW,20H                X-BODY,10H    Y-BODY,10H
     2  THETA,11H    DELSTAR,9H    X-NEW,10H     Y-NEW)
 1130 FORMAT(2F10.6,F10.3,3F10.6,10X,2F10.6,F10.3,3F10.6)
 1140 FORMAT(70X,2F10.6,F10.3,3F10.6)
 1150 FORMAT(2F10.6,F10.3,3F10.6)
 1160 FORMAT(1H086HNOTE THAT THE TRAILING EDGE OF THE EQUIVALENT AIRFOIL
     1 IS REDEFINED SO THAT X/C = 1.000)
 1170 FORMAT(1H155HPRESSURE DISTRIBUTION VS EQUIVALENT AIRFOIL COORDINAT
     1ES//5X,3HX/C,7X,3HY/C,7X,2HCP)
 1180 FORMAT(3F10.6)
 1190 FORMAT(1H05HCL = ,F5.3,5X,5HCM = ,F5.3,5X,5HCD = ,F7.5)
 1200 FORMAT(//////////////////////37X56H*********************************
     1***********************/37X56H* THE INVISCID-FLOW / BOUNDARY-LAYER
     2 ITERATION HAS NOT */37X56H* CONVERGED AND THE RESULTS FOR ALL ITE
     3RATIONS FOLLOW  */37X56H******************************************
     4**************)
 1210 FORMAT(1H124HINVISCID SOLUTION NUMBER,I2,10H WITH THE ,A6,5H GRID)
 1220 FORMAT(1H055HPRESSURE DISTRIBUTION VS EQUIVALENT AIRFOIL COORDINAT
     1ES//5X,3HX/C,7X,3HY/C,7X,2HCP)
 1230 FORMAT(1H068HINPUT FOR BOUNDARY LAYER CALCULATION FROM PREVIOUS IN
     1VISCID SOLUTION)
 1240 FORMAT(//////////////////////56H0THE INVISCID-FLOW / BOUNDARY-LAYER
     1ITERATIONS WITH THE ,A6,10H GRID ARE ,A8)
 1250 FORMAT(//28H0THE GRID IS REFINED FROM A ,A6,11H MESH TO A ,A6,5H M
     1ESH)
 1260 FORMAT(1H014HSUMMARY OF RUN,I4,4H FOR,15A4)
 1270 FORMAT(1H06HMACH =,F5.3,5X,7HALPHA =,F6.2,5X,12HRN X 10E-6 =,F6.2,
     15X,6HIFIX =,I3)
 1280 FORMAT(1H023HCALCULATIONS USING THE ,A6,7H GRID (,A8,1H))
 1290 FORMAT(1H05HCYCLE,12X,2HCL,11X,3HNCY,7X,11HX/C UPP SEP,4X,11HX/C L
     1OW SEP/)
 1300 FORMAT(I4,11X,F6.3,10X,I3)
 1310 FORMAT(44X,F5.3,10X,F5.3)
 1320 FORMAT(1H029HONE CALCOMP PLOT IS GENERATED)
 1330 FORMAT(1H032HFOUR CALCOMP PLOTS ARE GENERATED)
```

```
1340 FORMAT(1H124HINVISCID SOLUTION NUMBER,I2,10H WITH THE ,A6,18H GRID
    1 HAS DIVERGED)
1350 FORMAT(1H060HTHE INVISCID-FLOW / BOUNDARY-LAYER ITERATIONS ARE TER
    1MINATED)
1360 FORMAT(1H076HA NOMINAL BOUNDARY LAYER IS USED TO RESTART THE SOLUT
    1ION ON THE SECOND CYCLE)
1370 FORMAT(1H048HTHE INVISCID SOLUTION DIVERGED ON THE LAST CYCLE)
1380 FORMAT(1H09HTHERE ARE,I2,24H CALCOMP PLOTS GENERATED)
1390 FORMAT(15A4,4HRUN ,I3)
1400 FORMAT(1H111HA TOTAL OF ,I2,41H CALCOMP PLOTS ARE GENERATED FOR TH
    1IS JOB)
1410 FORMAT(1H05HCL = ,F5.3)
      END
      SUBROUTINE AIRFOL
C     READS IN DATA FOR AIRFOIL AND MAKES INITIAL GUESS FOR MAPPING
C     FUNCTION BY COMPUTING FOURIER COEFFICIENTS
C     IF ONLY X,Y COORDINATES ARE PRESCRIBED SMOOTHING IS DONE
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,F
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BO,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN,NCASE
      COMMON /E/ KCYCLE,FNU,FNL,IBNDLAY(13),ABNDLAY(19),XBODY(162),YBODY
     1(162),BODSLOP(162),BODCURV(82),XNEW(162),YNEW(162),CPSURF(162),CPB
     2DLY(82),IGRID,GRID,XUPLE(20),XUPARC(20),XLOLE(20),XLOARC(20),TITLO
     3UT(15)
      DIMENSION XXLO(155), XXUP(155), SUMLO(155), SUMUP(155), XXLOT(155)
     1, SUMLOT(155), XTECUR(3), TECUR(3)
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      DIMENSION XX(1), YY(1), CIRC(1), TT(1), DS(1), TITLE(15), CX(1), S
     1X(1)
      COMMON /G/ SS(310),TH(310),U(310),V(310),W(310),SP(310)
      EQUIVALENCE (XX(1),FP(1,2)), (YY(1),FP(1,5)), (CIRC(1),FP(1,9)), (
     1TT(1),FP(1,13)), (DS(1),FP(1,17)), (TITLE(1),FP(1,1)), (CX(1),FP(1
     2,21)), (SX(1),FP(1,25))
      SMOOTH(Q1,Q2,Q3,Q4,Q5)=.0625*(Q1+Q5+4.*(Q2+Q4)+6.*Q3)
      SMTH(Q1,Q2,Q3)=.25*(Q1+Q2+Q2+Q3)
      DIS(Q1)=(Q1-ERR)*((Q1-ERR)*(Q1-ERR)+CONST)
      TOL=1.E-6
      CONST=.2
      VAL=4HRUN
      XT=ABS(TE)
C     NMP IS THE NUMBER OF POINTS IN CIRCLE PLANE FOR FOURIER SERIES
      LC=NMP/2
      MC=NMP+1
      PILC=PI/FLOAT(LC)
      IF (KCYCLE.EQ.1) GO TO 30
C     REPLACE COORDINATES WITH NEWLY DEFINED EFFECTIVE INVISCID
C     AIRFOIL GEOMETRY
      NT=FNU+FNL-1.
      DO 10 I=NL,NT
      XX(I)=XNEW(I+1)
   10 YY(I)=YNEW(I+1)
      DO 20 I=1,NL
      J=NP-I
      XX(J)=XNEW(I)
   20 YY(J)=YNEW(I)
      XMIN=XX(NL)
      GO TO 70
   30 WRITE (N4,490)
      WRITE (N4,520)
      REWIND N3
      READ (N3,500) TITLE
      DO 40 I=1,15
   40 TITLOUT(I)=TITLE(I)
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
C     READ IN AIRFOIL DATA FROM CARDS
C     AND STORE ORIGINAL BODY COORDINATES
      READ (N3,510) FNU,FNL,FNCASE
      NCASE=FNCASE
      IF (NCASE.LT.1) NCASE=1
      EPSIL=1.1
      READ (N3,490)
      NT=FNU+FNL-1.
      NL=FNL
      NP=NL+1
      READ (N3,510) (XX(I),I=NL,NT)
      READ (N3,510) (YY(I),I=NL,NT)
      DO 50 I=NL,NT
      XBODY(I+1)=XX(I)
   50 YBODY(I+1)=YY(I)
      READ (N3,490)
      READ (N3,510) (XBODY(I),I=1,NL)
      READ (N3,510) (YBODY(I),I=1,NL)
      DO 60 I=1,NL
      J=NP-I
      XX(J)=XBODY(I)
   60 YY(J)=YBODY(I)
      XMIN=XX(NL)
      REWIND N3
C     DEFINE SLOPES SO THAT ARC LENGTHS CAN BE COMPUTED TO FIRST ORDER
   70 DO 80 I=1,NT
   80 TH(I)=0.
      SP(1)=0.
      SUM=0.
      DO 90 I=2,NT
      DUM=AMAX1(.1E-20,ABS(.5*(TH(I)-TH(I-1))))
      UP=XX(I)-XX(I-1)
      VP=YY(I)-YY(I-1)
      SUM=SUM+SQRT(UP*UP+VP*VP)*DUM/SIN(DUM)
   90 SP(I)=SUM
      ARC=SP(NT)
      SN=2./ARC
      SCALE=.25*ARC
      EE=.5*(1.-EPSIL)
      DO 100 L=1,NT
  100 SS(L)=ACOS(1.-SN*SP(L))
      SS(NT)=PI
      CALL SPLIF (NT,SS,XX,U,SP,W,1,0.,1,0.)
      CALL SPLIF (NT,SS,YY,V,TT,DS,1,0.,1,0.)
      IF (KCYCLE.NE.1) GO TO 110
      KKK=1
      UTEMP1=U(1)
      VTEMP1=V(1)
      UTEMP2=U(NT)
      VTEMP2=V(NT)
C     COMPUTE SLOPES OF ORIGINAL BODY
      GO TO 170
  110 DT=PI/FLOAT(NMP)
      ERR=SS(NL)
      DUM=DIS(0.)
      FAC=PI/(DIS(PI)-DUM)
      DO 120 L=1,MC
  120 CIRC(L)=FAC*(DIS(FLOAT(L-1)*DT)-DUM)
      CALL INTPL (NMP,CIRC,SX,SS,XX,U,SP,W)
      CALL INTPL (NMP,CIRC,CX,SS,YY,V,TT,DS)
      SX(MC)=XX(NT)
      CX(MC)=YY(NT)
      DO 130 L=2,MC
      XX(L)=SX(L)
  130 YY(L)=CX(L)
      GRID=6H CRUDE
      IF (IGRID.EQ.1) GRID=6HMEDIUM
      IF (IGRID.EQ.2) GRID=6H FINE
```

```
      IF (IS.EQ.0) GO TO 160
C     DO IS SMOOTHING ITERATIONS
      DO 150 K=1,IS
      XX(2)=SMTH(SX(1),SX(2),SX(3))
      YY(2)=SMTH(CX(1),CX(2),CX(3))
      XX(NMP)=SMTH(SX(MC),SX(NMP),SX(NMP-1))
      YY(NMP)=SMTH(CX(MC),CX(NMP),CX(NMP-1))
      DO 140 L=4,NMP
      XX(L-1)=SMOOTH(SX(L-3),SX(L-2),SX(L-1),SX(L),SX(L+1))
  140 YY(L-1)=SMOOTH(CX(L-3),CX(L-2),CX(L-1),CX(L),CX(L+1))
      DO 150 L=2,NMP
      SX(L)=XX(L)
  150 CX(L)=YY(L)
  160 NT=MC
      CALL SPLIF (NT,CIRC,XX,U,SP,W,1,0.,1,0.)
      CALL SPLIF (NT,CIRC,YY,V,TT,DS,1,0.,1,0.)
  170 U(1)=SP(1)
      V(1)=TT(1)
      U(NT)=SP(NT)
      V(NT)=TT(NT)
      DO 180 L=1,NT
      V(L)=-V(L)
  180 U(L)=-U(L)
C     COMPUTE SLOPES FROM VELOCITIES
      TH(1)=ATAN2(V(1),U(1))
      FAC=1.
      DO 200 I=2,NT
      TH(I)=ATAN2(V(I),U(I))
C     CHOOSE NEAREST BRANCH FOR THE ARCTANGENT
  190 IF (ABS(TH(I)-TH(I-1)).LT.1.) GO TO 200
      TH(I)=TH(I)-PI*FAC
      IF (ABS(TH(I)).LT.6.) GO TO 190
      IF (FAC.LT.0.) CALL EXIT
      FAC=-1.
      GO TO 190
  200 CONTINUE
      IF (KCYCLE.NE.1) GO TO 240
      IF (KKK.NE.1) GO TO 240
C     STORE ORIGINAL BODY SLOPES USING UNSMOOTHED COORDINATES
      DO 210 I=1,NL
      J=NP-I
      BODSLOP(I)=TH(J)*RAD
  210 TH(J)=0.0
      BODSLOP(NP)=BODSLOP(1)
      DO 220 I=NP,NT
      BODSLOP(I+1)=TH(I)*RAD
  220 TH(I)=0.0
      KKK=2
      DO 230 L=1,NT
      V(L)=-V(L)
  230 U(L)=-U(L)
      U(1)=UTEMP1
      V(1)=VTEMP1
      U(NT)=UTEMP2
      V(NT)=VTEMP2
      GO TO 110
  240 IF (EPSIL.GT.1.) EPSIL=(TH(1)-(PI+TH(NT)))/PI
      IF (KCYCLE.NE.1) EPSIL=(TH(1)-(PI+TH(NT)))/PI
C     COMPUTE ARC LENGTH TO FOURTH ORDER ACCURACY
      SP(1)=0.
      SUM=0.
      DO 250 I=2,NT
      DUM=AMAX1(.1E-20,ABS(.5*(TH(I)-TH(I-1))))
      UP=XX(I)-XX(I-1)
      VP=YY(I)-YY(I-1)
      SUM=SUM+SQRT(UP*UP+VP*VP)*DUM/SIN(DUM)
  250 SP(I)=SUM
      ARC=SP(NT)
      SN=2./ARC
```

```
      SCALF=.25*ARC
      EE=.5*(1.-EPSIL)
      DO 260 L=1,NT
  260 SS(L)=ACOS(1.-SN*SP(L))
      SS(NT)=PI
      CALL SPLIF (NT,SS,TH,U,V,W,3,0.,3,0.)
      IF (KCYCLE.GT.1) GO TO 360
      WRITE (N4,500) TITLE
      WRITE (N4,530) IS
C     STORE ORIGINAL BODY CURVATURES AND
C     PRINT OUT DATA ON THE AIRFOIL
      WRITE (N4,540)
      KKUP=0
      KKLO=0
      DO 280 L=1,NT
      VAL=TH(L)*RAD
      SUM=SN*U(L)/AMAX1(.1E-5,SIN(SS(L)))
      IF (VAL.LE.-90.0) GO TO 270
      KKLO=KKLO+1
      XXLO(KKLO)=XX(L)
      SUMLO(KKLO)=-SUM
      GO TO 280
  270 KKUP=KKUP+1
      XXUP(KKUP)=XX(L)
      SUMUP(KKUP)=-SUM
  280 WRITE (N4,550) XX(L),YY(L),SP(L),VAL,SUM,V(L),W(L)
C     STORE ARC LENGTH VS X/C NEAR THE NOSE
      ARCLE=(SP(KKLO)+SP(KKLO+1))/2.
      DO 290 L=1,20
      LPLS=KKLO+L
      LMNS=KKLO+1-L
      XUPLE(L)=XX(LPLS)
      XUPARC(L)=SP(LPLS)-ARCLE
      XLOLE(L)=XX(LMNS)
  290 XLOARC(L)=ARCLE-SP(LMNS)
C     REORDER LOWER SURFACE CURVATURES OF ORIGINAL BODY
      DO 300 L=1,KKLO
      SUMLOT(L)=SUMLO(L)
  300 XXLOT(L)=XXLO(L)
      DO 310 L=1,KKLO
      LM=KKLO-L+1
      SUMLO(LM)=SUMLOT(L)
  310 XXLO(LM)=XXLOT(L)
C     REDEFINE CURVATURE AT THE TRAILING EDGE
      KKUM1=KKUP-1
      KKUM3=KKUP-3
      DO 320 L=KKUM3,KKUM1
      LM=L-KKUM1+3
      XTECUR(LM)=XXUP(L)
  320 TECUR(LM)=SUMUP(L)
      CALL DISCOT (1.00,1.00,XTECUR,TECUR,TECUR,-010,3,0,SUMUP(KKUP))
      KKLM1=KKLO-1
      KKLM3=KKLO-3
      DO 330 L=KKLM3,KKLM1
      LM=L-KKLM1+3
      XTECUR(LM)=XXLO(L)
  330 TECUR(LM)=SUMLO(L)
      CALL DISCOT (1.00,1.00,XTECUR,TECUR,TECUR,-010,3,0,SUMLO(KKLO))
      NCURV=IBNDLAY(5)
C     INTERPOLATE FOR CURVATURES AT INPUT STATIONS FOR
C     BOUNDARY LAYER CALCULATION
      DO 350 J=1,NCURV
      XCURV=ABNDLAY(3)+FLOAT(J-1)*ABNDLAY(2)
      IF (XCURV.LE.0.0001) GO TO 340
      CALL DISCOT (XCURV,XCURV,XXLO,SUMLO,SUMLO,-010,KKLO,0,BODCURV(J))
      JPL=J+NCURV
      CALL DISCOT (XCURV,XCURV,XXUP,SUMUP,SUMUP,-010,KKUP,0,BODCURV(JPL)
     1)
```

```
      GO TO 350
  340 SUMLE=(SUMLO(1)+SUMUP(1))/2.
      BODCURV(J)=SUMLE
      JPL=J+NCURV
      BODCURV(JPL)=SUMLE
  350 CONTINUE
      WRITE (N4,560)
C     MAKE INITIAL GUESS OF ARC LENGTH AS A FUNCTION OF CIRCLE ANGLE
  360 DO 370 I=1,MC
      ANGL=FLOAT(I-1)*PILC
      CIRC(I)=ANGL
      CX(I)=COS(ANGL)
      SX(I)=SIN(ANGL)
      YY(I)=1.
      IF (EE.NE.0.) YY(I)=(2.-2.*CX(I))**EE
      FAC=SIGN(1.+CX(I),FLOAT(LC-I))
      SP(I)=ACOS(.5*FAC)
  370 XX(I)=SCALE*(2.-FAC)
C     DO AT MOST 100 ITERATIONS TO FIND THE FOURIER COEFFICIENTS
      DO 450 KCY=1,100
      CALL INTPL (NMP,SP,TT,SS,TH,U,V,W)
      DO 380 I=1,NMP
  380 TT(I)=TT(I)-.5*(1.+EPSIL)*(PI-CIRC(I))+.5*PI
C     COMPUTE THE FIRST NFC FOURIER COEFFICIENTS
      DO 400 I=1,NFC
      SUM=0.
      FAC=0.
      DO 390 L=1,NMP
      LT=1+MOD((L-1)*(I-1),NMP)
      SUM=SUM-TT(L)*CX(LT)
  390 FAC=FAC+TT(L)*SX(LT)
      BB(I)=SUM/FLOAT(LC)
  400 AA(I)=FAC/FLOAT(LC)
      BB(1)=.5*BB(1)
      BB(NFC)=.5*BB(NFC)
      DA=1.-EPSIL-AA(2)
      AA(2)=1.-EPSIL
C     ENSURE CLOSURE
C     COMPUTE THE CONJUGATE HARMONIC FUNCTION DS
      DO 420 I=1,NMP
      SUM=(1.-EPSIL)*CX(I)
      DO 410 K=3,NFC
      LT=1+MOD((K-1)*(I-1),NMP)
  410 SUM=SUM+AA(K)*CX(LT)+BB(K)*SX(LT)
  420 DS(I)=YY(I)*EXP(SUM)
      DS(MC)=DS(1)
      TT(1)=0.
      VAL=.5*PILC
C     INTEGRATE TO GET NEW ARC LENGTH
      DO 430 L=2,MC
  430 TT(L)=TT(L-1)+VAL*(DS(L)+DS(L-1))
      SCALE=ARC/TT(MC)
      ERR=0.
      DO 440 I=1,NMP
      VAL=SCALE*TT(I)
      DUM=ABS(XX(I)-VAL)
      ERR=AMAX1(ERR,DUM/ARC)
      SP(I)=ACOS(1.-SN*VAL)
  440 XX(I)=VAL
      IF (KCYCLE.EQ.1) WRITE (N4,590) ERR,DA,BB(2)
      IF (ERR.LT.TOL) GO TO 460
  450 CONTINUE
      IF (KCYCLE.EQ.1) WRITE (N4,570)
  460 AA(1)=ARC
      IF (KCYCLE.GT.1) GO TO 480
      WRITE (N4,580) EPSIL,NMP
      DO 470 L=1,NFC
  470 WRITE (N4,590) AA(L),BB(L)
  480 BB(1)=ALOG(SCALE)
      RETURN
```

```
C
C
  490 FORMAT (1H1)
  500 FORMAT (15A4)
  510 FORMAT (8F10.0)
  520 FORMAT(1H052HANALYSIS OF TWO-DIMENSIONAL, TRANSONIC, VISCOUS FLOW,
     113X,30HPAUL BAVITZ, GRUMMAN AEROSPACE/)
  530 FORMAT (10H0THERE ARE,I4,26H SMOOTHING ITERATIONS USED/)
  540 FORMAT (35H0AIRFOIL COORDINATES AND CURVATURES/1H0,6X,1HX,14X,1HY,
     19X,10HARC LENGTH,6X,5HTHETA,7X,5HKAPPA,10X,2HKP,11X,3HKPP//)
  550 FORMAT (F12.6,2F14.6,F14.3,F14.4,2E14.3)
  560 FORMAT (1H0,4X,3HERR,14X,2HDA,14X,2HDB//)
  570 FORMAT (32H FOURIER SERIES DID NOT CONVERGE)
  580 FORMAT (34H1MAPPING TO THE INSIDE OF A CIRCLE//3X11HDZ/DSIGMA =
     1 50H -(1/SIGMA**2)*(1-SIGMA)**(1-EPSIL)*(EXP(W(SIGMA))//3X,
     2 42HW(SIGMA) = SUM((A(N)+I*B(N))*SIGMA**(N-1))//3X,7HEPSIL =,
     3 F5.3,20X,I4,25H POINTS AROUND THE CIRCLE//7X4HA(N)10X4HB(N)//)
  590 FORMAT (3E15.6)
      END
      BLOCK DATA
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMMON /D/ SF,SIZE,ANG,XMAX,YMAX,XOR,YOR,PGSIZ
      DATA SF,SIZE,ANG,XMAX,YMAX,XOR,YOR,PGSIZ/1.0,.14,0.,11.,11.,0.,-3.
     10,9.0/
      DATA YR,YA,AQ,BQ,TE,XM,XPHI,EMO,PI,RAD,ITYP,MXP,IK,JK,NCY,JJ,NRN,I
     1I,NFC,NMP,M,N,NS,IDIM/4*0.,-1.,3*1.,3.14159265359,57.295779513,5*0
     2,2*1,130,16,300,160,32,400,162/
      DATA EP,ST,XS,KP,IZ,IS,FSYM/0.7,0.00005,1.4,10,130,10,4.0/
      END
      SUBROUTINE BNDLAY
C     COMPUTE BOUNDARY LAYER CHARACTERISTICS AND DEFINE AN EQUIVALENT
C     INVISCID AIRFOIL USING THE ORIGINAL AIRFOIL AND THE DISPLACEMENT
C     THICKNESS
      COMMON /C/ IDUMMY(20),DUMMY(15),EM
      COMMON /E/ KCYCLE,FNU,FNL,I(13),A(19),XBODY(162),YBODY(162),BODSLO
     1P(162),BODCURV(82),XNEW(162),YNEW(162),CPSURF(162),CPBDLY(82),IGRI
     2D,GRID,XUPLE(20),XUPARC(20),XLOLE(20),XLOARC(20),TITLOUT(15),CLOUT
     3(3,7),SUPOUT(3,7),SLOWOUT(3,7),CMOUT,CPUP(85),CPLO(85),XTEMUP(85),
     4XTEMLO(85),DELBLX(162),KOUNT,KOUNTUP,LOWGRD,INVDIV
      COMMON /F/ XBL(75),DELBL(75),THETBL(75),HBL(75),CFRBL(75),KTYPE,KK
     1K
      DIMENSION ITEMP(13), ATEMP(19), Z(1), U(110), TAU(110), UFUT(110),
     1 TFUT(110), TANA(110), TANB(110), TANAFU(110), TANBFU(110), V(110)
     2, W(110), WFUT(110), ROU(110), ROU2(110), AL(2,3), P(41), RD(41),
     3XXX(6), YYY(6), BBB(6), XLAST(4), DELLAST(4), COMF(377)
      EQUIVALENCE (COMF(1),XBL(1))
      N3=3
C     ROUTINE IS SET FOR TWO CYCLES IN ORDER TO CALCULATE BOTH UPPER AND
C     LOWER SURFACE B. L. CHARACTERISTICS
      NUMBER=2
      KSURF=1
      XTRAIL=1.0
      XLAST(1)=1.1
      XLAST(2)=1.1
      NT=FNU+FNL
      NU=FNU
      NL=FNL
      NL1=NL+1
      I15=I(5)
      IF (INVDIV.EQ.0) GO TO 40
C     NOMINAL B L DELSTAR FOR INIT INV SOL DIVERGENCE
      DO 20 J=NL1,NT
      IF (XBODY(J).GT.0.80) GO TO 10
      DELBLX(J)=XBODY(J)*0.003125
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      GO TO 20
   10 DELBLX(J)=0.121875*XBODY(J)*XBODY(J)-0.191875*XBODY(J)+0.078
   20 CONTINUE
      DO 30 J=1,NL
   30 DELBLX(J)=0.0
      WRITE (N3) COMF
      WRITE (N3) COMF
      GO TO 1340
C     STORE INPUT MATRICES FOR USE IN SECOND CYCLE
   40 DO 50 J=1,13
   50 ITEMP(J)=I(J)
      DO 60 J=1,19
   60 ATEMP(J)=A(J)
      NORUN=0
   70 KTSTSEP=0
      NCOUNT=0
      IO=1
      X=A(3)
      LIMIT=I(4)
      FP=1
      KKK=1
      XBL(1)=0.000
      DELBL(1)=0.000
      THETBL(1)=0.000
      HBL(1)=0.000
      CFRBL(1)=0.000
   80 CONTINUE
      WARN=0.
      I5=I(5)
      I6=I(6)
      I7=I(7)
      I8=I(8)
      I9=I(9)
      I10=I(10)
      I11=I(11)
      RI2=1.
      RI5=FLOAT(I5)
      RI6=FLOAT(I6)
      I61=I6+1
      I62=I6+2
      I64=I6+4
      RM=0.
      AY=A(13)*(A(12)-1.)
      IF (X-A(3)) 230,90,230
C     BOUNDARY VALUE INPUT
   90 TEMP=(1.+.5*(A(12)-1.)*EM**2.)**(A(12)/(1.-A(12)))
      DO 110 J=1,I5
      IF (KSURF.EQ.2) GO TO 100
      JPL=J+I5
      P(J)=TEMP*(1.+.5*A(12)*CPBDLY(JPL)*EM**2.)
      GO TO 110
  100 P(J)=TEMP*(1.+.5*A(12)*CPBDLY(J)*EM**2.)
  110 CONTINUE
      IF (I7.EQ.0) GO TO 140
      DO 130 J=1,I5
      IF (KSURF.EQ.2) GO TO 120
      JPL=J+I5
      RD(J)=BODCURV(JPL)
      GO TO 130
  120 RD(J)=BODCURV(J)
  130 CONTINUE
      GO TO 150
  140 RD(1)=0.
  150 Z(1)=0.
      PSTAT=ORDIN(P,A(2),X)
```

```
      VDLPDX=SLOPE(P,A(2),I(5),X)/ORDIN(P,A(2),X)
      RK2=2./(A(12)-1.)*(1.-ORDIN(P,A(2),X)**(1.-1./A(12)))
      RK=SQRT(RK2)
      RK3=AY*RK/(1.-.5*(A(12)-1.)*RK2)
      RK5=RK*RK3/2.
      RK4=(1.+RK5)/(1.+RK5/A(13))
      TAU0=A(5)*RK2*(1.+RK5)
C     SYNTHETIC STARTING PROFILE
      G=SQRT(A(5)*(1.+RK5))/A(9)
      T=0.1
      TTI=1.-15.*RK5/16.
  160 RE=A(4)/TTI/(1.+RK5)**(1.+A(14))
      B=1.-G*(2.+ALOG(A(9)*G*RE/T))
      TI=(G+.5*B-2.*G**2-0.375*B**2-1.59*B*G)/(1.+(49.-297.*G)/RE)
      HI=(G+.5*B)/TI+49./RE
      TTI=1.-RK5*(5.*HI-1.)*(HI-1.)/(3.*HI-1.)/(2.*HI-1.)
      IF (ABS(1.-TI/T).LE.0.005) GO TO 170
      T=TI
      GO TO 160
  170 DO 180 J=1,I6
      Y=FLOAT(J)/RI6
      U(J)=G*ALOG(Y)+1.-.5*B*(1.+COS(3.14159*Y))
      TAU(J)=(G/Y+1.5708*B*SIN(3.14159*Y))**2*(A(9)*Y)**2
      IF (Y.LE..2) TAU(J)=TAU(J)*(1.-3.7*Y**1.6)
      IF (Y.GT..2) TAU(J)=TAU(J)*1.58*EXP(-3.9*Y)
      W(J)=U(J)*RK/(RK4/AY-.5*U(J)**2*RK2)
      TAU(J)=TAU(J)/(1.+RK5)
      ROU(J)=U(J)*RK*(1.+.5*U(J)*RK*W(J))
  180 ROU2(J)=ROU(J)*RK*U(J)
C     A(1),A(4) IND A(6) ARE REASSIGNED BELOW
      A(4)=A(4)/A(1)/RK*(1.+RK5/A(13))**(1./(A(12)-1.)-A(14))
      RI16=1./RI6
      D1=1.-(5./6.*ROU(1)/RI6+SIMPSN(ROU,1,I6,RI16))/ROU(I6)
      TD=1.-D1-(5./7.*ROU2(1)/RI6+SIMPSN(ROU2,1,I6,RI16))/ROU2(I6)
      IF (RK.GT..06) GO TO 190
      TD=TI
  190 CONTINUE
      A(1)=A(1)/TD/RI6
      A(6)=A(1)*RI6*(1.-.005/G)
      DO 200 J=1,3
  200 TAU(J)=.25*FLOAT(J)*TAU(J)+(1.-.25*FLOAT(J))*(A(5)+.5*VDLPDX*(1.-.
     15*(A(12)-1.)*RK2)/RK2/A(12)*A(1)*FLOAT(J))
      TTITI=TTI*TI
      DO 210 J=I61,I62
      U(J)=RK
      TAU(J)=RK2*1.0E-10
  210 W(J)=RK3
      DO 220 J=1,I6
      U(J)=U(J)*RK
      W(J)=U(J)/(RK4/AY-U(J)**2/2.)
  220 TAU(J)=TAU(J)*RK2*(1.+RK5)/(1.+.5*U(J)*W(J))
      ALPHA=(TAU(1)*(1.+.5*U(1)*W(1))-TAU0)/A(1)
      BETA=0.
      GO TO 250
C     MOVES RECENTLY CALCULATED PROFILES INTO OLD PROFILE STORE
  230 DO 240 J=1,I62
      U(J)=UFUT(J)
      TAU(J)=TFUT(J)
      TANA(J)=TANAFU(J)
      TANB(J)=TANBFU(J)
      ROU(J)=UFUT(J)*(1.+.5*UFUT(J)*WFUT(J))
      W(J)=WFUT(J)
      ROU2(J)=ROU(J)*UFUT(J)
  240 CONTINUE
  250 IENT=INT(.25*A(6)/A(1))
C     TAU MAX FOR G
      IF (IENT.LT.1) IENT=1
      DO 270 J=IENT,I6
      IF (RM-TAU(J)) 260,270,270
```

```
  260 RM=TAU(J)
      L=J
  270 CONTINUE
      RM=RM/RK*SQRT((1.+.5*U(L)*W(L))/(1.+RK5))
      IF (X-A(3)) 380,280,380
  280 CONTINUE
      IF (I9-1) 290,300,310
  290 DIV=0.
      GO TO 340
  300 DIV=1./(X-ORDIN(Z,A(2),X))
      GO TO 340
  310 IF (I9-2) 340,320,330
  320 DIV=SLOPE(Z,A(2),I(5),X)/ORDIN(Z,A(2),X)
      GO TO 340
  330 DIV=ORDIN(Z,A(2),X)
  340 CONTINUE
C     V ETC FOR FIRST PROFILE
      V(1)=0.
      DO 350 J=1,161
      Y=1.+.25*(U(J+1)+U(J))*(W(J+1)+W(J))
  350 V(J+1)=(U(J+1)*V(J)-Y*(TAU(J+1)-TAU(J))-.5*(W(J+1)+W(J))*(TAU(J+1)
     1+TAU(J))*Y*(U(J+1)-U(J))*.5-.25*(U(J)+U(J+1))**2*A(1)*(VDLPDX*(1.-
     2Y*AY/A(12)/(Y-1.))+DIV))/U(J)
      DO 360 J=1,162
      ROU(J)=U(J)*(1.+.5*U(J)*W(J))
  360 ROU2(J)=ROU(J)*U(J)
      DO 370 J=1,162
  370 CALL TANCAL (G,A,V,J,RM,D,U,TAU,TANA,TANB,W)
  380 CONTINUE
      TAUO=TAUO
      K=INT(40./(A(4)*PSTAT*A(1)*SQRT(TAUO)))+1
      IF (K.LE.16) GO TO 390
      K=16
  390 PK=K
      DELTA1=R16*A(1)-(-46./A(4)/PSTAT+PK*A(1)*(ROU(K)-SQRT(TAUO)/A(9))+
     1SIMPSN(ROU,K,16,A(1)))/ROU(161)
      DELTA2=R16*A(1)-DELTA1-(-687.*SQRT(TAUO)/A(4)/PSTAT+PK*A(1)*(ROU2(
     1K)-2.*ROU(K)*SQRT(TAUO)/A(9)+2.*TAUO/A(9)**2)+SIMPSN(ROU2,K,16,A(1
     2)))/ROU2(161)
      H1=DELTA1/DELTA2
      CFR=2.*TAUO/RK2/(1.+RK5)
C     MOMENTUM INTEGRAL CHECK
      RMA2=RK2/(1.-.5*(A(12)-1.)*RK2)
      IF (X-A(3)) 400,400,410
  400 ORD1=.5*CFR+DELTA2*(VDLPDX*(H1+2.-RMA2)/(A(12)*RMA2)-DIV)
      THETA=DELTA2
      GO TO 420
  410 ORD2=.5*CFR+DELTA2*(VDLPDX*(H1+2.-RMA2)/(A(12)*RMA2)-DIV)
      THETA=THETA+.5*(ORD1+ORD2)*XSTEP
      ORD1=ORD2
  420 CONTINUE
      RMAX=0.
      IF (TANB(1)) 440,440,430
  430 IQ=1
      GO TO 590
C     COURANT FRIEDRICHS LEWY XSTEP CRITERION
  440 DO 480 J=1,16
      IF (RMAX-TANA(J)) 450,450,460
  450 RMAX=TANA(J)
      GO TO 480
  460 IF (RMAX+TANB(J)) 470,470,480
  470 RMAX=-TANB(J)
  480 CONTINUE
      XSTEP=A(11)*A(1)/RMAX
      IF (X+XSTEP-(R15-1.)*A(2)) 490,490,1140
  490 H=-XSTEP*TANB(1)/A(1)
      TORO=TAU(1)
      TOR=TAU(2)*H+TAU(1)*(1.-H)
      UL=U(2)*H+U(1)*(1.-H)
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
      TINT=TOR
      UINT=UL
      WW=W(2)*H+W(1)*(1.-H)
C     BOUNDARY CONDITION FOR SOLID SURFACE
      XXSTEP=X+XSTEP
      AZ=ORDIN(P,A(2),XXSTEP)
      B=AZ**(1./A(12)-1.)
      C=B/(A(13)*(B-1.)+1.)
      D=ALOG(A(1)*A(4)*AZ*C**(1.+A(14)))+A(10)
      E=2./A(13)/(A(12)-1.)/C
      UFUT(1)=U(1)+BETA*XSTEP
  500 F=WW*TOR/A(8)
      G=(A(15)+.5)*F-SQRT(((A(15)+.5)*F)**2+2.*TOR/A(8))
      XPDX2=X+.5*XSTEP
      DLPDX=SLOPE(P,A(2),I(5),XPDX2)/ORDIN(P,A(2),XPDX2)
      T=AY/A(12)*DLPDX
  510 ROOT=SQRT(TAU0)
      F4=TOR*(UFUT(1)+.5*G*TINT/TOR-UINT-XSTEP*(-DLPDX*A(13)*(1.-1./A(12
     1))/WW+A(8)*SQRT(TOR)*G/(UL*A(9)*(1.+.5*H)*A(1)))))/(-.5*G)
      AAT=ALPHA*A(1)/TAU0
      IF (AAT.LT.-1.0) GO TO 1100
      F1=UFUT(1)-ROOT/A(9)*(ALOG(ROOT)+D+FN(AAT))
      F5=1./(1.-UFUT(1)**2./E)
      F2=TAU0+ALPHA*A(1)+F4*F5
      F3=DLPDX/C/A(12)+.3*(UFUT(1)**2-U(1)**2)*(2.*F5+1.)/3./XSTEP-ALPHA
      F6=1./(1.+SQRT(1.+ALPHA*A(1)/TAU0))
      DF1DTW=-.5*UFUT(1)/TAU0-(.5-ALPHA*A(1)/TAU0*F6)/ROOT/A(9)
      DF1DA=-A(1)/A(9)/ROOT*F6
      F7=2.*UFUT(1)/(E-UFUT(1)**2)
      DF2DU=F5*(TOR/(-.5*G)+F4*F7)
      DF3DU=.2*F5*F7*(UFUT(1)**2-U(1)**2)/XSTEP+.2*UFUT(1)/XSTEP*(2.*F5+
     11.)
      DU=(DF1DTW*(F2+A(1)*F3)-F1-DF1DA*F3)/(1.-DF1DTW*(DF2DU+A(1)*DF3DU)
     1+DF1DA*DF3DU)
      DA=F3+DF3DU*DU
      DTW=-F2-DF2DU*DU-A(1)*DA
      UFUT(1)=UFUT(1)+DU
      TAU0=TAU0+DTW
      IF (TAU0) 520,520,530
  520 IQ=1
      GO TO 590
  530 ALPHA=ALPHA+DA
      A162=2.*A(16)
      IF (ABS(DU/UFUT(1))-A(16)) 540,540,510
  540 IF (ABS(DTW/TAU0)-A(16)) 550,550,510
  550 IF (ABS(DA/ALPHA)-A162) 560,560,510
  560 WFUT(1)=F7
      TFUT(1)=(TAU0+ALPHA*A(1))/(1.+.5*UFUT(1)*WFUT(1))
      IF (ABS(1.-TFUT(1)/TOR0)-2.*A(16)) 580,570,570
  570 H=-XSTEP/A(1)*(((A(15)-.5)*WFUT(1)*TFUT(1)-SQRT(((A(15)+.5)*WFUT(1
     1)*TFUT(1))**2+2.*A(8)*TFUT(1)))/UFUT(1))
      UINT=U(2)*H+U(1)*(1.-H)
      TINT=TAU(2)*H+TAU(1)*(1.-H)
      UL=.5*(UINT+UFUT(1))
      TOR=.5*(TINT+TFUT(1))
      WW=2.*UL/(E-UL**2)
      TOR0=TFUT(1)
      GO TO 500
  580 IQ=0
      BETA=(UFUT(1)-U(1))/XSTEP
      NP=0
  590 IF (NCOUNT-1) 600,660,600
  600 IF (NCOUNT-(NCOUNT/I(2))*I(2)) 610,660,610
  610 IF (IQ-1) 620,660,620
  620 IF ((X-EP*A(2))-A(3)) 630,630,640
  630 IF (CFR-A(7)) 650,680,680
  640 EP=EP+1.
  650 NP=1
  660 IF (X.LT.0.02) GO TO 670
      KKK=KKK+1
```

```
C       STORE B. L. CHARACTERISTICS FOR OUTPUT
        XBL(KKK)=X
        DELBL(KKK)=DELTA1
        THETBL(KKK)=DELTA2
        HBL(KKK)=H1
        CFRBL(KKK)=CFR
  670   CONTINUE
  680   NP=0
        IF (IQ-1) 690,1130,690
  690   IF (NCOUNT-I(3)) 700,1140,1140
  700   V(1)=-.5*UFUT(1)*A(1)*(TAU0/(A(5)*RK2*(1.+RK5))-1.)/XSTEP
        XFUT=X+XSTEP
        PSTAT=ORDIN(P,A(2),XFUT)
        VOLPDX=SLOPE(P,A(2),I(5),XFUT)/ORDIN(P,A(2),XFUT)
        RK2=2./(A(12)-1.)*(1.-ORDIN(P,A(2),XFUT)**(1.-1./A(12)))
        RK=SQRT(RK2)
        RK3=AY*RK/(1.-.5*(A(12)-1.)*RK2)
        RK5=RK*RK3/2.
        RK4=(1.+RK5)/(1.+RK5/A(13))
        IF (I7.EQ.0) GO TO 710
        RO=ORDIN(RO,A(2),XFUT)
        GO TO 720
  710   RO=0.
C       NEW PROFILE CALCULATION
  720   K=2
  730   DO 820 I1=1,2
        R=(-XSTEP/A(1))*TANB(K)
        GO TO (750,740), I1
  740   R=(-XSTEP/A(1))*TANA(K)
  750   IF (K.LE.INT(.8*FLOAT(I(6)))) GO TO 780
        IF (R) 770,770,760
  760   UINT=R*U(K+1)+(1.-R)*U(K)
        TINT=R*TAU(K+1)+(1.-R)*TAU(K)
        WINT=R*W(K+1)+(1.-R)*W(K)
        GO TO 790
  770   UINT=-R*U(K-1)+(1.+R)*U(K)
        TINT=-R*TAU(K-1)+(1.+R)*TAU(K)
        WINT=-R*W(K-1)+(1.+R)*W(K)
        GO TO 790
  780   CALL FINT (R,U,W,TAU,K,UINT,TINT,WINT)
  790   ZK=FLOAT(K)
        UL=.5*(UINT+U(K))
        TOR=.5*(TINT+TAU(K))
        IF (TOR.LT.0.0) GO TO 1120
        WW=.5*(WINT+W(K))
        RR=(ZK+.5*R)*A(1)/A(6)
        D=RM*GORD(RR)
        E=WW*TOR/A(8)
        THET=D*WW+1.
        SIGMA=D+(A(15)+.5)*E+SQRT((D+(A(15)+.5)*E)**2+THET*2.*TOR/A(8))*(-
       11.)**I1
        XYZ=4.5
        IF (RO.GT.0.) XYZ=7.
        RO=RO
        DD=RLORD(RR)
        DDF=DD
        RICH=2.*RO*DD*UL/SQRT(TOR)*A(6)*(1.+.5*UL*WW)
        DD=DD/(1.+XYZ*RICH)
        IF (DD.LT.2.*DDF) GO TO 800
  800   IF (DD.GT.0.) GO TO 810
        DD=2.*DDF
  810   CONTINUE
        AL(I1,1)=TOR*THET
        AL(I1,2)=-.5*SIGMA
  820   AL(I1,3)=-TINT*AL(I1,2)-TOR*(UINT*THET+XSTEP*(-T*THET/WW+A(8)/UL/A
       1(6)*(SQRT(TOR)/DD+RM*SLOG(RR))*SIGMA))
        CALL SOLVE (UFUT(K),TFUT(K),DET,AL)
        IF (DET.EQ.0.) GO TO 870
        IF ((TFUT(K)/RK2).LT.1.0E-10) TFUT(K)=RK2*1.E-10
```

```
      IF (UFUT(K).GT.RK) UFUT(K)=RK
      IF (K.LE.(I6-5)) GO TO 830
      IF (TFUT(K).GT.TFUT(K-1)) TFUT(K)=RK2*1.0E-10
      IF (UFUT(K).LT.UFUT(K-1)) UFUT(K)=RK
  830 CONTINUE
      WFUT(K)=UFUT(K)/(RK4/AY-.5*UFUT(K)**2)
      IF (1.-UFUT(K)/RK-1.0E-3) 840,850,850
  840 IF (TFUT(K)/RK**2-1.0E-6) 880,850,850
  850 IF (K-I(6)) 860,860,880
  860 K=K+1
      GO TO 730
  870 I(6)=K-1
      GO TO 890
  880 I(6)=K
  890 I6=I(6)
      DIV=0.
      IF (I(9)) 930,930,900
  900 DIV=1./(X+XSTEP-ORDIN(Z,A(2),XFUT))
      IF (I(9)-1) 930,930,910
  910 DIV=SLOPE(Z,A(2),I(5),XFUT)/ORDIN(Z,A(2),XFUT)
      IF (I(9)-2) 930,930,920
  920 DIV=ORDIN(Z,A(2),XFUT)
  930 I61=I(6)+1
      I62=I(6)+2
      I64=I(6)+4
      DO 940 J=I61,I62
      UFUT(J)=RK
      WFUT(J)=RK3
  940 TFUT(J)=RK2*1.0E-10
C     NEW V
      DO 950 J=1,I61
      Y=1.+.25*(UFUT(J+1)+UFUT(J))*(WFUT(J+1)+WFUT(J))
  950 V(J+1)=(UFUT(J+1)*V(J)-Y*(TFUT(J+1)-TFUT(J))-.5*(WFUT(J+1)+WFUT(J)
     1)*(TFUT(J+1)+TFUT(J))*Y*(UFUT(J+1)-UFUT(J))*.5-.25*(UFUT(J)+UFUT(J
     2+1))**2*A(1)*(VDLPDX*(1.-Y*AY/A(12)/(Y-1.))+DIV))/UFUT(J)
      DO 960 J=1,I62
      CALL TANCAL (G,A,V,J,RM,D,UFUT,TFUT,TANAFU,TANBFU,WFUT)
  960 CONTINUE
C     RECALCULATION NEAR SURFACE USING IMPROVED INTERPOLATION
      DO 1000 J=2,I8
      ZJ=FLOAT(J)
      DO 990 I1=1,2
      R=(-XSTEP/A(1))*0.5*(TANBFU(J)+TANB(J+1))
      GO TO (980,970), I1
  970 R=(-XSTEP/A(1))*0.5*(TANAFU(J)+TANA(J-1))
  980 CALL FINT (R,U,W,TAU,J,UINT,TINT,WINT)
      UL=0.5*(UINT+UFUT(J))
      TOR=0.5*(TINT+TFUT(J))
      WW=.5*(WINT+WFUT(J))
      RR=(ZJ+.5*R)*A(1)/A(6)
      D=GORD(RR)*RM
      E=WW*TOR/A(8)
      THET=1.+D*WW
      SIGMA=D+(A(15)+.5)*E+SQRT((D+(A(15)+.5)*E)**2+THET*2.*TOR/A(8))*(-
     11.)**I1
      XYZ=4.5
      IF (RO.GT.0.) XYZ=7.
      DD=RLORD(RR)
      DD=DD/(1.+2.*XYZ*RO*DD*UL/SQRT(TOR)*A(6)*(1.+.5*UL*WW))
      AL(I1,1)=TOR*THET
      AL(I1,2)=-.5*SIGMA
  990 AL(I1,3)=-TINT*AL(I1,2)-TOR*(UINT*THET+XSTEP*(-T*THET/WW+A(8)/UL/A
     1(6)*(SQRT(TOR)/DD+RM*SLOG(RR))*SIGMA))
      CALL SOLVE (UFUT(J),TFUT(J),DET,AL)
      WFUT(J)=UFUT(J)/(RK4/AY-.5*UFUT(J)**2)
      CALL TANCAL (G,A,V,J,RM,D,UFUT,TFUT,TANAFU,TANBFU,WFUT)
      IF (TANAFU(J).GT.10.**70.) GO TO 1110
```

```
 1000 CONTINUE
      DO 1010 J=1,I61
      Y=1.+.25*(UFUT(J+1)+UFUT(J))*(WFUT(J+1)+WFUT(J))
 1010 V(J+1)=(UFUT(J+1)*V(J)-Y*(TFUT(J+1)-TFUT(J))-.5*(WFUT(J+1)+WFUT(J)
     1)*(TFUT(J+1)+TFUT(J))*Y*(UFUT(J+1)-UFUT(J))*.5-.25*(UFUT(J)+UFUT(J
     2+1))**2*A(1)*(VDLPDX*(1.-Y*AY/A(12)/(Y-1.))+DIV))/UFUT(J)
      A(5)=TAU0/RK2/(1.+RK5)
C     DELTA 995 FOR SCALING L AND G
      I625=I(6)/4
      DO 1030 J=I625,I6
      IF (UFUT(J)/RK-.995) 1020,1030,1030
 1020 L=J
      RL=FLOAT(L)
 1030 CONTINUE
      A(6)=A(1)*((.995*RK-UFUT(L))/(UFUT(L+1)-UFUT(L))+RL)
C     REDUCTION IN NUMBER OF PROFILE POINTS IF REQUIRED
      IF (I(6)-LIMIT) 1090,1090,1040
 1040 IF (CFR-A(17)) 1090,1090,1050
 1050 I(6)=INT(FLOAT(I(6))*2./3.)
      I61=I(6)+1
      DO 1080 J=1,I61
      IS=3*J/2
      RIS=FLOAT(IS)
      IF (J-2*(J/2)) 1060,1070,1060
 1060 UFUT(J)=(UFUT(IS+1)*ALOG(1.+.5/RIS)-UFUT(IS)*ALOG(1.-.5/(RIS+1.)))
     1/ALOG(1.+1./RIS)
      TFUT(J)=.5*(TFUT(IS)+TFUT(IS+1))
      WFUT(J)=.5*(WFUT(IS)+WFUT(IS+1))
      TANAFU(J)=.5*(TANAFU(IS)+TANAFU(IS+1))
      TANBFU(J)=.5*(TANBFU(IS)+TANBFU(IS+1))
      V(J)=.5*(V(IS)+V(IS+1))
      GO TO 1080
 1070 UFUT(J)=UFUT(IS)
      WFUT(J)=WFUT(IS)
      TFUT(J)=TFUT(IS)
      TANAFU(J)=TANAFU(IS)
      TANBFU(J)=TANBFU(IS)
      V(J)=V(IS)
 1080 CONTINUE
      A(1)=A(1)*1.5
 1090 X=X+XSTEP
      NCOUNT=NCOUNT+1
      GO TO 80
 1100 KTYPE=3
      GO TO 1150
 1110 KTYPE=4
      GO TO 1150
 1120 KTYPE=5
      GO TO 1150
 1130 KTYPE=2
      KTSTSEP=1
      GO TO 1150
 1140 KTYPE=1
 1150 NORUN=NORUN+1
      IF (XBL(KKK).GT.0.15) GO TO 1190
C     DEFINE DISPLACEMENT THICKNESS TO BE ZERO EVERYWHERE ON GIVEN
C     SURFACE IF SEPARATION OCCURS NEAR THE LEADING EDGE
      IF (KSURF.EQ.2) GO TO 1170
      DO 1160 J=NLI,NT
 1160 DELBLX(J)=0.0
      SUPOUT(IGRID+1,KCYCLE)=XBL(KKK)
      GO TO 1310
 1170 DO 1180 J=1,NL
 1180 DELBLX(J)=0.0
      SLOWOUT(IGRID+1,KCYCLE)=XBL(KKK)
      GO TO 1310
 1190 IF (KSURF.EQ.2) GO TO 1230
C     INTERPOLATE FOR DELSTAR AT UPPER SURFACE STATIONS WHERE BODY
```

```
C      COORDINATES ARE KNOWN
       DO 1220 J=NL1,NT
       IF (XBODY(J).GT.XBL(KKK)) GO TO 1200
       CALL DISCOT (XBODY(J),XBODY(J),XBL,DELBL,DELBL,-040,KKK,0,DELBLXA)
       GO TO 1210
C      EXTRAPOLATION AFT OF LAST CALCULATION POINT IS ONLY TEMPORARY
C      AND IS CHANGED WHEN DEFINING THE EQUIVALENT INVISCID AIRFOIL
 1200  CALL DISCOT (XBODY(J),XBODY(J),XBL,DELBL,DELBL,-010,KKK,0,DELBLXA)
 1210  IF (DELBLXA.GT.0.025) DELBLXA=0.025
       IF (DELBLXA.LT.0.000) DELBLXA=0.000
 1220  DELBLX(J)=DELBLXA
       IF (KTSTSEP.EQ.1) SUPOUT(IGRID+1,KCYCLE)=XBL(KKK)
       XLAST(1)=XBL(KKK)-.005
       XLAST(3)=XLAST(1)-0.01
       CALL DISCOT (XLAST(1),XLAST(1),XBL,DELBL,DELBL,-040,KKK,0,DELLAST(
      11))
       CALL DISCOT (XLAST(3),XLAST(3),XBL,DELBL,DELBL,-040,KKK,0,DELLAST(
      13))
       GO TO 1310
C      INTERPOLATE FOR DELSTAR AT LOWER SURFACE STATIONS WHERE BODY
C      COORDINATES ARE KNOWN
 1230  IF (KTSTSEP.EQ.0) GO TO 1240
C      EXTRAPOLATION AFT OF LAST CALCULATION POINT IS LINEAR FOR NO
C      SEPARATION AND IS A FUNCTION OF THE CP DISTRIBUTION OTHERWISE
C      (LOWGRD IS DEFINED IN GETCP)
       SLOWOUT(IGRID+1,KCYCLE)=XBL(KKK)
       IF (LOWGRD.LT.0) GO TO 1280
 1240  DO 1270 J=1,NL
       IF (XBODY(J).GT.XBL(KKK)) GO TO 1250
       CALL DISCOT (XBODY(J),XBODY(J),XBL,DELBL,DELBL,-040,KKK,0,DELBLXA)
       GO TO 1260
 1250  CALL DISCOT (XBODY(J),XBODY(J),XBL,DELBL,DELBL,-010,KKK,0,DELBLXA)
 1260  IF (DELBLXA.GT.0.025) DELBLXA=0.025
       IF (DELBLXA.LT.0.000) DELBLXA=0.000
 1270  DELBLX(J)=DELBLXA
       IF (KTSTSEP.NE.1) GO TO 1310
       XLAST(2)=XBL(KKK)-.005
       XLAST(4)=XLAST(2)-0.01
       CALL DISCOT (XLAST(2),XLAST(2),XBL,DELBL,DELBL,-040,KKK,0,DELLAST(
      12))
       CALL DISCOT (XLAST(4),XLAST(4),XBL,DELBL,DELBL,-040,KKK,0,DELLAST(
      14))
       GO TO 1310
 1280  XSEP1=XBL(KKK)-0.10
       XSEP2=XBL(KKK)-0.08
       XSEP3=XBL(KKK)/2.+0.50
       XSEP4=1.00
       CP1=2.0*(ORDIN(P,A(2),XSEP1)/TEMP-1.0)/A(12)/(EM**2.)
       CP3=2.0*(ORDIN(P,A(2),XSEP3)/TEMP-1.0)/A(12)/(EM**2.)
       CALL DISCOT (XSEP1,XSEP1,XBL,DELBL,DELBL,-040,KKK,0,YSEP1)
       CALL DISCOT (XSEP2,XSEP2,XBL,DELBL,DELBL,-040,KKK,0,YSEP2)
       YSEP3=YSEP1+0.033*(CP3-CP1)-0.022*(XSEP3-XSEP1)
       YSEP4=(YSEP1+YSEP3)/2.0
       DELX2=XSEP2-XSEP1
       DELX3=XSEP3-XSEP1
       DELX4=XSEP4-XSEP1
       DELY2=YSEP2-YSEP1
       DELY3=YSEP3-YSEP1
       DELY4=YSEP4-YSEP1
       APOLY=(DELY2*DELX3*DELX4*(XSEP4-XSEP3)-DELY3*DELX2*DELX4*(XSEP4-XS
      1EP2)+DELY4*DELX2*DELX3*(XSEP3-XSEP2))/((XSEP2**3-XSEP1**3)*DELX3*D
      2ELX4*(XSEP4-XSEP3)-(XSEP3**3-XSEP1**3)*DELX2*DELX4*(XSEP4-XSEP2)+(
      3XSEP4**3-XSEP1**3)*DELX2*DELX3*(XSEP3-XSEP2))
       BPOLY=(DELY3*DELX2-DELY2*DELX3-APOLY*((XSEP3**3-XSEP1**3)*DELX2-(X
      1SEP2**3-XSEP1**3)*DELX3))/DELX2/DELX3/(XSEP3-XSEP2)
       CPOLY=(DELY2-BPOLY*DELX2*(XSEP1+XSEP2)-APOLY*(XSEP2**3-XSEP1**3))/
      1DELX2
       DPOLY=YSEP1-CPOLY*XSEP1-BPOLY*XSEP1**2-APOLY*XSEP1**3
       DO 1300 J=1,NL
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF (XBODY(J).GT.XSEP2) GO TO 1290
      CALL DISCOT (XBODY(J),XBODY(J),XBL,DELBL,DELBL,-040,KKK,0,DELBLXA)
      GO TO 1300
 1290 DELBLXA=APOLY*XBODY(J)**3+BPOLY*XBODY(J)**2+CPOLY*XBODY(J)+DPOLY
 1300 DELBLX(J)=DELBLXA
 1310 CONTINUE
      KSURF=2
      DO 1320 J=1,13
 1320 I(J)=ITEMP(J)
      DO 1330 J=1,19
 1330 A(J)=ATEMP(J)
      WRITE (N3) COMF
C     EITHER REPEAT B. L. CALCULATIONS FOR OTHER SURFACE OR DEFINE
C     EQUIVALENT INVISCID AIRFOIL IF BOTH SURFACES ARE COMPLETE
      IF (NORUN-NUMBER) 70,1340,1340
 1340 DO 1350 J=1,NT
      ANG=BODSLOP(J)*3.14159/180.0
      XNEW(J)=XBODY(J)+DELBLX(J)*SIN(ANG)
 1350 YNEW(J)=YBODY(J)-DELBLX(J)*COS(ANG)
      IF (XLAST(1).GT.1.0) GO TO 1410
C     DEFINE DELSTAR AFT OF LAST CALCULATION POINT ON UPPER SURFACE TO
C     MAINTAIN SLOPE OF EQUIV INV AIRF CONSTANT FROM LAST CALC POINT TO
C     THE TRAILING EDGE
      JJJ=NL1
 1360 IF (XBODY(JJJ).GT.XLAST(1)) GO TO 1370
      JJJ=JJJ+1
      GO TO 1360
 1370 ISURF=1
      ISTRT1=JJJ-5
      IFNSH1=JJJ
      ISTRT2=JJJ
      IFNSH2=NT
      IINDX=6-JJJ
 1380 DO 1390 J=ISTRT1,IFNSH1
      JM=IINDX-J*(-1)**ISURF
      XXX(JM)=XBODY(J)
      YYY(JM)=YBODY(J)
 1390 BBB(JM)=BODSLOP(J)
      XLST=XLAST(ISURF)
      XLSTM=XLAST(ISURF+2)
      CALL DISCOT (XLST,XLST,XXX,YYY,YYY,-030,6,0,YLST)
      CALL DISCOT (XLSTM,XLSTM,XXX,YYY,YYY,-030,6,0,YLSTM)
      CALL DISCOT (XLST,XLST,XXX,BBB,BBB,-030,6,0,BLST)
      CALL DISCOT (XLSTM,XLSTM,XXX,BBB,BBB,-030,6,0,BLSTM)
      ANG=BLST*3.14159/180.0
      XLST=XLST+DELLAST(ISURF)*SIN(ANG)
      YLST=YLST-DELLAST(ISURF)*COS(ANG)
      ANG=BLSTM*3.14159/180.0
      XLSTM=XLSTM+DELLAST(ISURF+2)*SIN(ANG)
      YLSTM=YLSTM-DELLAST(ISURF+2)*COS(ANG)
      SEPSLOP=(YLST-YLSTM)/(XLST-XLSTM)
      DO 1400 J=ISTRT2,IFNSH2
      ANG=BODSLOP(J)*3.14159/180.0
      DELBLXA=(YBODY(J)-YLST+SEPSLOP*(XLST-XBODY(J)))/(SEPSLOP*SIN(ANG)+
     1COS(ANG))
      IF (DELBLXA.GT.0.025) DELBLXA=0.025
      IF (DELBLXA.LT.0.000) DELBLXA=0.000
      DELBLX(J)=DELBLXA
      XNEW(J)=XBODY(J)+DELBLX(J)*SIN(ANG)
 1400 YNEW(J)=YBODY(J)-DELBLX(J)*COS(ANG)
      IF (ISURF.GT.1) GO TO 1440
 1410 IF (XLAST(2).GT.1.0) GO TO 1440
C     SIMILARLY DEFINE DELSTAR ON LOWER SURFACE IF PRES DIST WARRANTS IT
      JJJ=1
 1420 IF (XBODY(JJJ).LT.XLAST(2)) GO TO 1430
      JJJ=JJJ+1
      GO TO 1420
```

```
 1430 ISURF=2
      IF (JJJ.EQ.1) JJJ=2
      ISTRT1=JJJ-1
      IFNSH1=JJJ+4
      ISTRT2=1
      IFNSH2=JJJ-1
      IINDX=JJJ+5
      GO TO 1380
C     REDEFINE THE TRAILING EDGE COORDINATES OF EQUIV INV AIRF SO THAT
C     THE LAST X/C IS 1.0 ON BOTH SURFACES
 1440 NTM5=NT-5
      DO 1450 J=NTM5,NT
      JM=J-NT+6
      XXX(JM)=XNEW(J)
 1450 YYY(JM)=YNEW(J)
      CALL DISCOT (XTRAIL,XTRAIL,XXX,YYY,YYY,-030,6,0,YTRAIL)
      XNEW(NT)=XTRAIL
      YNEW(NT)=YTRAIL
      NLM5=NL-5
      DO 1460 J=NLM5,NL
      JM=J-NL+6
      XXX(JM)=XNEW(J)
 1460 YYY(JM)=YNEW(J)
      CALL DISCOT (XTRAIL,XTRAIL,XXX,YYY,YYY,-030,6,0,YTRAIL)
      XNEW(NL)=XTRAIL
      YNEW(NL)=YTRAIL
      RETURN
      END
      SUBROUTINE BOTH
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TP1,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      DIMENSION XY(2)
      EQUIVALENCE (XY(1),Z(1))
      PF=1./X
      DELR=X*DELR
      DELTH=X*DELTH
      DR=PF*DR
      DT=PF*DT
      RA3=PF*RA3
      RA4=PF*PF*RA4
      RA5=PF*RA5
      NCY=0
      MP=MM+1
      CALL PERMUT (R,NN,1)
      CALL PERMUT (RS,NN,1)
      CALL PERMUT (RI,NN,1)
      CALL PERMUT (CO,MP,1)
      CALL PERMUT (SI,MP,1)
      CALL PERMUT (PHIR,MP,1)
      CALL PERMUT (Z,MP,2)
      CALL PERMUT (XY(2),MP,2)
      CALL PERMUT (FM,MP,1)
      DO 10 L=1,NN
      CALL PERMUT (FP(1,L),MP,1)
   10 CALL PERMUT (PHI(1,L),MP,1)
      DO 20 L=1,MP
```

```
      CALL PERMUT (FP(L,1),NN,IDIM)
   20 CALL PERMUT (PHI(L,1),NN,IDIM)
      MM=M+1
      MP=MM+1
      LL=MP/2
      LP=LL+2
      IF (X.EQ..5) GO TO 70
      DO 30 L=1,M,2
      DO 30 J=1,NN,2
   30 PHI(L+1,J)=.5*(PHI(L,J)+PHI(L+2,J))
      DO 40 J=1,N,2
      DO 40 L=1,MM
   40 PHI(L,J+1)=.5*(PHI(L,J)+PHI(L,J+2))
      DO 50 K=1,NN
      FP(2,K)=FP(MP,K)
   50 PHI(MP,K)=PHI(2,K)+DPHI
      DO 60 L=1,MP
      BQ=FLOAT(L-1)*DT
      CO(L)=COS(BQ)
   60 SI(L)=SIN(BQ)
      CALL COSI
      RETURN
   70 NN=N+1
      DO 80 K=1,NN
      FP(MP,K)=FP(2,K)
   80 PHI(MP,K)=PHI(2,K)+DPHI
      CO(MP)=CO(2)
      SI(MP)=SI(2)
      PHIR(MP)=PHIR(2)+TP
      RETURN
      END
      SUBROUTINE CONVER (ICHECK,KCYCLE,M,CL,Z,CPSURF,NCY)
C     CHECK FOR CONVERGENCE OF INVISCID-FLOW/BOUNDARY-LAYER ITERATION
      COMPLEX Z
      DIMENSION CPSURF(162), CPOLD(162), Z(162), X(162)
      MPLUS=M+1
C     IF KCYCLE EQ 1 JUST DEFINE OLD VALUES FOR COMPARISON ON NEXT CYCLE
      IF (KCYCLE.EQ.1) GO TO 30
      ICHECK=1
C     NCY LE 20 INDICATES CONVERGENCE
      IF (NCY.LE.20) GO TO 50
C     THE CL AND EVERY CP REPEATING WITHIN A SPECIFIED TOLERANCE
C     INDICATES CONVERGENCE
      DELTACL=ABS(CL-CLOLD)
      IF (DELTACL.GT.0.02) GO TO 30
      DO 10 L=1,MPLUS
   10 X(L)=REAL(Z(L))
      DO 20 L=2,M
      SLOPE1=((CPSURF(L)-CPSURF(L-1))/(X(L)-X(L-1)))**2
      SLOPE2=((CPSURF(L)-CPSURF(L+1))/(X(L)-X(L+1)))**2
      SLOPE3=((CPOLD(L)-CPOLD(L-1))/(X(L)-X(L-1)))**2
      SLOPE4=((CPOLD(L)-CPOLD(L+1))/(X(L)-X(L+1)))**2
      SLOPEM=AMAX1(SLOPE1,SLOPE2,SLOPE3,SLOPE4)
      TEST=0.0005*SLOPEM+0.025
      DELTACP=ABS(CPSURF(L)-CPOLD(L))
      IF (DELTACP.GT.TEST) GO TO 30
   20 CONTINUE
      SLOPE1=ABS((CPSURF(1)-CPSURF(2))/(X(1)-X(2)))
      SLOPE2=ABS((CPSURF(M)-CPSURF(MPLUS))/(X(M)-X(MPLUS)))
      SLOPE3=ABS((CPOLD(1)-CPOLD(2))/(X(1)-X(2)))
      SLOPE4=ABS((CPOLD(M)-CPOLD(MPLUS))/(X(M)-X(MPLUS)))
      TESLOP1=((SLOPE1+SLOPE2)/2.0)**2
      TESLOP2=((SLOPE3+SLOPE4)/2.0)**2
      SLOPEM=AMAX1(TESLOP1,TESLOP2)
      TEST=0.0005*SLOPEM+0.025
      DELTACP=ABS(CPSURF(1)-CPOLD(1))
      IF (DELTACP.LE.TEST) GO TO 50
   30 ICHECK=0
      DO 40 L=1,MPLUS
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
   40 CPOLD(L)=CPSURF(L)
      CLOLD=CL
   50 RETURN
      END
      SUBROUTINE COSI
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      DO 10 L=1,M
      X=CO(L)
      CO(L)=X*CN-SI(L)*SN
   10 SI(L)=SI(L)*CN+X*SN
      CALL INIT
      RETURN
      END
      SUBROUTINE CPLOT (X,N)
      COMMON /D/ SF,SIZE,ANG,XMAX,YMAX,XOR,YOR,PGSIZ
      DIMENSION X(2)
C     CHANGE RELATIVE MOVEMENTS TO ABSOLUTE INCHES
      XX=XOR+SF*X(1)
      YY=YOR+SF*X(2)
C     CHECK TO SEE IF WE ARE WITHIN THE PAGE
      IF ((XX.LT.0.).OR.(YY.LT.0.).OR.(XX.GT.XMAX).OR.(YY.GT.YMAX)) GO T
     1O 20
   10 CALL PLOT (XX,YY,IABS(N))
      IF (N.GT.0) RETURN
      XOR=XX
      YOR=YY
      RETURN
   20 IF (N.LT.0) GO TO 30
      XX=AMAX1(0.,AMIN1(XX,XMAX))
      YY=AMAX1(0.,AMIN1(YY,YMAX))
      GO TO 10
C     GO TO NEXT PAGE
   30 XOR=0.
      YOR=-3.0
      CALL PLOT (PGSIZ,0.,N)
      RETURN
      END
      SUBROUTINE CRUDER
C     DOUBLES THE MESH SIZE
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      X=.5
      M=M/2
      N=N/2
      II=II/2+1
      JJ=JJ/2+1
      CALL BOTH
      RETURN
      END
      SUBROUTINE CSYMBL (X,N,L)
      COMMON /D/ SF,SIZE,ANG,XMAX,YMAX,XOR,YOR,PGSIZ
      DIMENSION X(2)
```

```
C     CHANGE RELATIVE MOVEMENTS TO ABSOLUTE INCHES
      XX=XOR+SF*X(1)
      YY=YOR+SF*X(2)
C     CHECK TO SEE IF WE ARE WITHIN THE PAGE
      IF ((XX.LT.0.).OR.(YY.LT.0.).OR.(XX.GT.XMAX).OR.(YY.GT.YMAX)) RETU
     1RN
      CALL SYMBOL (XX,YY,SIZE,N,ANG,L)
      RETURN
      END
      SUBROUTINE DISCOT (XA,ZA,TABX,TABY,TABZ,NC,NY,NZ,ANS)
C     INTERPOLATION AND EXTRAPOLATION ROUTINE
      DIMENSION TABX(2), TABY(2), TABZ(2), NPX(8), NPY(8), YY(8)
      CALL UNS (NC,IA,IDX,IDZ,IMS)
      IF (NZ-1) 10,10,20
   10 CALL DISSER (XA,TABX(1),1,NY,IDX,NN)
      NNN=IDX+1
      CALL LAGRAN (XA,TABX(NN),TABY(NN),NNN,ANS)
      GO TO 120
   20 ZARG=ZA
      IP1X=IDX+1
      IP1Z=IDZ+1
      IF (IA) 30,50,30
   30 IF (ZARG-TABZ(NZ)) 50,50,40
   40 ZARG=TABZ(NZ)
   50 CALL DISSER (ZARG,TABZ(1),1,NZ,IDZ,NPZ)
      NX=NY/NZ
      NPZL=NPZ+IDZ
      I=1
      IF (IMS) 60,60,80
   60 CALL DISSER (XA,TABX(1),1,NX,IDX,NPX(1))
      DO 70 JJ=NPZ,NPZL
      NPY(I)=(JJ-1)*NX+NPX(1)
      NPX(I)=NPX(1)
   70 I=I+1
      GO TO 100
   80 DO 90 JJ=NPZ,NPZL
      IS=(JJ-1)*NX+1
      CALL DISSER (XA,TABX(1),IS,NX,IDX,NPX(I))
      NPY(I)=NPX(I)
   90 I=I+1
  100 DO 110 LL=1,IP1Z
      NLOC=NPX(LL)
      NLOCY=NPY(LL)
  110 CALL LAGRAN (XA,TABX(NLOC),TABY(NLOCY),IP1X,YY(LL))
      CALL LAGRAN (ZARG,TABZ(NPZ),YY(1),IP1Z,ANS)
  120 RETURN
      END
      SUBROUTINE DISSER (XA,TAB,I,NX,ID,NPX)
      DIMENSION TAB(2)
      NPT=ID+1
      NPB=NPT/2
      NPU=NPT-NPB
      IF (NX-NPT) 20,10,20
   10 NPX=I
      RETURN
   20 NLOW=I+NPB
      NUPP=I+NX-(NPU+I)
      DO 30 II=NLOW,NUPP
      NLOC=II
      IF (TAB(II)-XA) 30,40,40
   30 CONTINUE
      NPX=NUPP-NPB+1
      RETURN
   40 NL=NLOC-NPB
      NU=NL+ID
      DO 50 JJ=NL,NU
      NDIS=JJ
      IF (TAB(JJ)-TAB(JJ+1)) 50,60,50
```

```
   50 CONTINUE
      NPX=NL
      RETURN
   60 IF (TAB(NDIS)-XA) 80,70,70
   70 NPX=NDIS-ID
      RETURN
   80 NPX=NDIS+1
      RETURN
      END
      SUBROUTINE FINT (R,U,W,T,J,A,B,C)
      DIMENSION U(110), T(110), W(110)
      C1=1.-R**2
      C2=0.5*(R**2+R)
      C3=0.5*(R**2-R)
      A=C1*U(J)+C2*U(J+1)+C3*U(J-1)
      B=C1*T(J)+C2*T(J+1)+C3*T(J-1)
      C=C1*W(J)+C2*W(J+1)+C3*W(J-1)
      RETURN
      END
      FUNCTION FN (Z)
      A=SQRT(1.+Z)
      FN=2.*(ALOG(2./(1.+A))+A-1.)
      RETURN
      END
      SUBROUTINE FORCES (CDF)
C     COMPUTE LIFT, DRAG, AND PITCHING MOMENT COEFFICIENTS BY
C     INTEGRATING THE PRESSURE DISTRIBUTION
      COMMON /E/ KCYCLE,FNU,FNL,IBNDLAY(13),ABNDLAY(19),XBODY(162),YBODY
     1(162),BODSLOP(162),BODCURV(82),XNEW(162),YNEW(162),CPSURF(162),CPB
     2DLY(82),IGRID,GRID,XUPLE(20),XUPARC(20),XLOLE(20),XLOARC(20),TITLO
     3UT(15),CLOUT(3,7),SUPOUT(3,7),SLOWOUT(3,7),CMOUT,CPUP(85),CPLO(85)
     4,XTEMUP(85),XTEMLO(85),DELBLX(162),KOUNT,KOUNTUP,LOWGRD,INVDIV
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,CCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BO,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN,NCASE
      DIMENSION XTMB(101), YTMB(101), XTMN(101), XBN(162), YBN(162), CPB
     1N(162)
      NU=FNU
      NUM1=NU-1
      NL=FNL
      NLM1=NL-1
C     TRANSFER PRESSURE DISTRIBUTION FROM SURFACE OF EQUIVALENT
C     INVISCID AIRFOIL TO ACTUAL AIRFOIL FOR COMPUTATION OF FORCES
      DO 10 K=1,KOUNT
      IF (XTEMLO(K).LE.XNEW(NLM1)) GO TO 10
      KSTOPL=K-1
      GO TO 20
   10 CONTINUE
   20 DO 30 K=1,NLM1
      XTMB(K)=XBODY(K)
      YTMB(K)=YBODY(K)
   30 XTMN(K)=XNEW(K)
      DO 40 K=1,KSTOPL
      KMNS=KSTOPL+2-K
      CALL DISCOT (XTEMLO(K),XTEMLO(K),XTMN,XTMB,XTMB,-030,NLM1,0,XBN(KM
     1NS))
      CALL DISCOT (XBN(KMNS),XBN(KMNS),XTMB,YTMB,YTMB,-030,NLM1,0,YBN(KM
     1NS))
   40 CPBN(KMNS)=CPLO(K)
      XBN(1)=XBODY(NL)
      YBN(1)=YBODY(NL)
      CPBN(1)=CPLO(KOUNT)
      DO 50 K=1,NUM1
      KPLS=K+NL
      XTMB(K)=XBODY(KPLS)
      YTMB(K)=YBODY(KPLS)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
   50 XTMN(K)=XNEW(KPLS)
      DO 70 K=1,KOUNTUP
      KPLS=K+KSTOPL+1
      IF (XTEMUP(K).LE.XTMN(NUM1)) GO TO 60
      KSTOPU=K-1
      GO TO 80
   60 CALL DISCOT (XTEMUP(K),XTEMUP(K),XTMN,XTMB,XTMB,-030,NUM1,0,XBN(KP
     1LS))
      CALL DISCOT (XBN(KPLS),XBN(KPLS),XTMB,YTMB,YTMB,-030,NUM1,0,YBN(KP
     1LS))
   70 CPBN(KPLS)=CPUP(K)
   80 KTOT=KSTOPU+KSTOPL+1
      XBN(KTOT+1)=XBODY(NU+NL)
      YBN(KTOT+1)=YBODY(NU+NL)
      CPBN(KTOT+1)=CPUP(KOUNTUP)
      CLP=0.0
      CDP=0.0
      CMP=0.0
C     INTEGRATE PRESSURE DISTRIBUTION
      DO 90 K=1,KTOT
      DX=XBN(K+1)-XBN(K)
      DY=YBN(K+1)-YBN(K)
      XAVE=0.5*(XBN(K+1)+XBN(K))
      YAVE=0.5*(YBN(K+1)+YBN(K))
      CPA=0.5*(CPBN(K+1)+CPBN(K))
      DCL=-CPA*DX
      DCD=CPA*DY
      DCM=DCD*YAVE-DCL*(XAVE-.25)
      CLP=CLP+DCL
      CDP=CDP+DCD
   90 CMP=CMP+DCM
C     CORRECT FOR BASE PRESSURE AND FRICTION DRAG
      CDT=CDP+CDF+CPBN(1)*(YBN(1)-YBN(KTOT+1))
      ALPHA=ALP/RAD
C     ADJUST COEFFICIENTS FOR ANGLE OF ATTACK
      CDC=CDT*COS(ALPHA)+CLP*SIN(ALPHA)
      CLC=CLP*COS(ALPHA)-CDT*SIN(ALPHA)
      CLOUT(IGRID+1,KCYCLE)=CLC
      CMOUT=CMP
      X=CDC
      RETURN
      END
      SUBROUTINE GETCP
C     COMPUTE PRESSURE COEFFICIENTS
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,QA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMMON /E/ KCYCLE,FNU,FNL,IBNDLAY(13),ABNDLAY(19),XBODY(162),YBODY
     1(162),BODSLOP(162),BODCURV(82),XNEW(162),YNEW(162),CPSURF(162),CPB
     2DLY(82),IGRID,GRID,XUPLE(20),XUPARC(20),XLOLE(20),XLOARC(20),TITLO
     3UT(15),CLOUT(3,7),SUPOUT(3,7),SLOWOUT(3,7),CMOUT,CPUP(85),CPLO(85)
     4,XTEMUP(85),XTEMLO(85),DELBLX(162),KOUNT,KOUNTUP,LOWGRD,INVDIV
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      INTEGER A
      COMPLEX CLCD,TMP,CI,ZER,CEXP
      DIMENSION QS(1), XFIT(3), YFIT(3)
      EQUIVALENCE (QS(1),PHIR(1))
      CPR(Q)=C5*(AMAX1(0.,C4-C6*Q)**C7-1.)
      IF (INVDIV.EQ.0) GO TO 20
      DO 10 L=1,MM
   10 CPSURF(L)=0.0
      CLOUT(IGRID+1,KCYCLE)=999.0
      RETURN
```

```
   20 LLL=MM-M/40+1
      LLLM1=LLL-1
      LLLM2=LLL-2
      JJJ=M/40
      JJJP1=JJJ+1
      JJJP2=JJJ+2
      DO 30 L=JJJP1,LLLM1
      U=(PHI(L+1,1)-PHI(L-1,1))*DELTH-SI(L)
      QS(L)=(U*U)/FP(L,1)
   30 CPSURF(L)=CPR(QS(L))
C     DEFINE CP AT THE TRAILING EDGE BY CONSIDERING THE GRADIENTS IN
C     THE CP DISTRIBUTION NEAR THE T. E. RATHER THAN AVERAGING THE UPPER
C     AND LOWER CP AT THE POINTS JUST FORWARD OF THE T. E.
      TESLOPU=(CPSURF(LLLM1)-CPSURF(LLLM2))/(REAL(Z(LLLM1))-REAL(Z(LLLM2
     1)))
      TESLOPL=(CPSURF(JJJP1)-CPSURF(JJJP2))/(REAL(Z(JJJP1))-REAL(Z(JJJP2
     1)))
      IF (TESLOPL.GT.2.0) GO TO 60
      LOWGRD=-1
      DO 40 L=2,JJJ
      U=(PHI(L+1,1)-PHI(L-1,1))*DELTH-SI(L)
      QS(L)=(U*U)/FP(L,1)
   40 CPSURF(L)=CPR(QS(L))
      DO 50 L=LLL,MM
      CPSURF(L)=CPSURF(LLLM1)+TESLOPU*(REAL(Z(L))-REAL(Z(LLLM1)))
   50 QS(L)=(C4-(CPSURF(L)/C5+1.)**(1./C7))/C6
      CPSURF(1)=CPSURF(MM)
      QS(1)=QS(MM)
      GO TO 160
   60 LOWGRD=1
      CPTEU=CPSURF(LLLM1)+TESLOPU*(REAL(Z(MM))-REAL(Z(LLLM1)))
      CPTEL=CPSURF(JJJP1)+TESLOPL*(REAL(Z(1))-REAL(Z(JJJP1)))
      IF (CPTEU-CPTEL) 70,100,130
   70 DO 80 L=1,JJJ
      CPSURF(L)=CPSURF(JJJP1)+TESLOPL*(REAL(Z(L))-REAL(Z(JJJP1)))
   80 QS(L)=(C4-(CPSURF(L)/C5+1.)**(1./C7))/C6
      XFIT(1)=REAL(Z(LLLM2))
      XFIT(2)=REAL(Z(LLLM1))
      XFIT(3)=REAL(Z(1))
      YFIT(1)=CPSURF(LLLM2)
      YFIT(2)=CPSURF(LLLM1)
      YFIT(3)=CPSURF(1)
      DO 90 L=LLL,MM
      XLOOK=REAL(Z(L))
      CALL DISCOT (XLOOK,XLOOK,XFIT,YFIT,YFIT,-020,3,0,CPSURF(L))
   90 QS(L)=(C4-(CPSURF(L)/C5+1.)**(1./C7))/C6
      GO TO 160
  100 DO 110 L=1,JJJ
      CPSURF(L)=CPSURF(JJJP1)+TESLOPL*(REAL(Z(L))-REAL(Z(JJJP1)))
  110 QS(L)=(C4-(CPSURF(L)/C5+1.)**(1./C7))/C6
      DO 120 L=LLL,MM
      CPSURF(L)=CPSURF(LLLM1)+TESLOPU*(REAL(Z(L))-REAL(Z(LLLM1)))
  120 QS(L)=(C4-(CPSURF(L)/C5+1.)**(1./C7))/C6
      GO TO 160
  130 DO 140 L=LLL,MM
      CPSURF(L)=CPSURF(LLLM1)+TESLOPU*(REAL(Z(L))-REAL(Z(LLLM1)))
  140 QS(L)=(C4-(CPSURF(L)/C5+1.)**(1./C7))/C6
      XFIT(1)=REAL(Z(JJJP2))
      XFIT(2)=REAL(Z(JJJP1))
      XFIT(3)=REAL(Z(MM))
      YFIT(1)=CPSURF(JJJP2)
      YFIT(2)=CPSURF(JJJP1)
      YFIT(3)=CPSURF(MM)
      DO 150 L=1,JJJ
      XLOOK=REAL(Z(L))
      CALL DISCOT (XLOOK,XLOOK,XFIT,YFIT,YFIT,-020,3,0,CPSURF(L))
  150 QS(L)=(C4-(CPSURF(L)/C5+1.)**(1./C7))/C6
  160 CALL PHIRR
      RETURN
      END
```

```
      SUBROUTINE GOPLOT (N)
C     INITIATE PLOT
      CALL CALCOMP
      CALL LEROY
      RETURN
      END
      FUNCTION GORD (Z)
      IF (Z-.63) 10,20,20
   10 GORD=17.5*Z**1.86
      GO TO 50
   20 IF (Z-.89) 30,40,40
   30 GORD=90.9*Z-49.75
      GO TO 50
   40 GORD=18.7*Z+14.85
   50 RETURN
      END
      FUNCTION GRAD (FR,H,IA,IB,N)
      DIMENSION FR(110)
      IF (N-IA) 20,10,20
   10 G=(-1.*FR(IA+2)+4.*FR(IA+1)-3.*FR(IA))/(2.*H)
      GO TO 50
   20 IF (N-IB) 40,30,40
   30 G=(3.*FR(IB)-4.*FR(IB-1)+FR(IB-2))/(2.*H)
      GO TO 50
   40 G=(FR(N+1)-FR(N-1))/(2.*H)
   50 GRAD=G
      RETURN
      END
      SUBROUTINE GRAFIC
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMMON /E/ KCYCLE,FNU,FNL,IBNDLAY(13),ABNDLAY(19),XBODY(162),YBODY
     1(162),BODSLOP(162),BODCURV(82),XNEW(162),YNEW(162),CPSURF(162),CPB
     2DLY(82),IGRID,GRID,XUPLE(20),XUPARC(20),XLOLE(20),XLOARC(20),TITLO
     3UT(15),CLOUT(3,7),SUPOUT(3,7),SLOWOUT(3,7),CMOUT
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      COMMON /D/ SF,SIZE,ANG,XMAX,YMAX,XOR,YOR,PGSIZ
      DATA NPLOT,PF,EPF,SCF/0,-.5,7.0,5./
      PE(Q)=C5*(AMAX1(0.,C4-C6*Q)**C7-1.)
C     INITIATE PLOT OR GO TO NEXT PAGE
      IF (NPLOT.EQ.0) CALL GOPLOT (NRN)
      IF (NPLOT.GT.0) CALL CPLOT ((13.,-12.),-3)
      NPLOT=1
C     MOVE THE ORIGIN TO THE LOCATION X=0.,CP=0.
      CALL CPLOT (CMPLX(2.,EPF),-3)
C     PLOT CP CURVE AS A FUNCTION OF X
      CPF=1./PF
      CCP=CPF*CPSURF(1)
      CALL CPLOT (CMPLX(SCF*REAL(Z(1)),CCP),3)
      DO 10 L=2,MM
      TEMPCP=CPSURF(L)
      IF (TEMPCP.LT.-3.2) TEMPCP=-3.2
      CCP=CPF*TEMPCP
   10 CALL CPLOT (CMPLX(SCF*REAL(Z(L)),CCP),2)
C     DRAW AND LABEL CP-AXIS
      ANG=90.
      CALL XYAXES ((-.5,0.),3.,5.,PF)
      ANG=0.
C     COMPUTE AND PLOT CRITICAL SPEED
      YMX=CPF*PE(QCRIT)
      SIZE=.28
      CALL CSYMBL (CMPLX(-.5,YMX),15,-1)
```

```
      SIZE=.14
      CALL CSYMBL ((-1.2,1.5),1HC,1)
      CALL CSYMBL ((-1.05,1.4),1HP,1)
      YOR=1.5
C     LABEL THE PLOT
      SF=1.
      SIZE=.11
      CL1=CLOUT(IGRID+1,KCYCLE)
      ENCODE (60,20,A) NRN,CL1,CMOUT,X
      CALL CSYMBL ((0.0,-1.0),A,60)
      ENCODE (60,30,A) IBNDLAY(13),NCY,EP,M,N
      CALL CSYMBL ((0.0,-1.5),A,60)
      ENCODE (60,40,A) TITLOUT
      CALL CSYMBL ((-0.1,8.4),A,60)
      ENCODE (60,50,A) EM,ALP,ABNDLAY(18)
      CALL CSYMBL ((0.0,7.9),A,60)
      RETURN
C
C
   20 FORMAT (6HRUN = ,I3,4X,5HCL = ,F5.3,4X,5HCM = ,F5.3,4X,5HCD = ,F7.
     15)
   30 FORMAT (7HIFIX = ,I2,4X,6HNCY = ,I3,5X,5HEP = ,F3.2,6X,6HMXN = ,I3
     1,1HX,I2)
   40 FORMAT (15A4)
   50 FORMAT (7HMACH = ,F4.3,5X,8HALPHA = ,F5.2,5X,13HRN X 10E-6 = ,F6.2
     1)
      END
      SUBROUTINE INIT
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AO,BO,KP,YP,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      CO(MP)=CO(2)
      CO(MM)=CO(1)
      SI(MM)=SI(1)
      SI(MP)=SI(2)
      ALPO=ALP
      CN=COS(ALP+BB(1))
      SN=SIN(ALP+BB(1))
      CALL PHIRR
      RETURN
      END
      SUBROUTINE INTPL (NX,SI,FI,S,F,FP,FPP,FPPP)
C     GIVEN S,F(S) AND THE FIRST THREE DERIVATIVES AT A SET OF POINTS
C     FIND FI(SI) AT THE NX VALUES OF SI BY EVALUATING THE TAYLOR SERIES
C     OBTAINED BY USING THE FIRST THREE DERIVATIVES
      DIMENSION SI(1), FI(1), S(1), F(1), FP(1), FPP(1), FPPP(1)
      DATA PT/.33333333333333/
      J=0
      DO 30 I=1,NX
      VAL=0.
      SS=SI(I)
   10 J=J+1
      TT=S(J)-SS
      IF (FLOAT(J-1)*TT) 10,30,20
   20 J=J-1
      SS=SS-S(J)
      VAL=SS*(FP(J)+.5*SS*(FPP(J)+SS*PT*FPPP(J)))
   30 FI(I)=F(J)+VAL
      RETURN
      END
```

APPENDIX

```
      SUBROUTINE LAGRAN (XA,X,Y,N,ANS)
      DIMENSION X(2), Y(2)
      SUM=0.0
      DO 30 I=1,N
      PROD=Y(I)
      DO 20 J=1,N
      A=X(I)-X(J)
      IF (A) 10,20,10
   10 B=(XA-X(J))/A
      PROD=PROD*B
   20 CONTINUE
   30 SUM=SUM+PROD
      ANS=SUM
      RETURN
      END
      SUBROUTINE MAP
C     SUM UP FOURIER SERIES TO OBTAIN FIRST GUESS
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,FL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      DATA TOL/1.E-12/
      DO 50 J=1,N
      RN=R(J)
      DO 10 KK=2,NFC
      A(KK)=AA(KK)*RN
      B(KK)=BB(KK)*RN
      IF (RN.LE.TOL) GO TO 20
   10 RN=R(J)*RN
      KK=NFC
   20 DO 40 L=1,MM
      S=BB(1)
      DO 30 K=2,KK
      LT=1+MOD((K-1)*(L-1),M)
   30 S=S+A(K)*CO(LT)+B(K)*SI(LT)
   40 FP(L,J)=S
   50 FP(MP,J)=FP(2,J)
      DO 60 L=1,MP
      Z(L)=(0.,0.)
   60 FP(L,NN)=0.
      CALL MAP1
      RETURN
      END
      SUBROUTINE MAP1
      COMPLEX TMP,CEXP,TT,ZER,ONE
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMMON /E/ KCYCLE
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      DIMENSION XY(2)
      COMMON /G/ SS(310),TH(310),U(310),V(310),W(310),SP(310)
      EQUIVALENCE (XY(1),TMP)
      DATA ZER,ONE/(0.,0.),(1.,0.)/
C     DO MAPPING
      SN=2./AA(1)
```

```
      IF (KCYCLE.EQ.1) WRITE (N2,70)
      CALL MAP2
C     FIND SLOPES AT EQUALLY SPACED POINTS IN THE CIRCLE PLANE
      SP(MM)=PI
      CALL INTPL (MM,SP,FM,SS,TH,U,V,W)
C     COMPUTE ANGLE OF ZERO LIFT
      S=.5*(FM(1)+FM(MM))
      DO 10 L=2,M
   10 S=S+FM(L)
      BB(1)=-(.5*PI+S/FLOAT(M))
      S=-BB(1)*RAD
      IF (KCYCLE.EQ.1) WRITE (N2,50) S,BQ
C     COMPUTE DS
      DO 20 L=1,MM
      FM(L)=FM(L)+PI
      Q=FP(L,1)
      Z(L)=Q*CEXP((0.,1.)*FM(L))
   20 FP(L,1)=Q*Q
      FP(MP,1)=FP(2,1)
      Z(MP)=Z(2)
      TMP=ZER
      S=0.
      Q=0.
      BQ=0.
      DO 30 L=1,MM
      TT=TMP+.5*DT*(Z(L+1)+Z(L))
      Z(L)=TMP
      TMP=TT
      S=AMIN1(S,XY(1))
      Q=AMIN1(Q,XY(2))
      BQ=AMAX1(BQ,XY(2))
   30 FP(L,NN)=1.
      TC=(Q-BQ)/S
      CHD=-1./S
      DO 40 L=1,MM
   40 Z(L)=ONE+CHD*Z(L)
      IF (KCYCLE.EQ.1) WRITE (N2,60) TC
      CN=COS(BB(1)+ALP)
      SN=SIN(BB(1)+ALP)
      RETURN
C
C
   50 FORMAT (21HOANGLE OF ZERO LIFT =F9.5,7X,22HOUTER MAPPING RADIUS =F
     19.5)
   60 FORMAT (32H THE THICKNESS TO CHORD RATIO IS,F6.4/)
   70 FORMAT (1H0,5X,5HDEL S,8X,3HRES,9X,3HS/L,8X,4HW(0))
      END
      SUBROUTINE MAP2
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMMON /E/ KCYCLE
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      DIMENSION SPO(1)
      COMMON /G/ SS(310),TH(310),U(310),V(310),W(310),SP(310)
      EQUIVALENCE (SPO(1),Z(82))
      EE=.5*(1.-EPSIL)
      AQ=1.+EPSIL
      IM=N/2+1
C     COMPUTE ABS(1-SIGMA)**(1-EPSIL)
      DO 10 L=1,M
      FM(L)=1.
```

```
   10 PHIR(L)=(2.-2.*CO(L))**EE
      PHIR(MM)=1.
C     DO AT MOST NS CYCLES
      DO 120 K=1,NS
      XR=0.
      YR=0.
      X=0.
C     COMPUTE DS AND FIND THE MEAN VALUE OF FP AT R=.5
      DO 20 L=1,M
      XR=XR+FP(L,IM)
   20 SP(L)=EXP(FP(L,1))*PHIR(L)
      XR=XR/FLOAT(M)
      SP(MM)=SP(1)
      BQ=0.
C     COMPUTE ARC LENGTH AT EQUALLY SPACED POINTS IN CIRCLE PLANE
      DO 30 L=1,M
      AL=BQ+.5*DT*(SP(L+1)+SP(L))
      SP(L)=BQ
   30 BQ=AL
      SP(MM)=BQ
      BQ=AA(1)/AL
C     BQ IS THE RATIO OF ARC LENGTH BASED ON READ IN COORDINATES TO THE
C     ARC LENGTH COMPUTED IN THE CIRCLE PLANE
      IF (K.NE.1) GO TO 50
      DO 40 L=1,MM
   40 SPO(L)=BQ*SP(L)
   50 DO 60 L=1,MM
C     SET PP AT INFINITY TO MEAN VALUE TO ENSURE ANALYTICITY THERE
      FP(L,NN)=XR
C     SCALE ARC LENGTH TO TRUE ARC LENGTH
      SP(L)=BQ*SP(L)
C     COMPUTE MAXIMUM CHANGE IN ARC LENGTH AT EQUALLY SPACED POINTS
      AL=SP(L)-SPO(L)
      X=AMAX1(X,ABS(AL))
C     UPDATE ARC LENGTH AT EQUALLY SPACED POINTS IN THE CIRCLE PLANE
      SPO(L)=SPO(L)+XM*AL
C     COMPUTE T(S) SINCE SPLINE FIT USES T AS INDEPENDENT VARIABLE
   60 SP(L)=ACOS(1.-SN*SPO(L))
C     NORMALIZE X
      X=X/AA(1)
C     COMPUTE KAPPA AT THE POINTS CORRESPONDING TO SPO(L)
      CALL INTPL (M,SP,FM,SS,U,V,W,Z)
      DO 70 L=2,M
      FM(L)=SN*FM(L)/SIN(SP(L))
   70 CONTINUE
C     COMPUTE KAPPA*ABS(1-SIGMA)**(1-EPSIL) AT THE TAIL
      FM(MM)=.5*(FM(2)*PHIR(2)+FM(M)*PHIR(M))
      I=2
C     DO POINT RELAXATION
      CALL MAP3
      DO 80 J=1,NN
   80 FP(MP,J)=FP(2,J)
      DO 90 I=3,MM
   90 CALL MAP3
      DO 100 J=1,N
  100 FP(1,J)=FP(MM,J)
      IF (KCYCLE.NE.1) GO TO 110
      IF (MOD(K-1,KP).EQ.0) WRITE (N2,170) YR,X,BQ,XR
C     CHECK FOR CONVERGENCE
  110 IF (AMAX1(YR,X).LT.ST) GO TO 130
  120 CONTINUE
  130 NCY=K
C     NOW COMPUTE MAP FUNCTION
      DO 150 L=1,M
      DO 140 J=2,N
      Q=EXP(FP(L,J)-XR)*(1.+RS(J)-2.*R(J)*CO(L))**EE
```

```
  140 FP(L,J)=Q*Q
      FP(L,1)=PHIR(L)*EXP(FP(L,1)-XR)
      FP(L,NN)=1.
  150 CONTINUE
      DO 160 J=1,NN
      FP(MM,J)=FP(1,J)
  160 FP(MP,J)=FP(2,J)
C     COMPUTE OUTER MAPPING RADIUS
      BQ=EXP(XR)
      RETURN
C
  170 FORMAT (2E12.3,2F12.5)
      END
      SUBROUTINE MAP3
C     DO POINT RELAXATION FOR LAPLACES EQUATION IN POLAR COORDINATES
C     ALONG LINE I FROM J=N TO J = 1
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      J=N
   10 TA=RAS*RS(J)
      XX=(FP(I+1,J)+FP(I-1,J)+TA*(FP(I,J+1)+FP(I,J-1))+RA5*R(J)*(FP(I,J+
     11)-FP(I,J-1)))/(2.+TA+TA)
      XX=XX-FP(I,J)
      YR=AMAX1(ABS(XX),YR)
      FP(I,J)=FP(I,J)+XX*XS
      J=J-1
      IF (J.GT.1) GO TO 10
C     USE REFLECTION ON THE BOUNDARY
      TA=FP(1,2)-DR*(AQ+FM(I)*PHIR(I)*2.*EXP(FP(I,1)))
      XX=(FP(I+1,1)+FP(I-1,1)+RAS*(FP(I,2)+TA)+RA5*(FP(I,2)-TA))/(2.+2.*
     1RAS)
      XX=XX-FP(I,1)
      YR=AMAX1(ABS(XX),YR)
      FP(I,1)=FP(I,1)+XX*XS
      RETURN
      END
      SUBROUTINE MOVC (I1,W1,I2,W2)
      J1=(10-I1)*6
      J2=(10-I2)*6
      DO 10 K=1,6
      CALL SETBIT (J1,J2,W1,W2)
      J1=J1+1
   10 J2=J2+1
      RETURN
      END
      SUBROUTINE MURMAN
C     SET UP COEFFICIENT ARRAYS FOR THE TRIDIAGONAL SYSTEM USED FOR LINE
C     RELAXATION AND COMPUTE THE UPDATED PHI ON THIS LINE
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
C     DO THE BOUNDARY
      FAC=FLOAT(IM-IMM)
      KK=0
```

```
      U=RP(1)*DELTH-SI(I)
C     CHECK FOR THE TAIL POINT
      IF (I.EQ.MM) GO TO 10
      BQ=U/FP(I,1)
      AQ=U*BQ
      BQ=AQ*BQ*(FP(I-1,1)-FP(I+1,1))
   10 CS=C1-C2*AQ
      RP(1)=AQ
      C(1)=-CS*RAS
      C(1)=C(1)+C(1)
      A(1)=CS-AQ
      X=CO(I)*(C(1)*DR+RA4*(CS+AQ))
      IF (AQ.LE.QCRIT) GO TO 20
C     FLOW IS SUPERSONIC, BACKWARD DIFFERENCES
      D(1)=A(1)*(RA8*PHI(IMM,1)-RA9*PHI(IM,1)-EP*PHI(IM3,1))+RA3*BQ+X
      A(1)=-(C(1)+A(1)*RA7)
      KK=1
      GO TO 30
C     FLOW SUBCRITICAL, CENTRAL DIFFERENCES
   20 D(1)=A(1)*(PHI(I+1,1)+PHI(I-1,1))+RA3*BQ+X
      A(1)=A(1)+A(1)-C(1)
C     DO NON-BOUNDARY POINTS
   30 DO 50 J=2,N
      DU=RP(J)
      U=DU*R(J)*DELTH-SI(I)
      DV=PHI(I,J+1)-PHI(I,J-1)
      V=DV*RS(J)*DELR-CO(I)
      US=U*U
      VS=V*V
      BQ=1./FP(I,J)
      US=BQ*US
      VS=BQ*VS
      QS=US+VS
      RP(J)=QS
      CS=C1-C2*QS
      UVS=BQ*QS
      C(J)=RS(J)*(VS-CS)*RAS
      B(J-1)=C(J)
      A(J)=CS-US
      UV=BQ*U*V
      X=RA2*UV*R(J)
C     COMPUTE CONTRIBUTION OF RIGHT-HAND SIDE FROM LOW ORDER TERMS
      D(J)=RA5*(CS+US-VS-VS)*R(J)*DV-DT*UV*DU+RA3*UVS*(RI(J)*U*(FP(I-1,J
     1)-FP(I+1,J))+RA*V*(FP(I,J-1)-FP(I,J+1)))
      IF (QS.LE.QCRIT) GO TO 40
C     SUPERSONIC FLOW, USE BACKWARD DIFFERENCING
      KK=KK+1
      X=X*FAC
      D(J)=D(J)+A(J)*(RA8*PHI(IMM,J)-RA9*PHI(IM,J)-EP*PHI(IM3,J))+X*(EP*
     1(PHI(IMM,J+1)-PHI(IMM,J-1))+EL*(PHI(IM,J-1)-PHI(IM,J+1)))
      A(J)=-(RA7*A(J)+2.*C(J))
      QS=C3*X
      B(J-1)=B(J-1)+QS
      C(J)=C(J)-QS
      GO TO 50
   40 CONTINUE
C     SUBSONIC FLOW, USE CENTRAL DIFFERENCES
      D(J)=D(J)+A(J)*(PHI(I+1,J)+PHI(I-1,J))+X*(PHI(I+1,J+1)+PHI(I-1,J-1
     1)-PHI(I+1,J-1)-PHI(I-1,J+1))
      A(J)=2.*(A(J)-C(J))
   50 CONTINUE
C     ADJUST FOR BOUNDARY CONDITION AT INFINITY
      D(N)=D(N)-C(N)*PHI(I,NN)
      MXP=MAX0(MXP,KK)
C     SOLVE THE TRIDIAGONAL SYSTEM
      CALL TRID
      DO 70 J=1,N
C     FIND THE LOCATION OF THE MAXIMUM RESIDUAL
      IF (ABS(E(J)-PHI(I,J)).LE.YR) GO TO 60
```

```
      YR=ABS(E(J)-PHI(I,J))
      IK=I
      JK=J
C     COMPUTE RELAXATION FACTOR
   60 X=AMIN1(XS,AMAX1(XM,YM+Y*RP(J)))
C     SAVE OLD VALUE OF PHI
      RP(J)=PHI(I,J)
C     COMPUTE NEW PHI AT EACH POINT ON THE LINE
   70 PHI(I,J)=X*E(J)+(1.-X)*PHI(I,J)
      RETURN
      END
      FUNCTION ORDIN (A,G,X)
      DIMENSION A(110)
      J=INT(X/G)
   10 R=(X-AINT(X/G)*G)/G
      ORDIN=(1.-R)*A(J+1)+R*A(J+2)
      RETURN
      END
      SUBROUTINE PERMUT (AX,NX,JX)
C     REORDERS POINTS WITHIN AN ARRAY
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      DIMENSION AX(1)
      L=1
      JY=JX+JX
      NY=2*((NX-1)/2)+1
      NZ=2*(NX/2)
      IF (X.EQ.2.) GO TO 30
      NY=JX*(NY-1)+1
      NZ=JX*(NZ-1)
      DO 10 J=1,NY,JY
      A(L)=AX(J)
   10 L=L+1
      DO 20 J=JX,NZ,JY
      A(L)=AX(J+1)
   20 L=L+1
      GO TO 60
   30 DO 40 J=1,NY,2
      A(J)=AX(L)
   40 L=L+JX
      DO 50 J=2,NZ,2
      A(J)=AX(L)
   50 L=L+JX
   60 L=1
      DO 70 J=1,NX
      AX(L)=A(J)
   70 L=L+JX
      RETURN
      END
      SUBROUTINE PHIRR
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
```

```
C     ADJUST CONDITION AT INFINITY FOR MACH NUMBER AND ANGLE OF ATTACK
      BQ=-4.
      DO 20 L=1,MP
      PHIR(L)=ATAN2(-BET*SI(L),-CO(L))
   10 IF (PHIR(L).GE.BQ) GO TO 20
      PHIR(L)=PHIR(L)+TP
      GO TO 10
   20 BQ=PHIR(L)
      RETURN
      END
      SUBROUTINE PLOT (X,Y,I)
      CALL CALPLT (X,Y,I)
      RETURN
      END
      SUBROUTINE PLOTN (NPLT,NBIN,NLIN,XAXIS,YAXIS,PSYMBL,MAXPLT,MAXPT)
C     GENERATES OUTPUT PLOTS INCLUDED IN THE LISTING
      DIMENSION XAXIS(MAXPT), PSYMBL(MAXPLT), YAXIS(MAXPT,MAXPLT)
      DIMENSION PS(10)
      DATA BLANK/1H /,
      DATA ZLINE/10H----------/
      ITEST=0
      NHI=(NBIN+9)/10
      IF (NPLT.GT.0) GO TO 10
      NPLT=-NPLT
      ITEST=1
   10 CONTINUE
      YMAX=YAXIS(1,1)
      YMIN=YMAX
      DO 20 J=1,NPLT
      DO 20 I=1,NBIN
      YMAX=AMAX1(YMAX,YAXIS(I,J))
   20 YMIN=AMIN1(YMIN,YAXIS(I,J))
      YSPAN=YMAX-YMIN
      AYMX=YMAX
      HYMX=YMAX
      IF (YMIN.GE.0.0) GO TO 30
      IF ((YMAX.GT.0.0).AND.(YMIN.LT.0.0)) GO TO 50
      HYMX=YMIN
      AYMX=-YMIN
      YMIN=YMAX
      YMAX=HYMX
   30 IF (YSPAN.GT.(AYMX/2.)) GO TO 50
      IF (ABS(YSPAN).LE.1.E-15) YSPAN=10.
      HYMX=HYMX/2.
      DO 40 J=1,NPLT
      DO 40 I=1,NBIN
   40 YAXIS(I,J)=YAXIS(I,J)-HYMX
      AYMX=AYMX/2.
      GO TO 30
   50 CONTINUE
      NLINES=((IABS(NLIN)-1)/50+1)*50
      DY=AMAX1(YSPAN,AYMX)/FLOAT(NLINES)
      RESTY=YMAX-HYMX
      HYMX=DIM(HYMX,0.0)
      XMYH=HYMX
      NIBN=NBIN
      NGROS=(NBIN-1)/100
      IGROS=NGROS
      HYMX=XMYH
      PRINT 170
      IF (IGROS.EQ.0) GO TO 60
      NBIN=100
      GO TO 70
   60 NBIN=NIBN-NGROS*100
   70 NHI=(NBIN+9)/10
      ICOR=100*(NGROS-IGROS)
      DO 80 J=1,NPLT
      DO 80 I=1,NBIN
```

```
   80 YAXIS(I,J)=YAXIS(I+ICOR,J)
      DO 90 I=1,NBIN
   90 XAXIS(I)=XAXIS(I+ICOR)
      LFLG=0
      DO 140 K=1,NLINES
      IF ((HYMX.GT.0.0).OR.(LFLG.NE.0)) GO TO 110
      LFLG=1
      DO 100 I=1,NHI
  100 PS(I)=ZLINE
      PRINT 150, (PS(I),I=1,NHI)
  110 DO 120 I=1,NHI
  120 PS(I)=BLANK
      DO 130 J=1,NPLT
      DO 130 I=1,NBIN
      YI=YAXIS(I,J)
      IF ((HYMX.LT.YI).OR.((HYMX-YI).GT.DY)) GO TO 130
      I1=(I+9)/10
      I2=I-(I1-1)*10
      CALL MOVC (1,PSYMBL(J),I2,PS(I1))
  130 CONTINUE
      CONT=REDUC(HYMX+RESTY)
      IF (ITEST.EQ.1) CONT=-CONT
      PRINT 160, CONT,(PS(I),I=1,NHI)
      PRINT 180
  140 HYMX=HYMX-DY
      RETURN
C
C
  150 FORMAT (1H+,19X,10A10)
  160 FORMAT (1H+,F12.4,7X,10A10)
  170 FORMAT (1H1)
  180 FORMAT (1H )
      END
      FUNCTION REDUC (X)
      REDUC=X
      RETURN
      END
      SUBROUTINE REFINE
C     HALVES THE MESH SIZE
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      X=2.
      M=M+M
      MM=M+1
      N=N+N
      NN=N+1
      II=II+II-1
      JJ=JJ+JJ-1
      CALL BOTH
      RETURN
      END
      SUBROUTINE RESTRT
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
```

```
     1),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      COMMON /E/ KCYCLE,DUMY(1836),INVDIV
C     SET UP CONSTANTS
      TP=PI+PI
      TPI=1./TP
      RAD=180./PI
      IF (KCYCLE.EQ.1) ITYP=1
      MM=M+1
      MP=MM+1
      LL=MP/2
      LP=LL+2
      NN=N+1
      DR=-1./FLOAT(N)
      DT=TP/FLOAT(M)
      DELR=.5/DR
      DELTH=.5/DT
      RA=DT/DR
      RAS=RA*RA
      RA2=-.5*RA
      RA3=-.125/DELTH
      RA4=.25/(DELTH*DELTH)
      RA5=-RA*(RA3+RA3)
      DO 10 K=1,N
      R(K)=1.+DR*FLOAT(K-1)
      RS(K)=R(K)*R(K)
      RI(K)=1./R(K)
   10 CONTINUE
      R(NN)=0.
      RS(NN)=0.
      DO 20 L=1,MP
      TH=FLOAT(L-1)*DT
      CO(L)=COS(TH)
   20 SI(L)=SIN(TH)
      CALL MAP
C     DO NOT REINITIALIZE PHI ON CYCLES OTHER THAN THE FIRST, UNLESS
C     THE INVISCID SOLUTION DIVERGED
      IF (KCYCLE.EQ.1) GO TO 30
      IF (INVDIV.EQ.1) GO TO 30
      RETURN
   30 BQ=-4.
      DPHI=PI*SN/(CHD*BET)
      CL=TP*SN/BET
      DO 70 L=1,M
      X=CO(L)
      CO(L)=X*CN-SI(L)*SN
      SI(L)=SI(L)*CN+X*SN
      PHIR(L)=ATAN2(-SI(L)*BET,-CO(L))
   40 IF (PHIR(L).GE.BQ) GO TO 50
      PHIR(L)=PHIR(L)+TP
      GO TO 40
   50 BQ=PHIR(L)
      DO 60 J=1,NN
   60 PHI(L,J)=R(J)*CO(L)+DPHI*PHIR(L)*TPI
   70 CONTINUE
      PHIR(MM)=PHIR(1)+TP
      PHIR(MP)=PHIR(2)+TP
      DO 80 J=1,NN
      PHI(MM,J)=PHI(1,J)+DPHI
   80 PHI(MP,J)=PHI(2,J)+DPHI
      CALL INIT
      RETURN
      END
      FUNCTION RLORD (Z)
      RLORD=.4*Z
      IF (Z-.18) 40,10,10
   10 IF (Z-1.1) 20,30,30
   20 RLORD=.095-.055*(2.*Z-1.)**2
```

```
      GO TO 40
   30 RLORD=.016*EXP(-10.*(Z-1.1))
   40 RETURN
      END
      SUBROUTINE SETCP
C     INTERPOLATES FOR CP AT INPUT STATIONS TO B. L. CALCULATION
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMMON /E/ KCYCLE,FNU,FNL,IBNDLAY(13),ABNDLAY(19),XBODY(162),YBODY
     1(162),BODSLOP(162),BODCURV(82),XNEW(162),YNEW(162),CPSURF(162),CPB
     2DLY(82),IGRID,GRID,XUPLE(20),XUPARC(20),XLOLE(20),XLOARC(20),TITLO
     3UT(15),CLOUT(3,7),SUPOUT(3,7),SLOWOUT(3,7),CMOUT,CPUP(85),CPLO(85)
     4,XTEMUP(85),XTEMLO(85),DELBLX(162),KOUNT,KOUNTUP
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      MNEG=M/2-5
      MPOS=M/2+5
      KSTOP=M+1
C     REORDER CP DISTRIBUTION
      CPNOS1=CPSURF(MNEG)
      KOUNT1=MNEG
      DO 10 L=MNEG,MPOS
      IF (CPSURF(L).LE.CPNOS1) GO TO 10
      CPNOS1=CPSURF(L)
      KOUNT1=L
   10 CONTINUE
      XNOS2=REAL(Z(MNEG))
      KOUNT2=MNEG
      DO 20 L=MNEG,MPOS
      IF (REAL(Z(L)).GE.XNOS2) GO TO 20
      XNOS2=REAL(Z(L))
      KOUNT2=L
   20 CONTINUE
      IF (KOUNT1.GT.KOUNT2) GO TO 30
      KOUNT=KOUNT1
      KOUNTUP=M-KOUNT2+2
      KSTART=KOUNT2
      GO TO 40
   30 KOUNT=KOUNT2
      KOUNTUP=M-KOUNT1+2
      KSTART=KOUNT1
   40 DO 50 L=1,KOUNT
      J=KOUNT-L+1
      XTEMLO(L)=REAL(Z(J))
   50 CPLO(L)=CPSURF(J)
      DO 60 L=KSTART,KSTOP
      J=L-KSTART+1
      XTEMUP(J)=REAL(Z(L))
   60 CPUP(J)=CPSURF(L)
      KMAXCYC=IBNDLAY(12)
C     RETURN IF KCYCLE IS A MAXIMUM
      IF (KCYCLE.EQ.KMAXCYC) GO TO 180
C     DEFINE ADDITIONAL INPUTS TO BOUNDARY LAYER ROUTINE
      JUP=1
   70 IF (CPUP(JUP).LT.1.0.AND.XTEMUP(JUP).GT.XUPLE(1)) GO TO 80
      JUP=JUP+1
      GO TO 70
   80 CALL DISCOT (XTEMUP(JUP),XTEMUP(JUP),XUPLE,XUPARC,XUPARC,-010,20,0
     1,ARC1)
      RNLDS=ABNDLAY(18)*10.**6
      AMNTM1=0.292/SQRT(RNLDS/ARC1*SQRT(1.0-CPUP(JUP)))
      JLO=1
   90 IF (CPLO(JLO).LT.1.0.AND.XTEMLO(JLO).GT.XLOLE(1)) GO TO 100
      JLO=JLO+1
      GO TO 90
```

```
100 CALL DISCOT (XTEMLO(JLO),XTEMLO(JLO),XLOLE,XLOARC,XLOARC,-010,20,0
   1,ARC2)
    AMNTM2=0.292/SQRT(RNLDS/ARC2*SQRT(1.0-CPLO(JLO)))
    ABNDLAY(1)=(AMNTM1+AMNTM2)/2.0
    ABNDLAY(4)=RNLDS*ABNDLAY(1)
    NPRES=IBNDLAY(5)
    I13=IBNDLAY(13)
    XTEMLO(1)=0.0
    XTEMUP(1)=0.0
C   INTERPOLATE FOR CP AT REQUIRED X/C
    DO 110 J=1,NPRES
    XPRES=ABNDLAY(3)+FLOAT(J-1)*ABNDLAY(2)
    CALL DISCOT (XPRES,XPRES,XTEMLO,CPLO,CPLO,-010,KOUNT,0,CPBDLY(J))
    JPL=J+NPRES
    CALL DISCOT (XPRES,XPRES,XTEMUP,CPUP,CPUP,-010,KOUNTUP,0,CPBDLY(JP
   1L))
110 CONTINUE
C   LIMIT CP AT NOSE SO B. L. ROUTINE DOES NOT BLOW UP
    IF (CPBDLY(1).LE.0.8) GO TO 120
    CPBDLY(1)=0.8
120 IF (CPBDLY(NPRES+1).LE.0.8) GO TO 130
    CPBDLY(NPRES+1)=0.8
130 IF (NPRES-I13.LE.0) GO TO 180
C   ADJUST AFT SURFACE PRESSURE COEFFICIENTS FOR BOUNDARY LAYER INPUT
    NMAX=NPRES/2+1
    CPMAXLO=CPBDLY(NMAX-1)
    JMAX=NMAX-1
    DO 140 J=NMAX,NPRES
    IF (CPBDLY(J).LT.CPMAXLO) GO TO 140
    CPMAXLO=CPBDLY(J)
    JMAX=J
140 CONTINUE
    DO 150 J=JMAX,NPRES
150 CPBDLY(J)=CPMAXLO
    IF (NPRES-I13.LT.2) GO TO 170
    X1=ABNDLAY(3)+FLOAT(I13-2)*ABNDLAY(2)
    X2=ABNDLAY(3)+FLOAT(I13-1)*ABNDLAY(2)
    X3=ABNDLAY(3)+FLOAT(NPRES-1)*ABNDLAY(2)
    Y1=CPBDLY(NPRES+I13-1)
    Y2=CPBDLY(NPRES+I13)
    Y3=CPBDLY(NPRES)
    APARAB=((Y2-Y1)*(X3-X1)-(Y3-Y1)*(X2-X1))/(X2-X1)/(X3-X1)/(X2-X3)
    BPARAB=(Y3-Y1-APARAB*(X3*X3-X1*X1))/(X3-X1)
    CPARAB=Y1-BPARAB*X1-APARAB*X1*X1
    I13P1=I13+1
    DO 160 J=I13P1,NPRES
    JPL=J+NPRES
    XPRES=ABNDLAY(3)+FLOAT(J-1)*ABNDLAY(2)
160 CPBDLY(JPL)=APARAB*XPRES*XPRES+BPARAB*XPRES+CPARAB
    GO TO 180
170 CPBDLY(NPRES+NPRES)=CPBDLY(NPRES)
180 RETURN
    END
    SUBROUTINE SICO
    COMMON PHI(162,33),FP(162,33)
    COMMON /B/ AA(100),BB(100)
    COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
   1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
   2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
   3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
   4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
    COMPLEX Z
    COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
   11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
C   COMPUTE COS(THETA+ALPO),SIN(THETA+ALPO)
    CN=COS(ALP-ALPO)
    SN=SIN(ALP-ALPO)
    CALL COSI
    RETURN
    END
```

APPENDIX

```
      FUNCTION SIMPSN (FR,IA,N,H)
      DIMENSION FR(110)
      J=(N-IA)/2
      IF (N-IA-2*J) 30,10,30
   10 S=0.
      N1=N-1
      DO 20 I=IA,N1,2
   20 S=S+H*(FR(I)+4.*FR(I+1)+FR(I+2))/3.
      GO TO 50
   30 S=H*(5.*FR(IA)+8.*FR(IA+1)-FR(IA+2))/12.
      IA1=IA+1
      N1=N-1
      DO 40 I=IA1,N1,2
   40 S=S+H*(FR(I)+4.*FR(I+1)+FR(I+2))/3.
   50 SIMPSN=S
      RETURN
      END
      FUNCTION SLOG (Z)
      SLOG=19.*(1.-EXP(-2.083*Z))+78./(1.+655.*(ABS(Z-.76))**3)
      RETURN
      END
      FUNCTION SLOPE (A,G,N,X)
      DIMENSION A(110)
      J=INT(X/G)
   10 R=(X-AINT(X/G)*G)/G
      SLOPE=(1.-R)*GRAD(A,G,1,N,J+1)+R*GRAD(A,G,1,N,J+2)
      RETURN
      END
      SUBROUTINE SOLVE (D,E,DET,A)
      DIMENSION A(2,3)
      DET=A(1,1)*A(2,2)-A(2,1)*A(1,2)
      IF (DET) 10,20,10
   10 D=(A(1,2)*A(2,3)-A(2,2)*A(1,3))/DET
      E=(A(1,3)*A(2,1)-A(2,3)*A(1,1))/DET
   20 CONTINUE
      RETURN
      END
      SUBROUTINE SPLIF (N,S,F,FP,FPP,FPPP,KM,VM,KN,VN)
C     GIVEN S AND F AT N CORRESPONDING POINTS, FIT A CUBIC SPLINE
      DIMENSION S(1), F(1), FP(1), FPP(1), FPPP(1)
      K=1
      M=1
      I=M
      J=M+K
      DS=S(J)-S(I)
      D=DS
      IF (DS.EQ.0.) GO TO 110
      DF=(F(J)-F(I))/DS
      IF (KM-2) 10,20,30
   10 U=.5
      V=3.*(DF-VM)/DS
      GO TO 50
   20 U=0.
      V=VM
      GO TO 50
   30 U=-1.
      V=-DS*VM
      GO TO 50
   40 I=J
      J=J+K
      DS=S(J)-S(I)
      IF (D*DS.LE.0.) GO TO 110
      DF=(F(J)-F(I))/DS
      B=1./(DS+DS+U)
      U=B*DS
      V=B*(6.*DF-V)
   50 FP(I)=U
      FPP(I)=V
```

```
      U=(2.-U)*DS
      V=6.*DF+DS*V
      IF (J.NE.N) GO TO 40
      IF (KN-2) 60,70,80
   60 V=(6.*VN-V)/U
      GO TO 90
   70 V=VN
      GO TO 90
   80 V=(DS*VN+FPP(I))/(1.+FP(I))
   90 B=V
      D=DS
  100 DS=S(J)-S(I)
      U=FPP(I)-FP(I)*V
      FPPP(I)=(V-U)/DS
      FPP(I)=U
      FP(I)=(F(J)-F(I))/DS-DS*(V+U+U)/6.
      V=U
      J=I
      I=I-K
      IF (J.NE.M) GO TO 100
      FPPP(N)=FPPP(N-1)
      FPP(N)=B
      FP(N)=DF+D*(FPP(N-1)+B+B)/6.
      RETURN
  110 STOP
      END
      SUBROUTINE SWEEP
C     SWEEP THROUGH THE GRID ONE TIME
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      YR=0.
      IK=LL
      JK=1
      DO 10 J=1,N
   10 RP(J)=PHI(LL-1,J)
      MXP=0
C     SWEEP THROUGH THE GRID FROM NOSE TO TAIL ON UPPER SURFACE
      DO 30 I=LL,M
      IM=I-1
      IMM=I-2
      IM3=I-3
      DO 20 J=1,N
   20 RP(J)=PHI(I+1,J)-RP(J)
   30 CALL MURMAN
      AQ=0.
      BQ=0.
      I=MM
      DO 40 J=1,N
   40 RP(J)=PHI(MP,J)-RP(J)
      CALL MURMAN
C     UPDATE PHI AT THE TAIL FROM UPPER SURFACE
      DO 50 J=1,N
   50 PHI(1,J)=PHI(MM,J)-DPHI
C     SWEEP THROUGH THE GRID FROM NOSE TO TAIL ON LOWER SURFACE
      DO 60 J=1,N
   60 RP(J)=PHI(LL,J)
      DO 80 J=3,LL
      I=LP-J
      IM=I+1
```

```
      IMM=I+2
      IM3=I+3
      DO 70 L=1,N
   70 RP(L)=RP(L)-PHI(I-1,L)
   80 CALL MURMAN
C     ADJUST CIRCULATION TO SATISFY THE KUTTA CONDITION
      IF (XPHI.EQ.0.) GO TO 90
      YA=XPHI*((PHI(M,1)-PHI(MP,1))*DELTH+SI(1))
   90 DPHI=DPHI+YA
      YA=YA*TPI
      DO 100 L=1,MP
      PHI(L,NN)=DPHI*TPI*PHIR(L)
      DO 100 J=1,N
  100 PHI(L,J)=PHI(L,J)+YA*PHIR(L)
      DO 110 J=1,N
  110 PHI(MP,J)=PHI(2,J)+DPHI
      RETURN
      END
      SUBROUTINE SYMBOL (X,Y,H,N,ANGLE,NCHAR)
      IF (NCHAR.GT.0) GO TO 20
      IF (N.EQ.26) GO TO 10
      IF (N.EQ.20) GO TO 10
      IF (N.EQ.17) GO TO 10
      GO TO 20
   10 IF (N.EQ.26) N=13
      SINA=SIN(ANGLE)
      COSA=COS(ANGLE)
      X=X+.5*H*(SINA-COSA)
      Y=Y-.5*H*(SINA+COSA)
   20 CONTINUE
      CALL NOTATE (X,Y,H,N,ANGLE,NCHAR)
      RETURN
      END
      SUBROUTINE TANCAL (G,A,V,J,RM,D,U,T,TA,TB,W)
      DIMENSION A(15), V(110), U(110), T(110), TA(110), TB(110), W(110)
      G=0.
      AAJ=FLOAT(J)*A(1)/A(6)
      D=A(8)*RM*GORD(AAJ)
      C=W(J)*T(J)
      P1=(V(J)+(A(15)-.5)*C+D)/U(J)
      TEMP=(D+(A(15)+.5)*C)**2+2.*(A(8)*T(J)+D*C)
      IF (TEMP.GE.0.) GO TO 10
      TA(J)=10.**71.
      TB(J)=TA(J)
      GO TO 20
   10 P2=SQRT(TEMP)/U(J)
      TA(J)=P1+P2
      TB(J)=P1-P2
   20 RETURN
      END
      SUBROUTINE TRID
C     SOLVE N DIMENSIONAL TRIDIAGONAL SYSTEM OF EQUATIONS
      COMMON PHI(162,33),FP(162,33)
      COMMON /B/ AA(100),BB(100)
      COMMON /C/ M,MM,MP,N,NN,LL,LP,I,IM,IMM,IM3,II,JJ,IK,JK,IZ,ITYP,MXP
     1,NS,NCY,TE,PI,RAD,TP,TPI,DT,DR,DELTH,DELR,RA,RAS,RA2,RA3,RA4,RA5,E
     2M,QCRIT,C1,C2,C4,C5,C6,C7,BET,EPSIL,TC,CL,CHD,ALP,ALPO,DPHI,XPHI,C
     3N,SN,EP,C3,RA7,RA8,RA9,EL,XM,XS,FSYM,ST,X,Y,YM,XA,YA,AQ,BQ,KP,YR,E
     4MO,EE,IDIM,NFC,NMP,IS,N2,N3,N4,N5,M4,NRN
      COMPLEX Z
      COMMON /A/ A(40),B(40),C(40),D(40),E(40),RHO(40),RP(40),R(41),RS(4
     11),RI(41),SI(162),CO(162),Z(162),FM(162),PHIR(162)
      XX=1./A(1)
      E(1)=XX*D(1)
C     DO ELIMINATION
      DO 10 J=2,N
      C(J-1)=C(J-1)*XX
      XX=1./(A(J)-B(J-1)*C(J-1))
   10 E(J)=(D(J)-B(J-1)*E(J-1))*XX
```

```
C     DO BACK SUBSTITUTION
      DO 20 J=2,N
      L=NN-J
   20 E(L)=E(L)-C(L)*E(L+1)
      RETURN
      END
      SUBROUTINE UNS (IC,IA,IDX,IDZ,IMS)
      IF (IC) 10,10,20
   10 IMS=1
      NC=-IC
      GO TO 30
   20 IMS=0
      NC=IC
   30 IF (NC-100) 40,50,50
   40 IA=0
      GO TO 60
   50 IA=1
      NC=NC-100
   60 IDX=NC/10
      IDZ=NC-IDX*10
      RETURN
      END
      SUBROUTINE XYAXES (X,BOT,TOP,SCF)
C     X IS THE LOCATION OF THE ORIGIN ON THE AXIS
C     BOT IS THE LENGTH OF THE AXIS TO THE LEFT OF THE ORIGIN
C     TOP IS THE LENGTH TO THE RIGHT OF THE ORIGIN
      COMPLEX ZB,ZT,H,COR
      COMMON /D/ SF,SIZE,ANG,XMAX,YMAX,XOR,YOR,PGSIZ
      DIMENSION X(2), Y(2)
      ANGO=ANG
      SFO=SF
      SIZO=SIZE
      Y(1)=XOR
      Y(2)=YOR
      ANG=0.
      SF=1.
      SIZE=.14
      XOR=X(1)+XOR
      YOR=X(2)+YOR
      ZB=CMPLX(-BOT,0.)
      ZT=CMPLX(TOP,0.)
      COR=(-.25,-.3)
      NC=16
      IF (ABS(ANGO).NE.90.) GO TO 10
C     VERTICAL Y-AXIS
      ZB=(0.,1.)*ZB
      ZT=(0.,1.)*ZT
      COR=(-.6,0.)
      NC=15
C     DRAW LINE FOR THE AXIS
   10 CALL CPLOT (ZT,3)
      CALL CPLOT (ZB,2)
      K=BOT
      L=TOP
      N=1+K+L
      S=-FLOAT(K)*SCF
      H=ZT/TOP
      ZB=-FLOAT(K)*H
      ZT=ZB+COR
      DO 20 I=1,N
C     DRAW HATCH MARK
      CALL CSYMBL (ZB,NC,-1)
      B=S+FLOAT(I-1)*SCF
      ENCODE (10,30,A) B
```

```
C     LABEL AXIS
      CALL CSYMBL (ZT,A,4)
      ZB=ZB+H
   20 ZT=ZT+H
      SF=SFO
      SIZE=SIZO
      ANG=ANGO
      XOR=Y(1)
      YOR=Y(2)
      RETURN
C
   30 FORMAT (F4.1)
      END
```

REFERENCES

1. Stevens, W. A.; Goradia, S. H.; and Braden, J. A.: Mathematical Model for Two-Dimensional Multi-Component Airfoils in Viscous Flow. NASA CR-1843, 1971.

2. Nieuwland, G. Y.; and Spee, B. M.: Transonic Airfoils: Recent Developments in Theory, Experiment, and Design. Annual Review of Fluid Mechanics, Vol. 5, Milton Van Dyke, Walter G. Vincenti, and J. V. Wehausen, eds., Annual Reviews, Inc., c.1973, pp. 119-150.

3. Garabedian, P. R.; and Korn, D. G.: Analysis of Transonic Airfoils. Commun. Pure & Appl. Math., vol. 24, no. 6, Nov. 1971, pp. 841-851.

4. Murman, Earll M.; and Cole, Julian D.: Calculation of Plane Steady Transonic Flows. AIAA J., vol. 9, no. 1, Jan. 1971, pp. 114-121.

5. Jameson, Antony: Transonic Flow Calculations for Airfoils and Bodies of Revolution. Rep. 390-71-1, Grumman Aerospace Corp., 1971.

6. Sells, C. C. L.: Plane Subcritical Flow Past a Lifting Aerofoil. Proc. Roy. Soc. (London), ser. A, vol. 308, no. 1494, Jan. 14, 1968, pp. 377-401.

7. Bradshaw, P.; Ferriss, D. H.; and Atwell, N. P.: Calculation of Boundary-Layer Development Using the Turbulent Energy Equation. J. Fluid Mech., vol. 28, pt. 3, May 26, 1967, pp. 593-616.

8. Bradshaw, P.; and Ferriss, D. H.: Calculation of Boundary-Layer Development Using the Turbulent Energy Equation: Compressible Flow on Adiabatic Walls. J. Fluid Mech., vol. 46, pt. 1, Mar. 15, 1971.

9. Morkovin, Mark V.: Effects of Compressibility on Turbulent Flows. The Mechanics of Turbulence, Gordan & Breach, Sci. Publ., Inc., c.1964, pp. 367-380.

10. Bauer, F.; Garabedian, P.; and Korn, D.: A Theory of Supercritical Wing Sections, With Computer Programs and Examples. Lecture Notes in Economics and Mathematical Systems, M. Beckmann and H. P. Kunzi, eds., Springer-Verlag, 1972.

11. Ferriss, D. H.; and Bradshaw, P.: A Computer Program for the Calculation of Boundary-Layer Development Using the Turbulent Energy Equation. NPL Aero. Rep. 1269, Brit. A.R.C., May 16, 1968.

12. Kacprzynski, J. J.; Ohman, L. H.; Garabedian, P. R.; and Korn, D. G.: Analysis of the Flow Past a Shockless Lifting Airfoil in Design and Off-Design Conditions. LR-554 (NRC No. 12315), Nat. Res. Counc. Can. (Ottawa), Nov. 1971. (Also available as AIAA paper No. 71-567.)

13. Melnik, R. E.; and Ives, D. C.: On Viscous and Wind-Tunnel Wall Effects in Transonic Flows Over Airfoils. AIAA Paper No. 73-660, July 1973.

14. Murman, Earll M.: Analysis of Embedded Shock Waves Calculated by Relaxation Methods. AIAA J., vol. 12, no. 5, May 1974, pp. 626-633.

15. Steger, Joseph L.; and Baldwin, Barrett S.: Shock Waves and Drag in the Numerical Calculation of Isentropic Transonic Flow. NASA TN D-6997, 1972.

16. Stivers, Louis S., Jr.: Effects of Subsonic Mach Number on the Forces and Pressure Distributions on Four NACA 64A-Series Airfoil Sections at Angles of Attack as High as 28°. NACA TN 3162, 1954.

17. Whitcomb, Richard T.: Review of NASA Supercritical Airfoils. ICAS Paper No. 74-No. 74-10, Aug. 1974.

18. Stratford, B. S.: The Prediction of Separation of the Turbulent Boundary Layer. J. Fluid Mech., vol. 5, pt. 1, Jan. 1959, pp. 1-16.

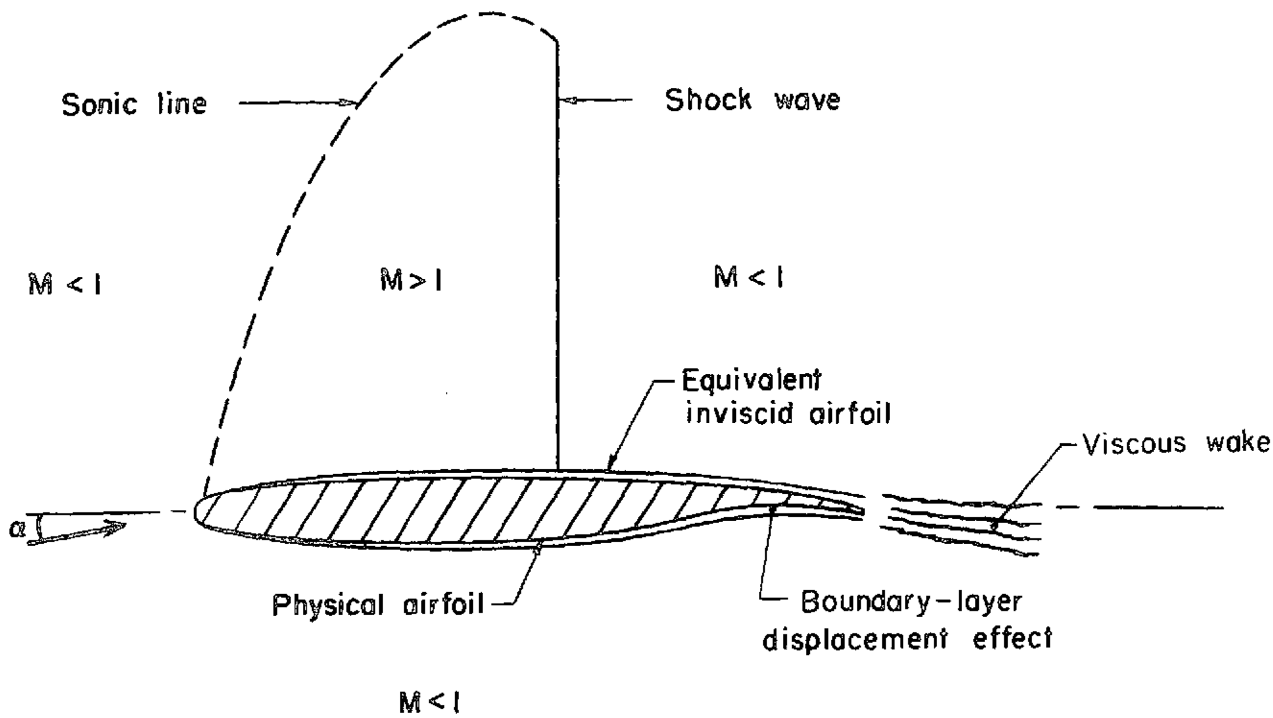


Figure 1.- Illustration of fundamental parameters.

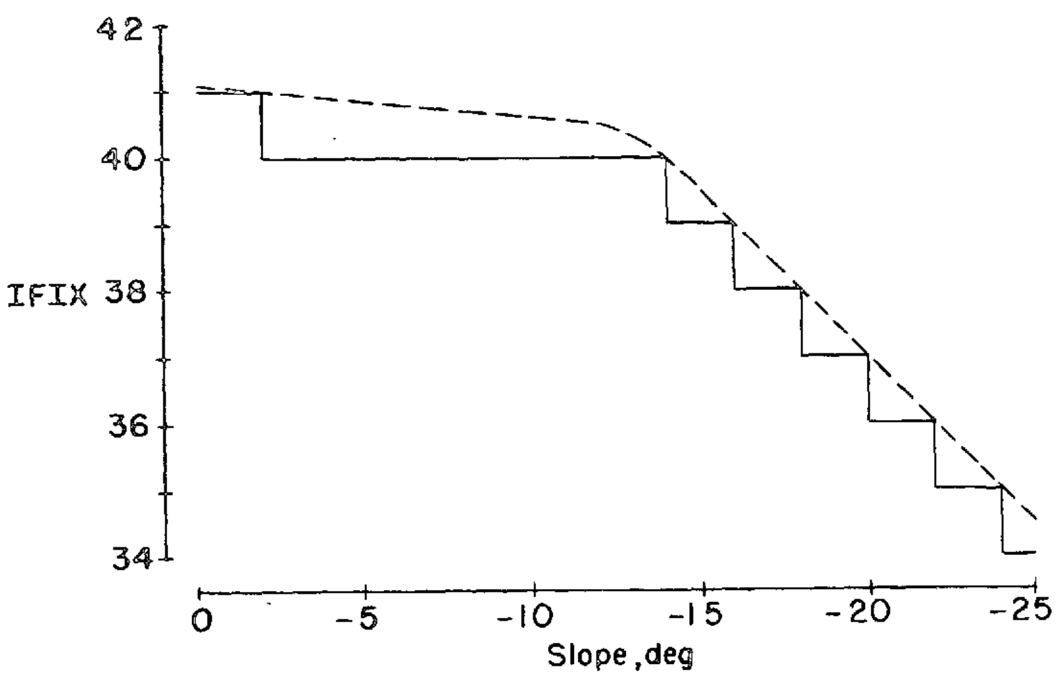


Figure 2.- Variation of empirical trailing-edge deviation point with slope.

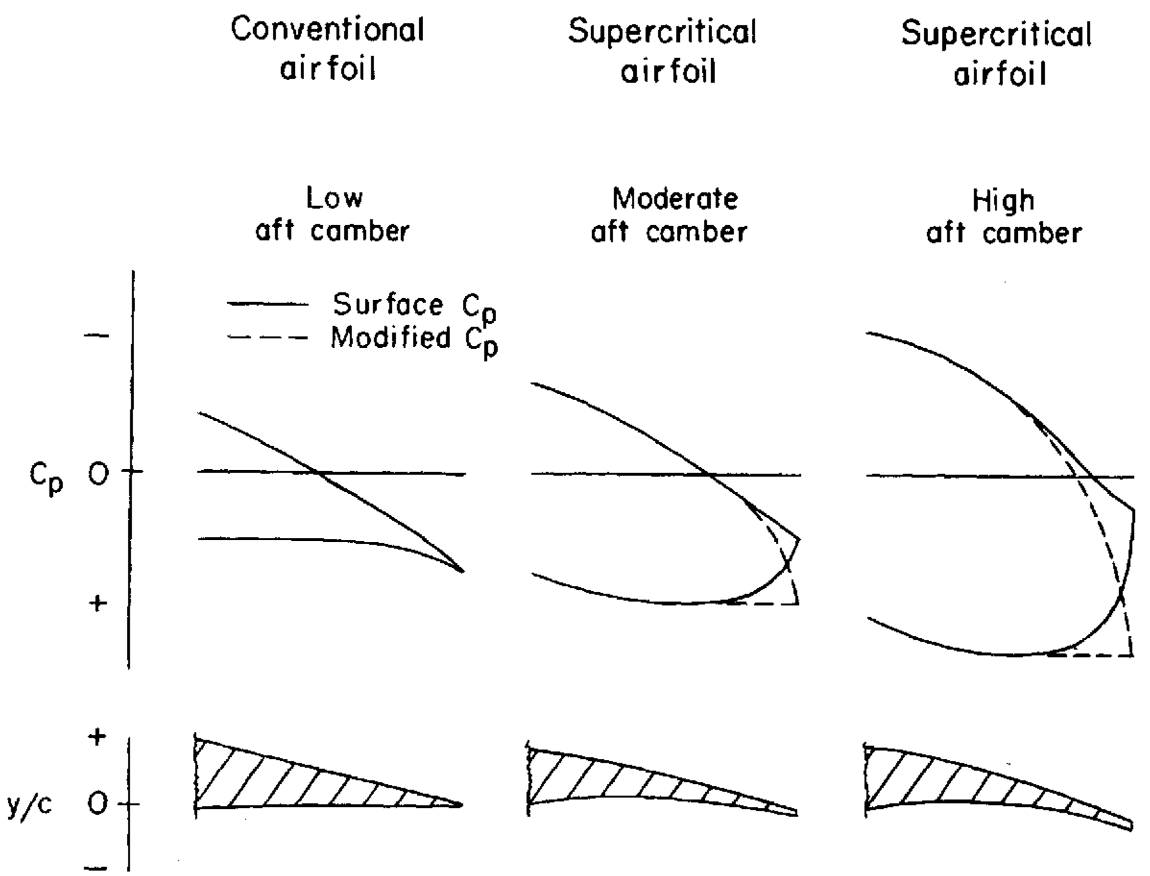
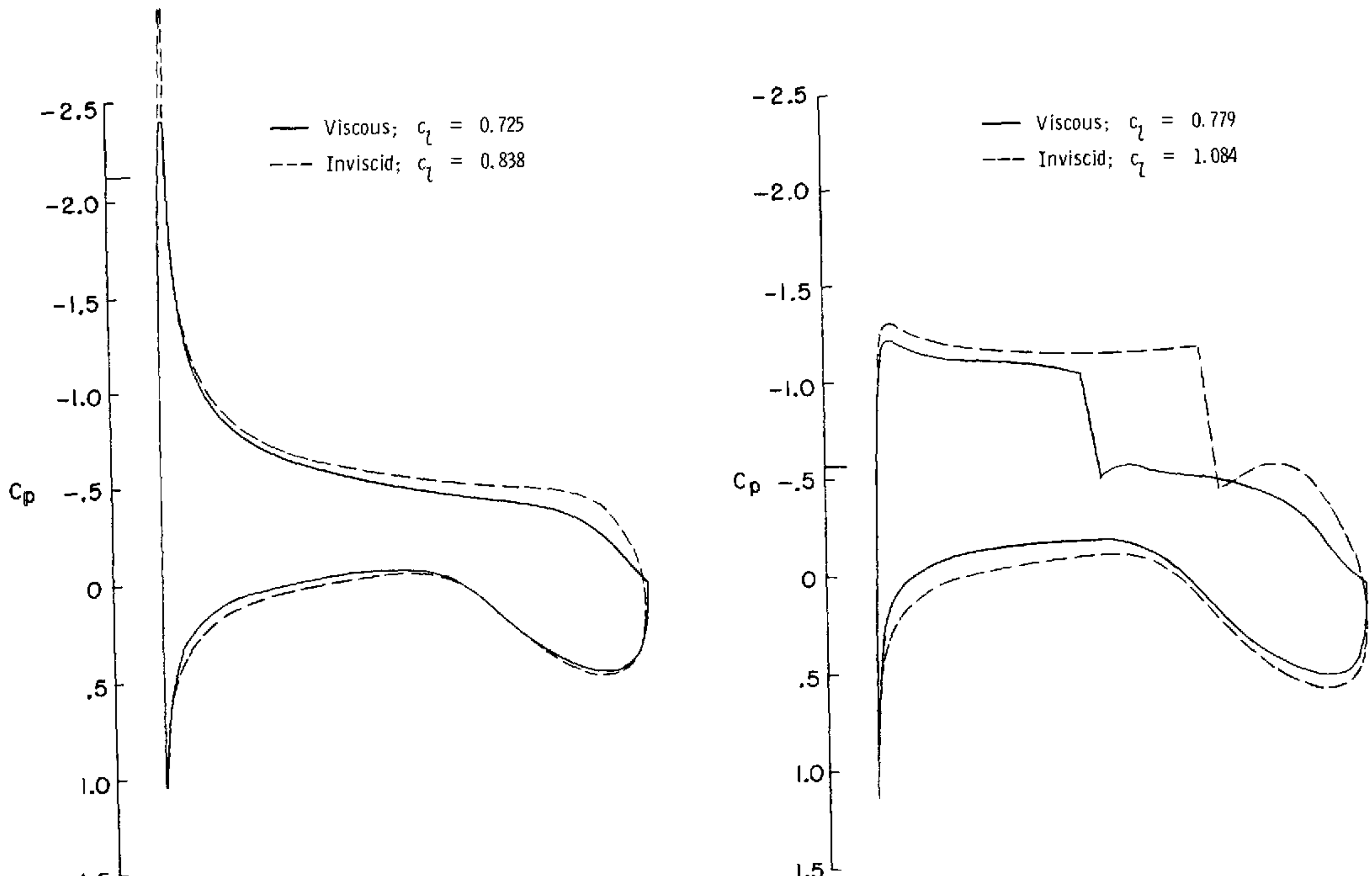


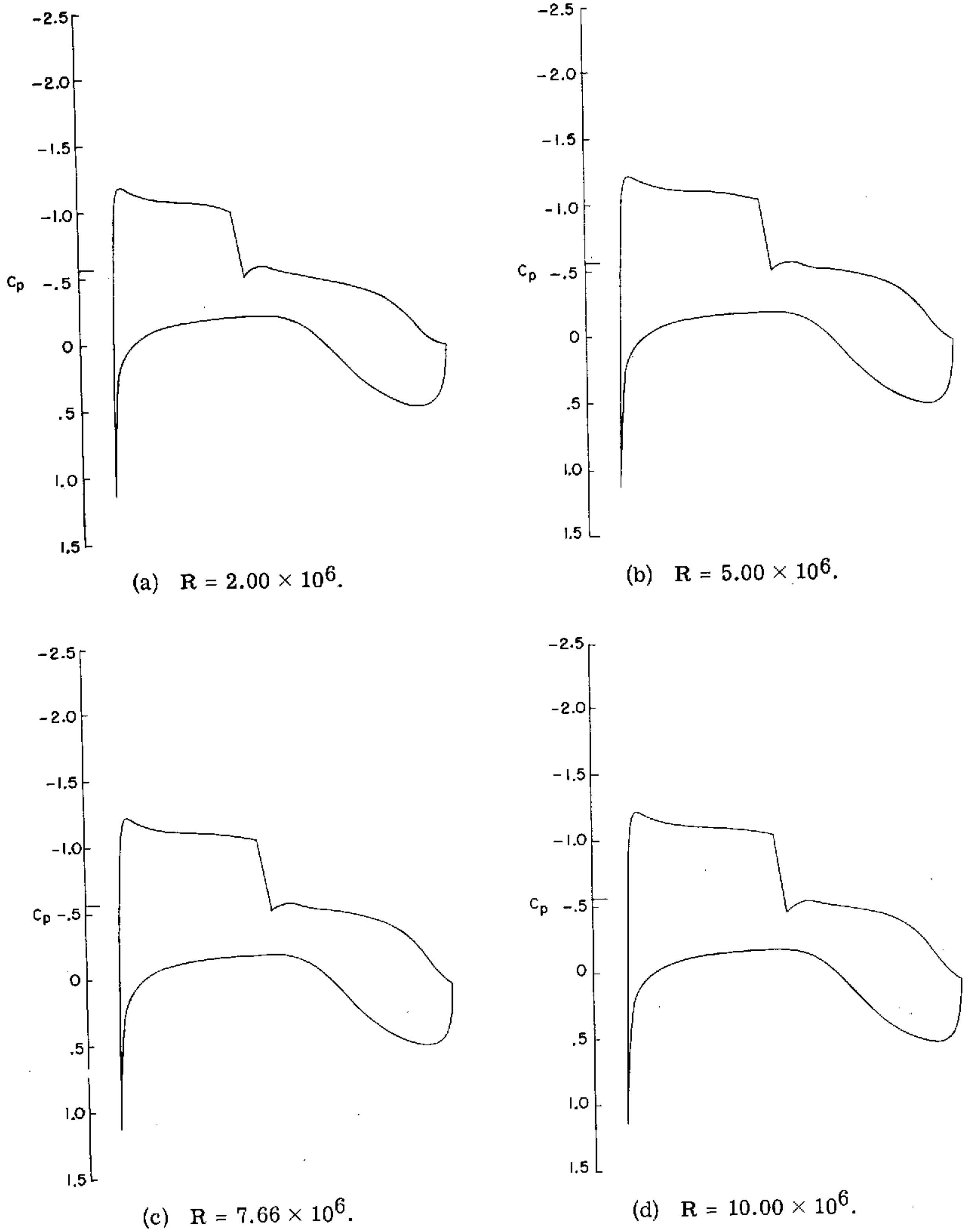
Figure 3.- Typical empirical modifications to aft pressure distribution.



(a) M = 0.500; α = 2.28°.

(b) M = 0.759; α = 0.95°.

Figure 4.- Effect of boundary layer on pressure distributions for a typical supercritical airfoil.



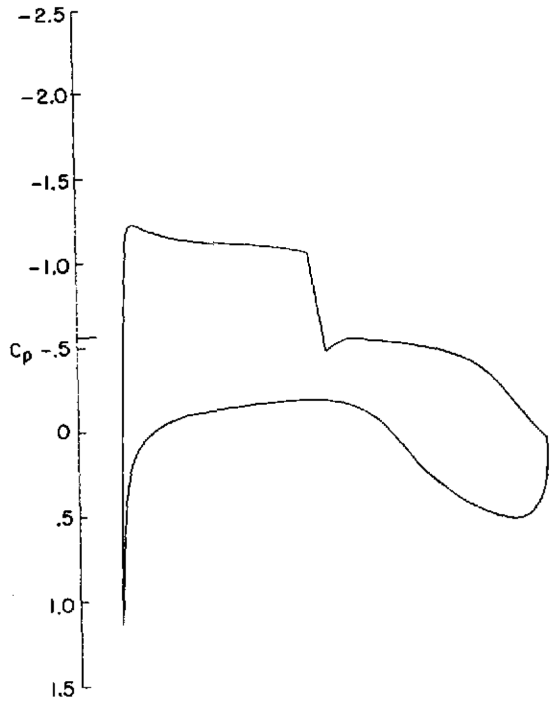
(a) R = 2.00 × 10^6.

(b) R = 5.00 × 10^6.

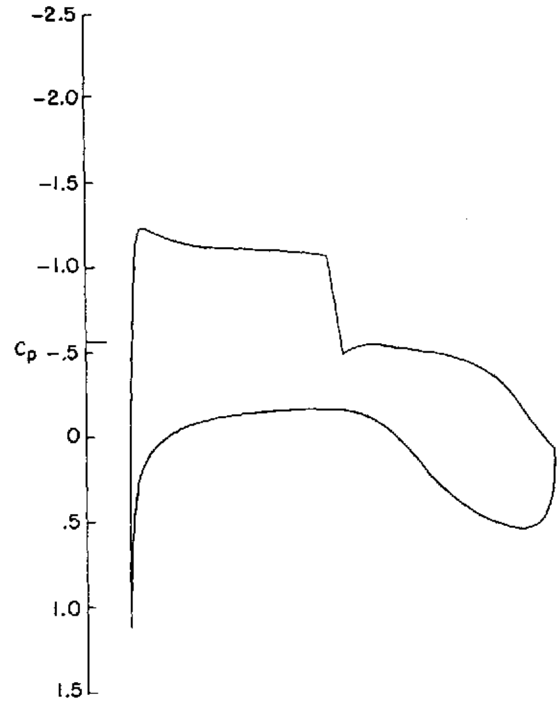
(c) R = 7.66 × 10^6.

(d) R = 10.00 × 10^6.

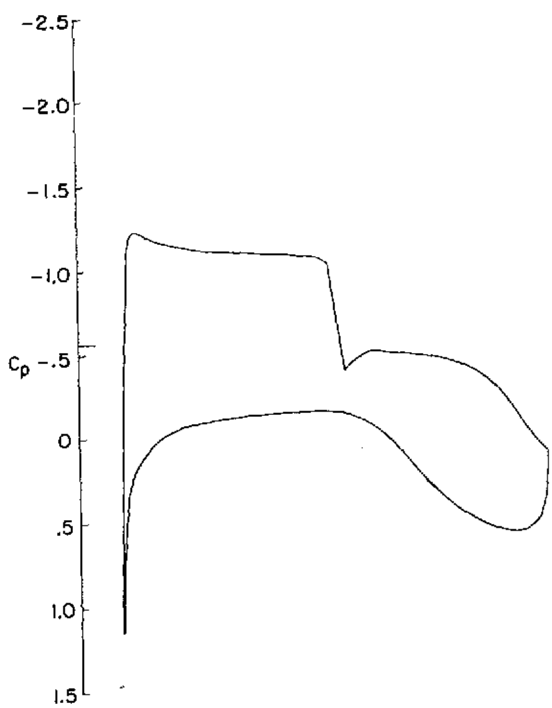
Figure 5.- Pressure distributions for a typical supercritical airfoil. M = 0.759; $\alpha = 0.95^{\circ}$.



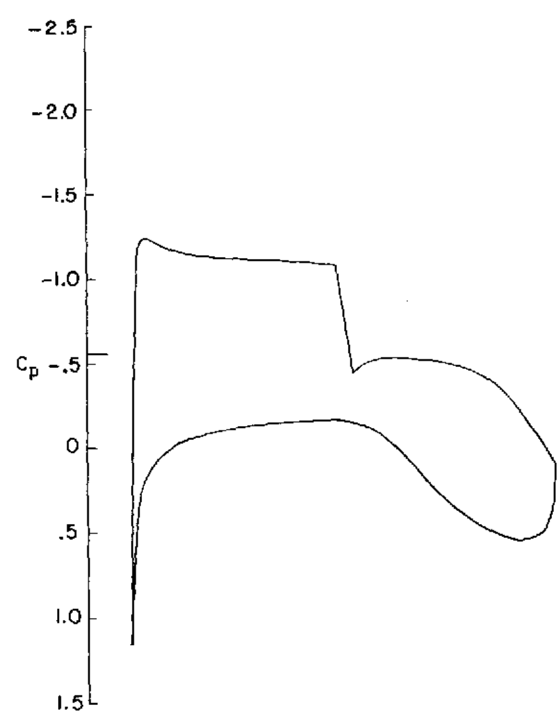
(e) R = 25.00 × 10^6.



(f) R = 50.00 × 10^6.

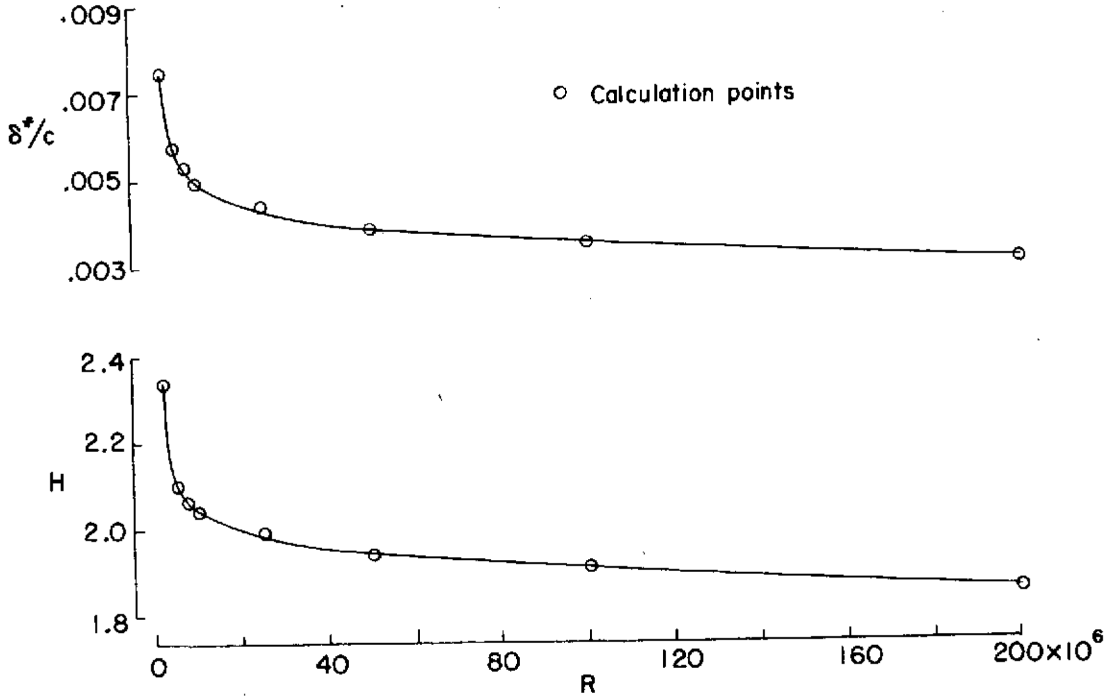


(g) R = 100.00 × 10^6.

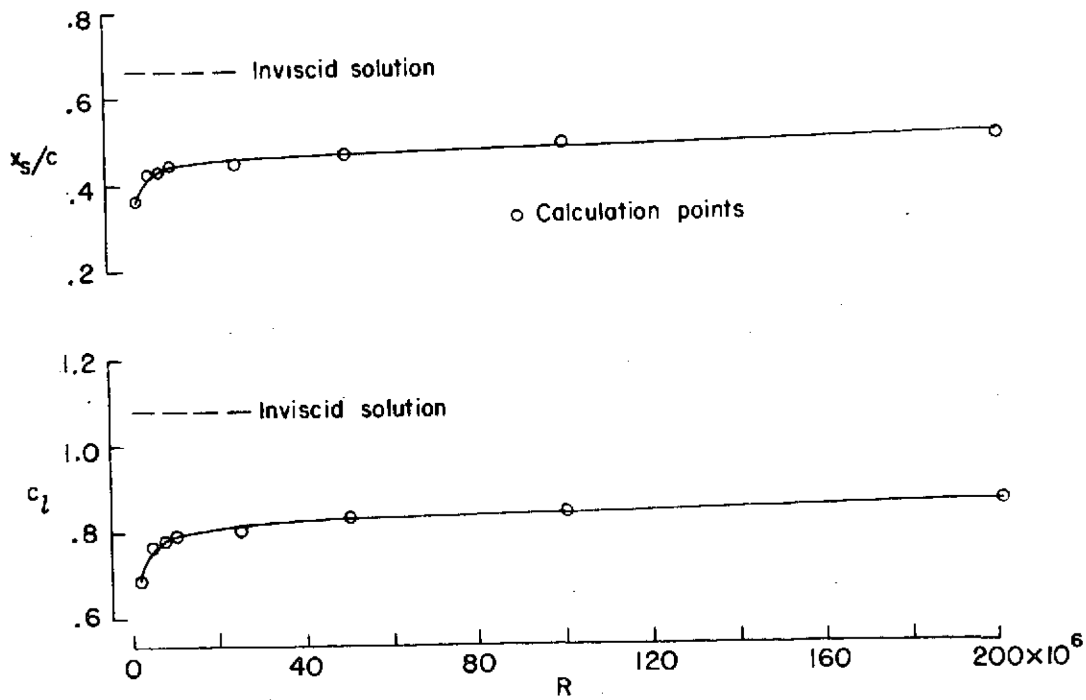


(h) R = 200.00 × 10^6.

Figure 5.- Concluded.

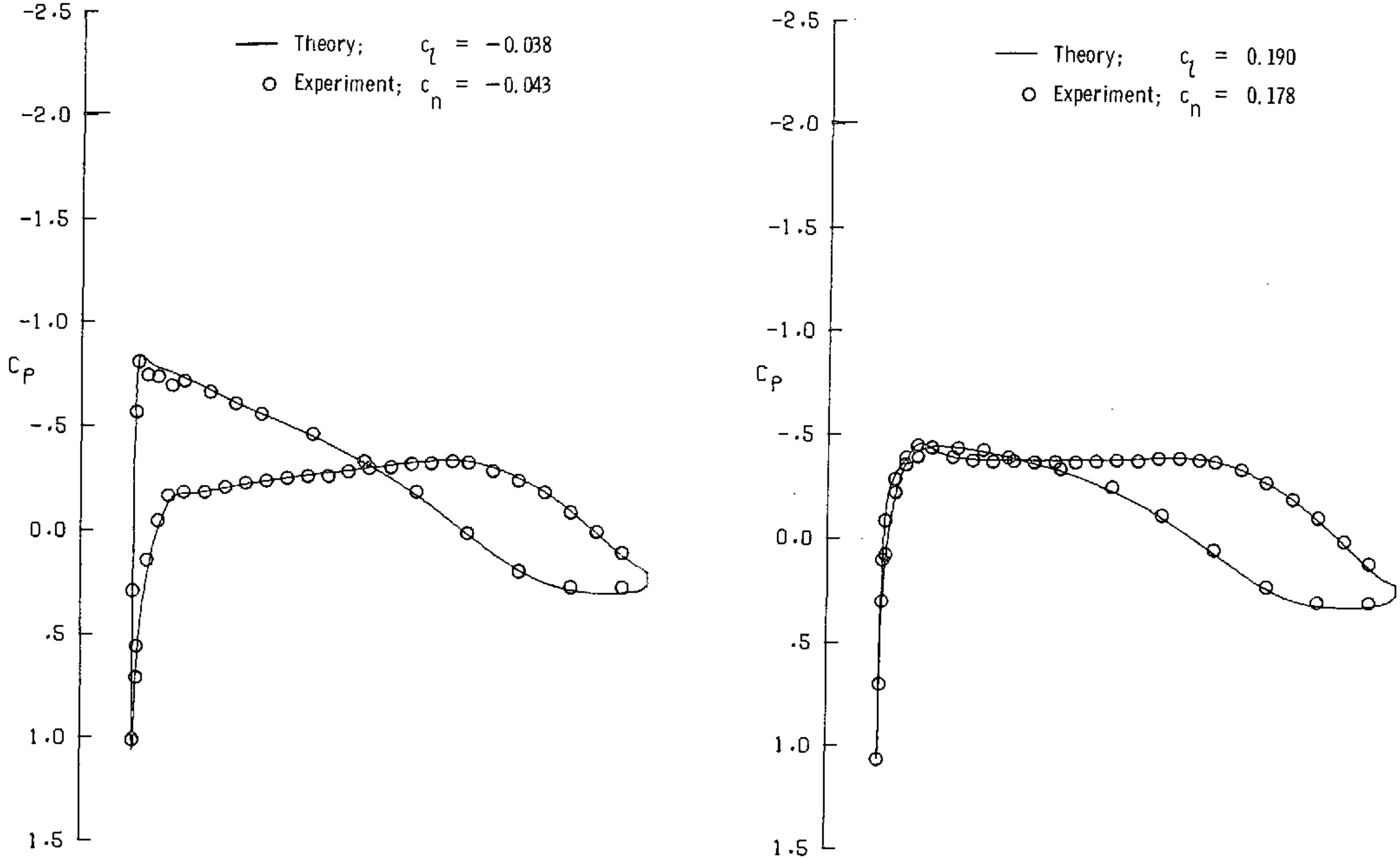


(a) Boundary-layer characteristics; upper surface; $x/c = 0.95$.



(b) External flow characteristics.

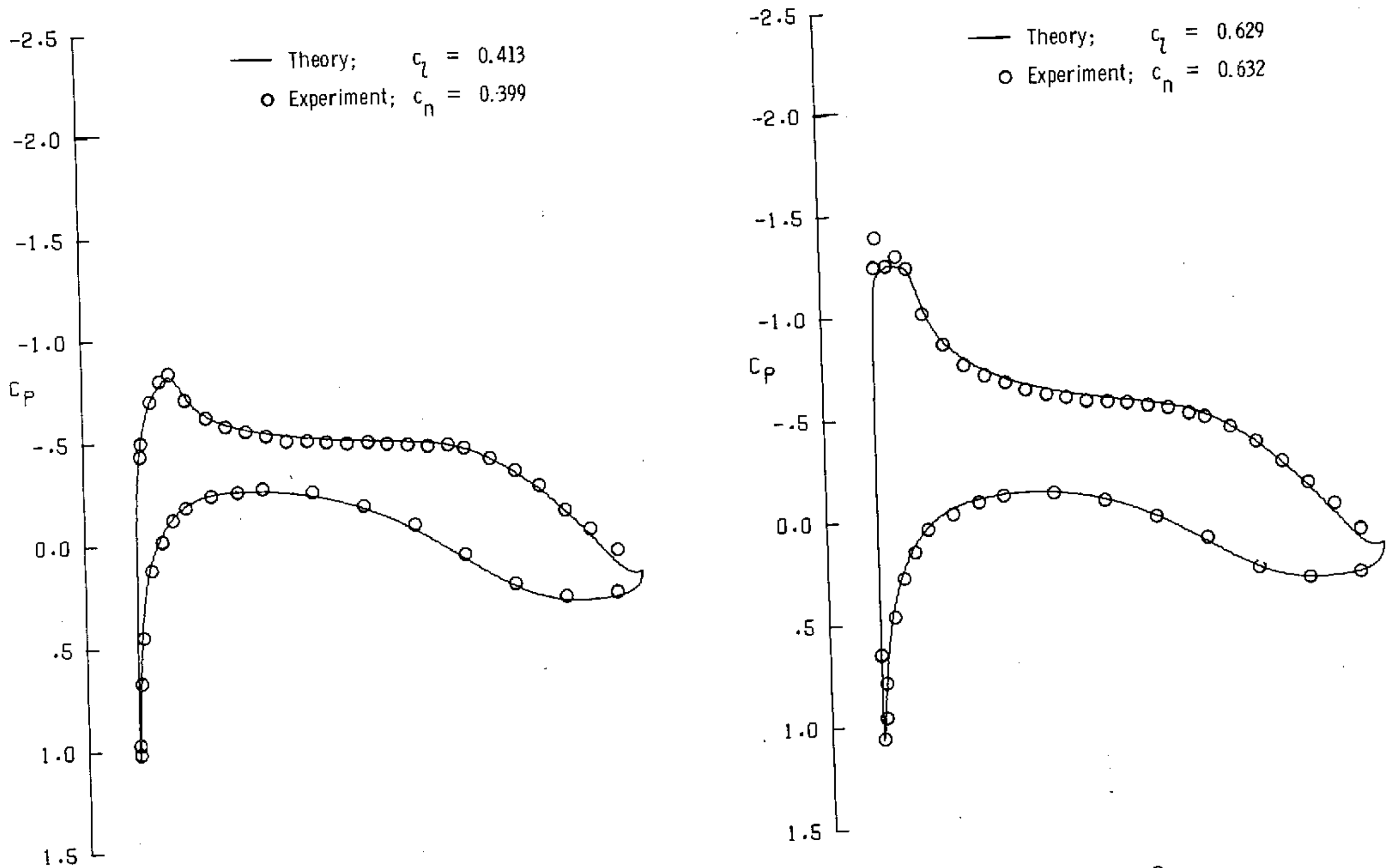
Figure 6.- Variation of selected quantities with Reynolds number. $M = 0.759$; $\alpha = 0.95^{\circ}$.



(a) α = -2.75°.

(b) α = -1.00°.

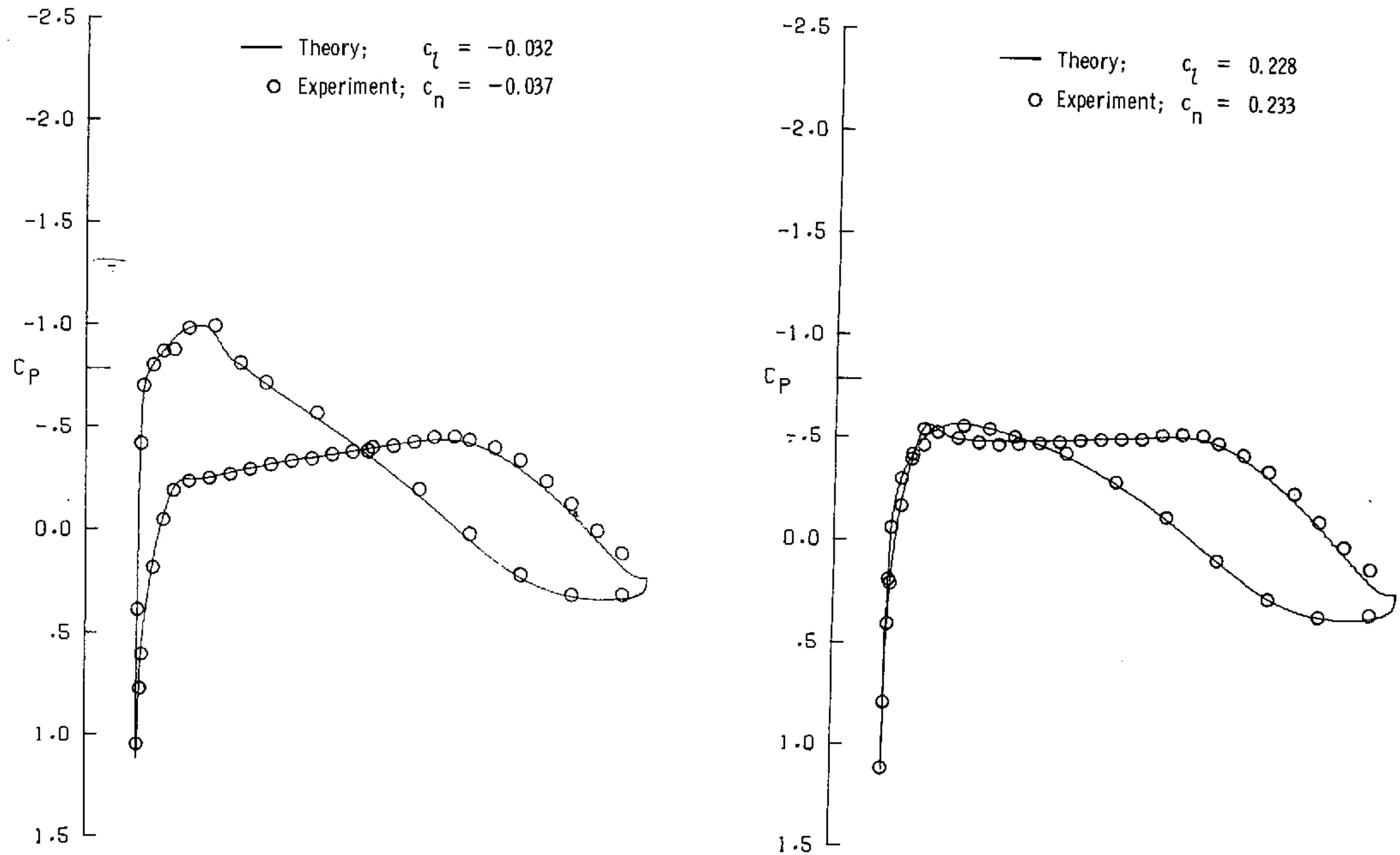
Figure 7.- Pressure distributions for a Korn supercritical airfoil. M = 0.512; R = 21.50 × 10^6.



(c) $\alpha = 0.75^{\circ}$.

(d) $\alpha = 2.40^{\circ}$.

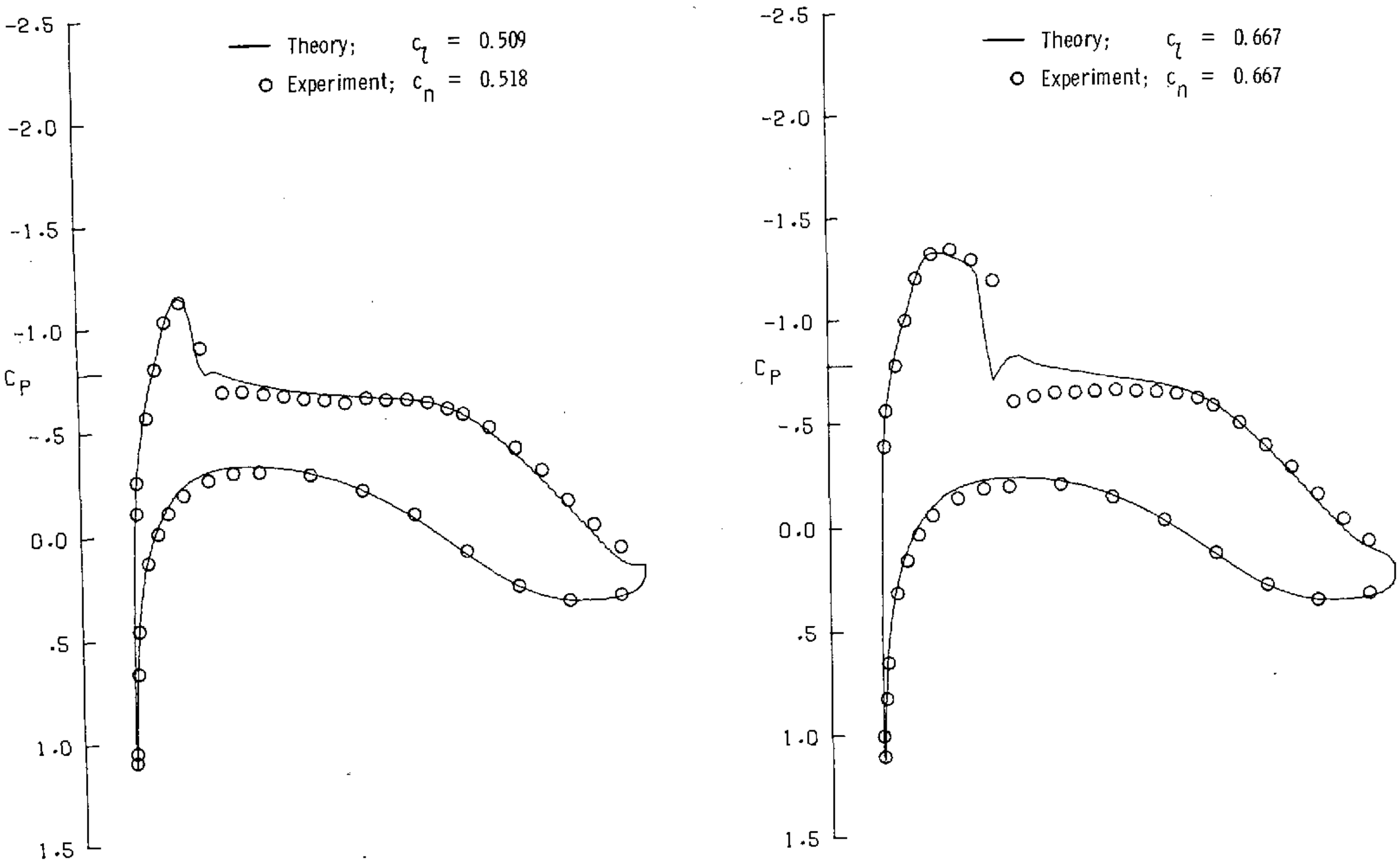
Figure 7.- Concluded.



(a) α = -2.40°.

(b) α = -0.90°.

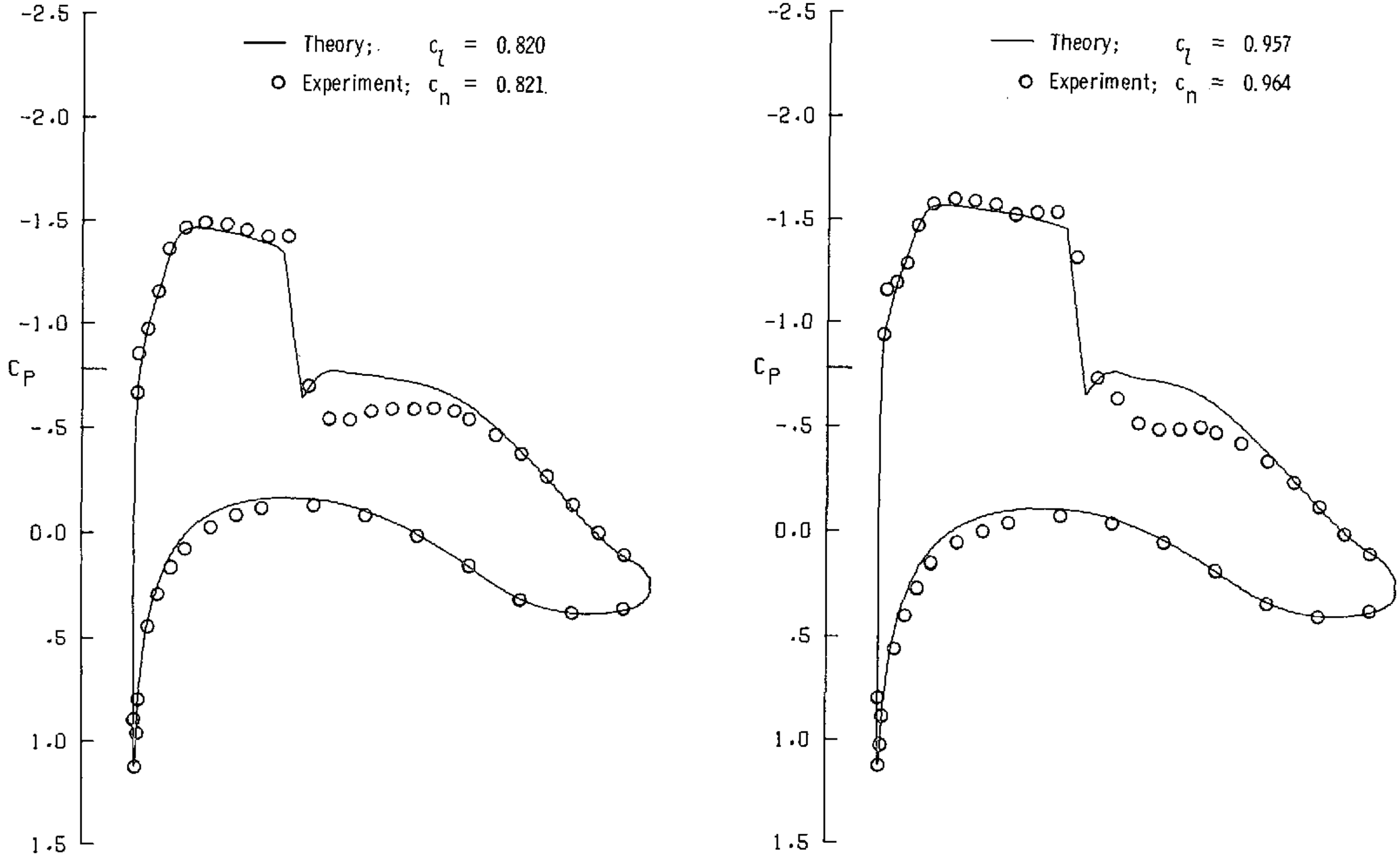
Figure 8.- Pressure distributions for a Korn supercritical airfoil. M = 0.700; R = 21.15×10^6.



(c) $\alpha = 0.70^{\circ}$.

(d) $\alpha = 1.50^{\circ}$.

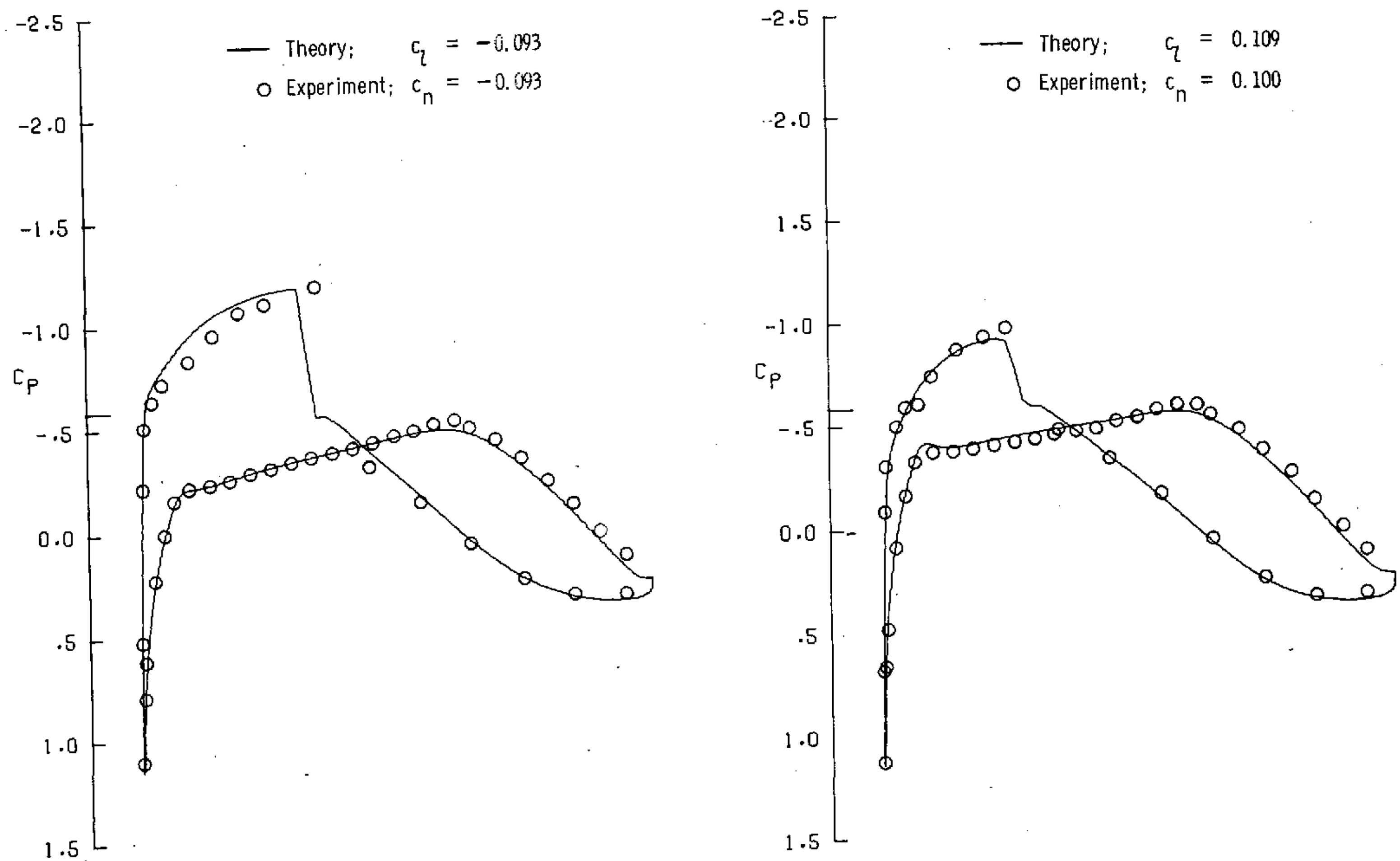
Figure 8.- Continued.



(e) $\alpha = 2.25^{\circ}$.

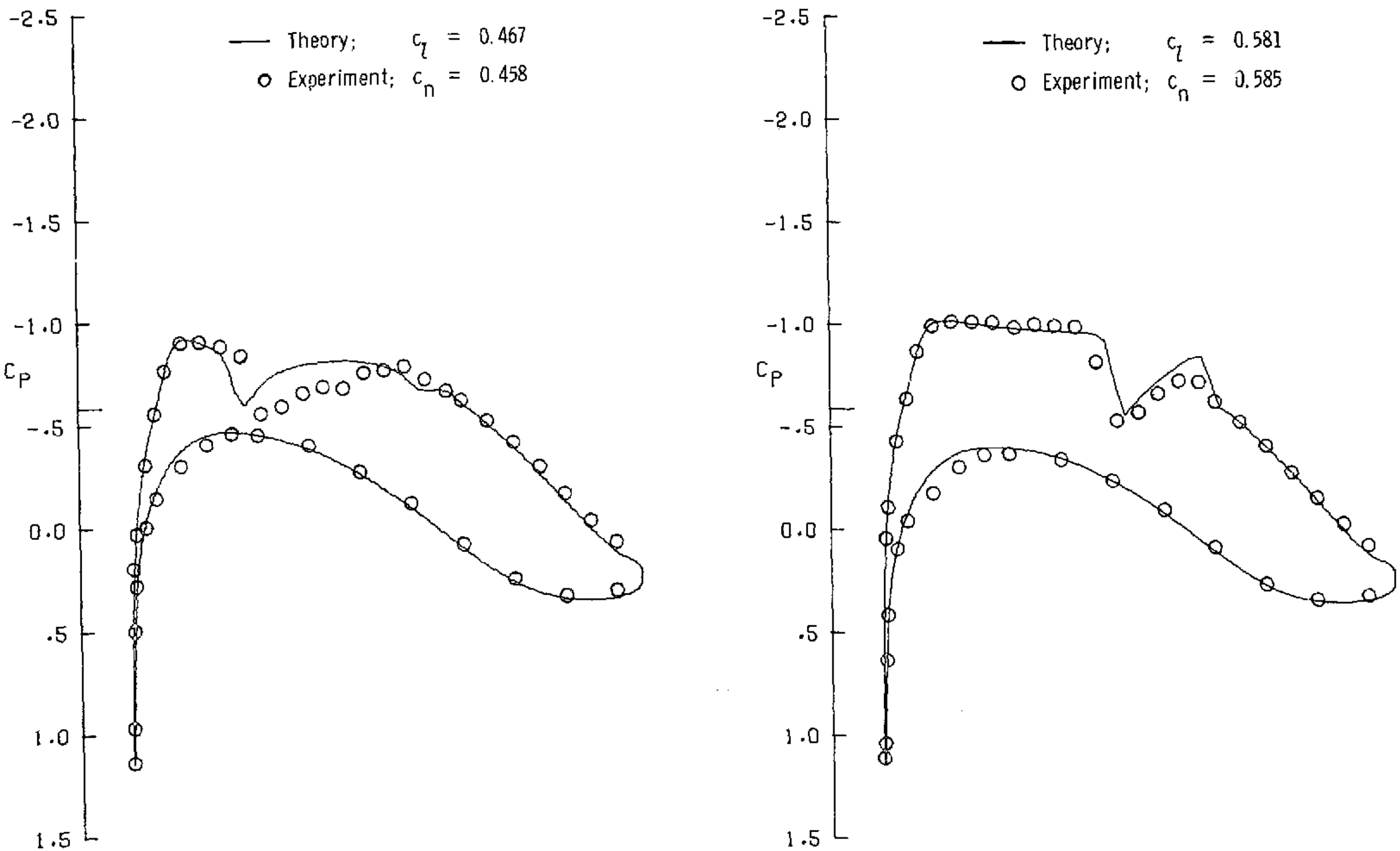
(f) $\alpha = 2.90^{\circ}$.

Figure 8.- Concluded.



(a) α = -2.60°. (b) α = -1.60°.

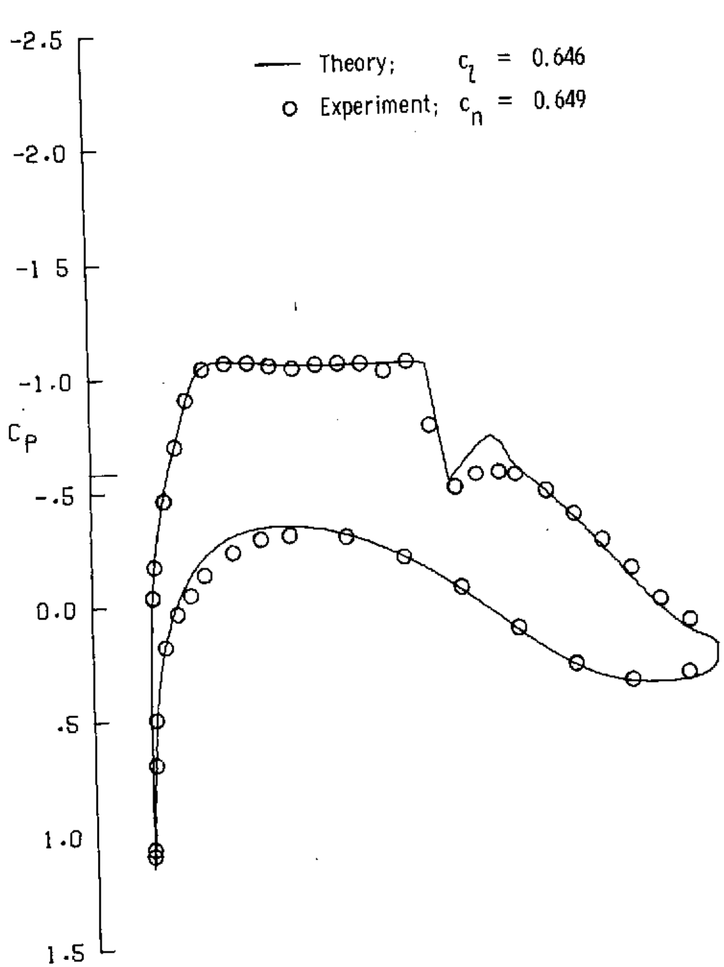
Figure 9.- Pressure distributions for a Korn supercritical airfoil. M = 0.752; R = 20.95 × 10^6.



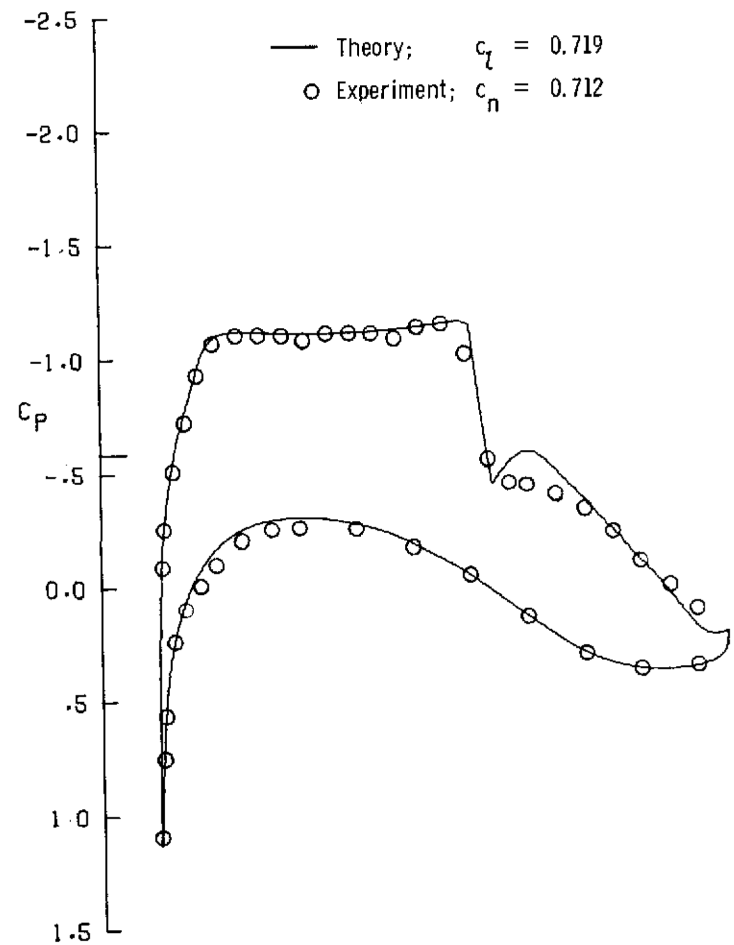
(c) $\alpha = 0.00^{\circ}$.

(d) $\alpha = 0.45^{\circ}$.

Figure 9.- Continued.

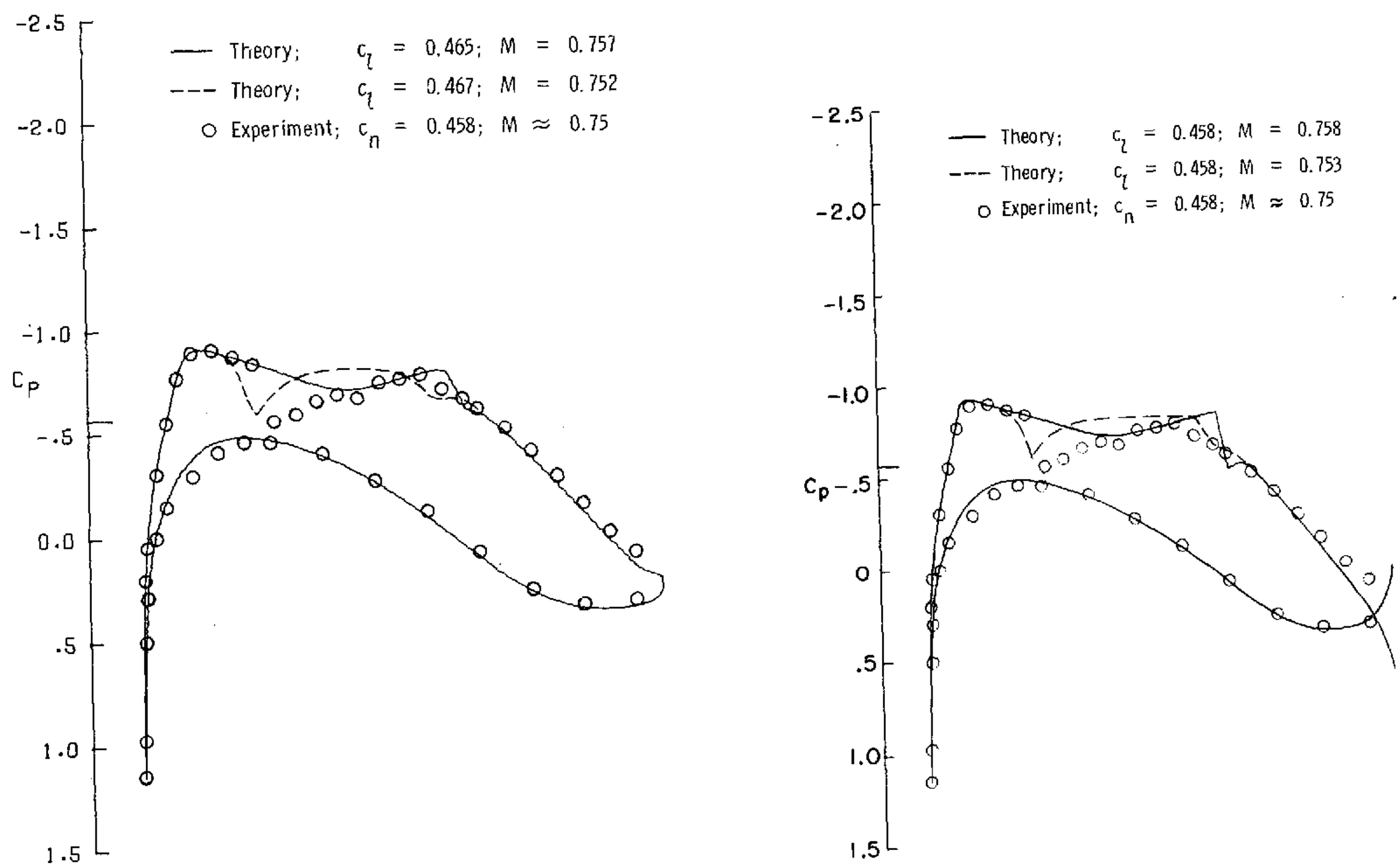


(e) $\alpha = 0.90^{\circ}$.



(f) $\alpha = 1.25^{\circ}$.

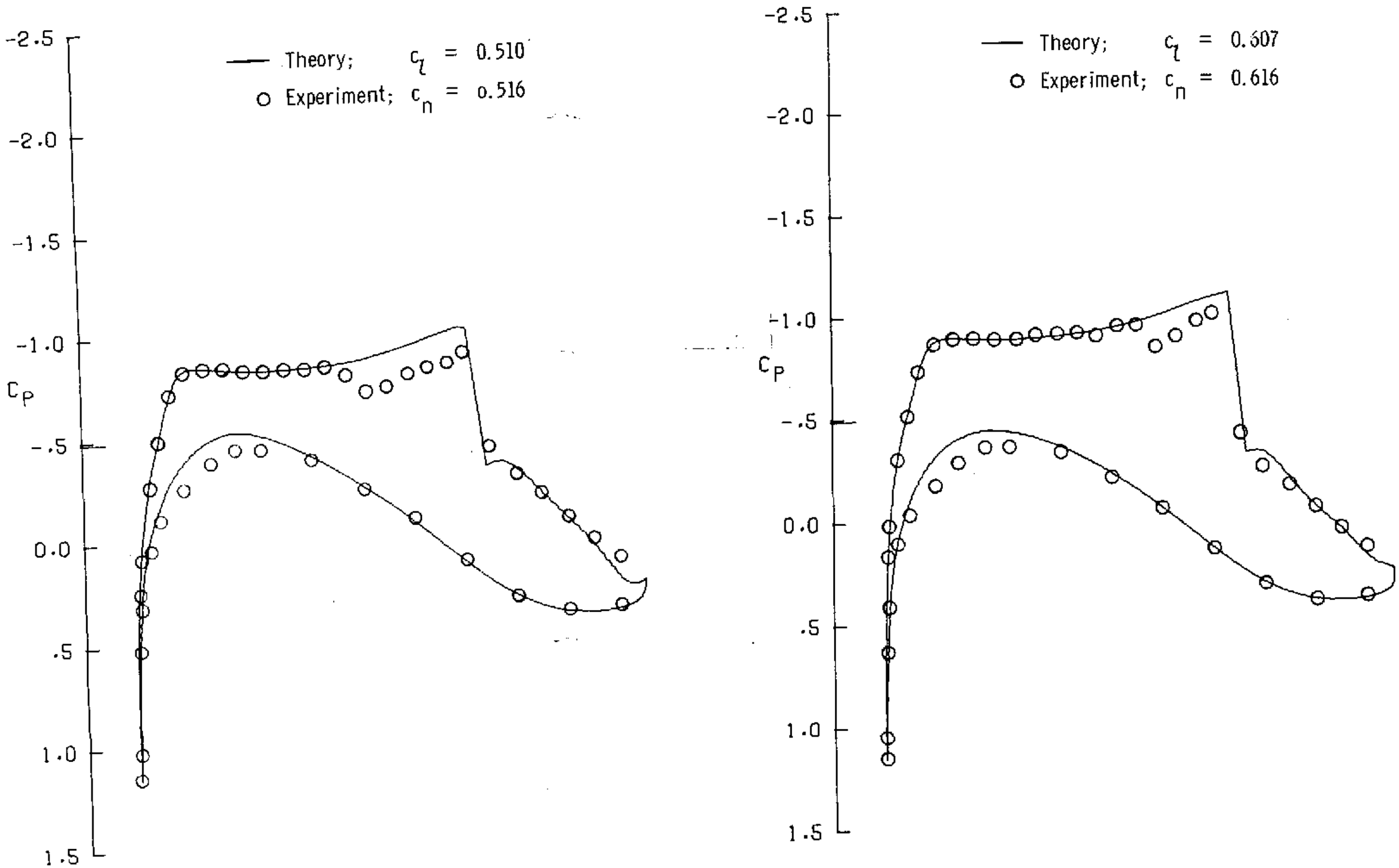
Figure 9.- Concluded.



(a) Present analysis method.

(b) Analysis method of reference 13.

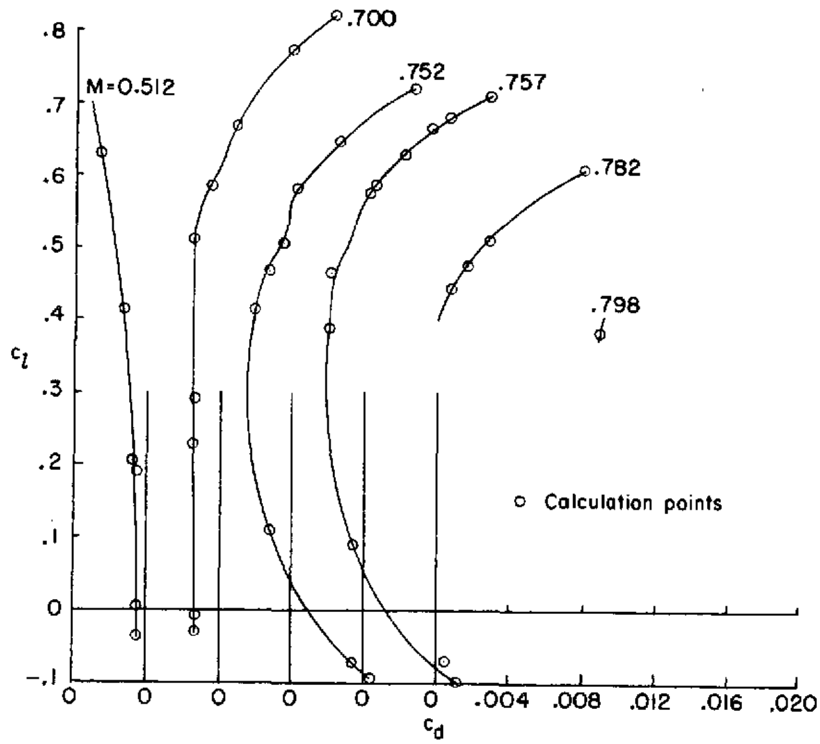
Figure 10.- Effect of Mach number on pressure distributions for a Korn supercritical airfoil.



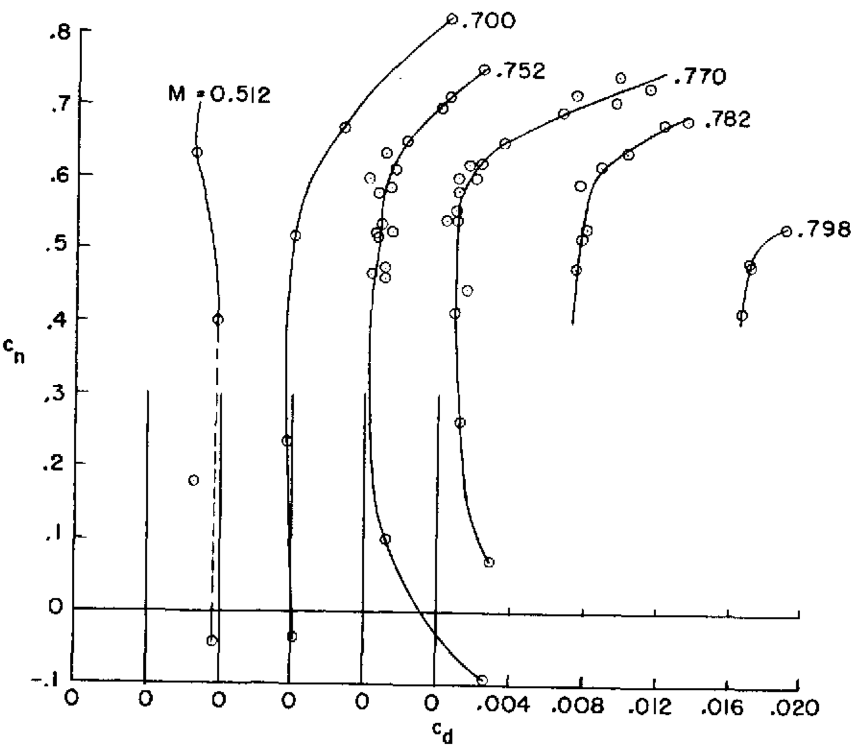
(a) $\alpha = 0.10^{\circ}$.

(b) $\alpha = 0.40^{\circ}$.

Figure 11.- Pressure distributions for a Korn supercritical airfoil. $M = 0.782$; $R = 20.60 \times 10^{6}$.



(a) Theory.



(b) Experiment.

Figure 12.- Drag polars for a Korn supercritical airfoil.

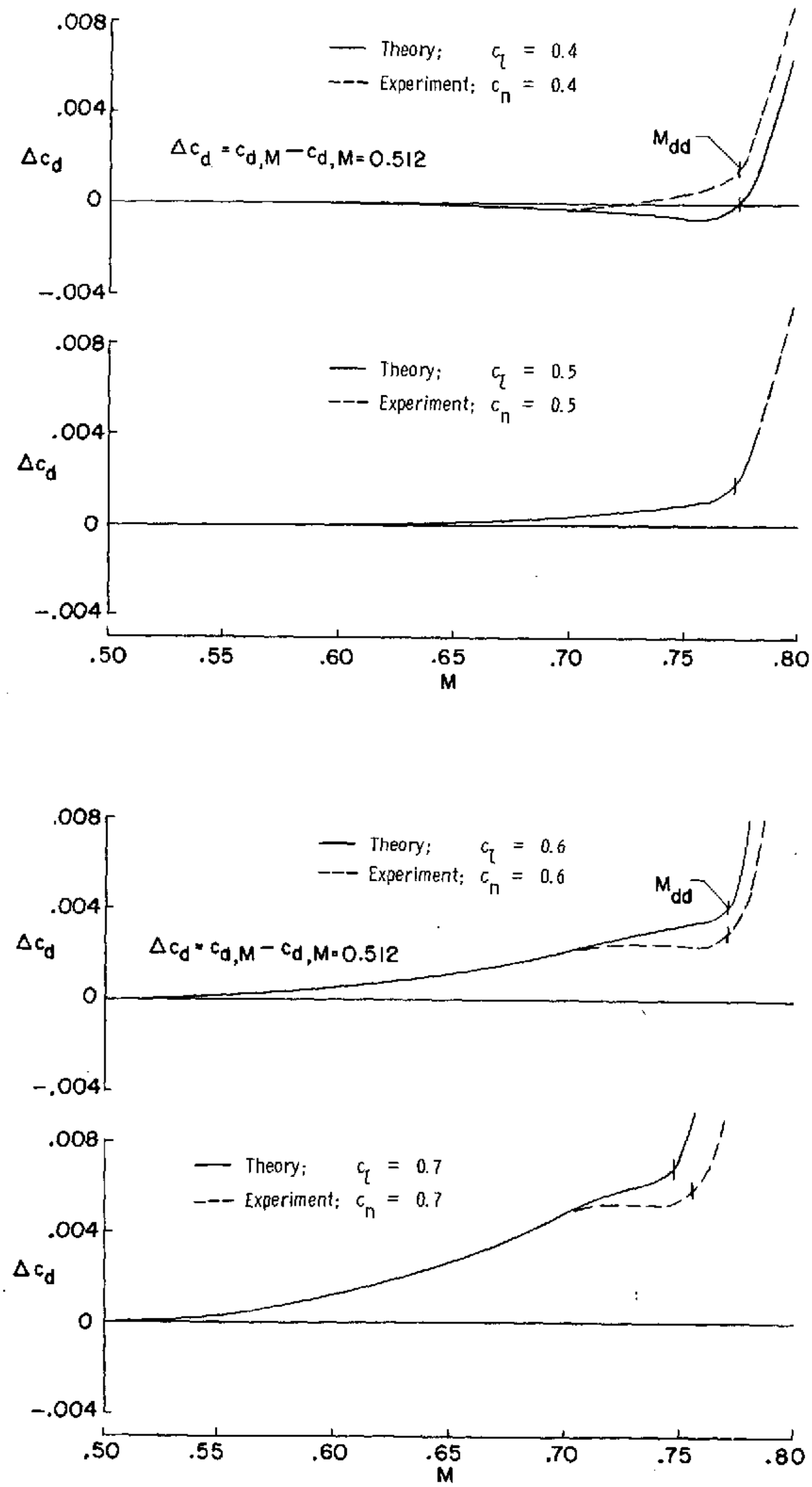


Figure 13.- Drag creep for a Korn supercritical airfoil.

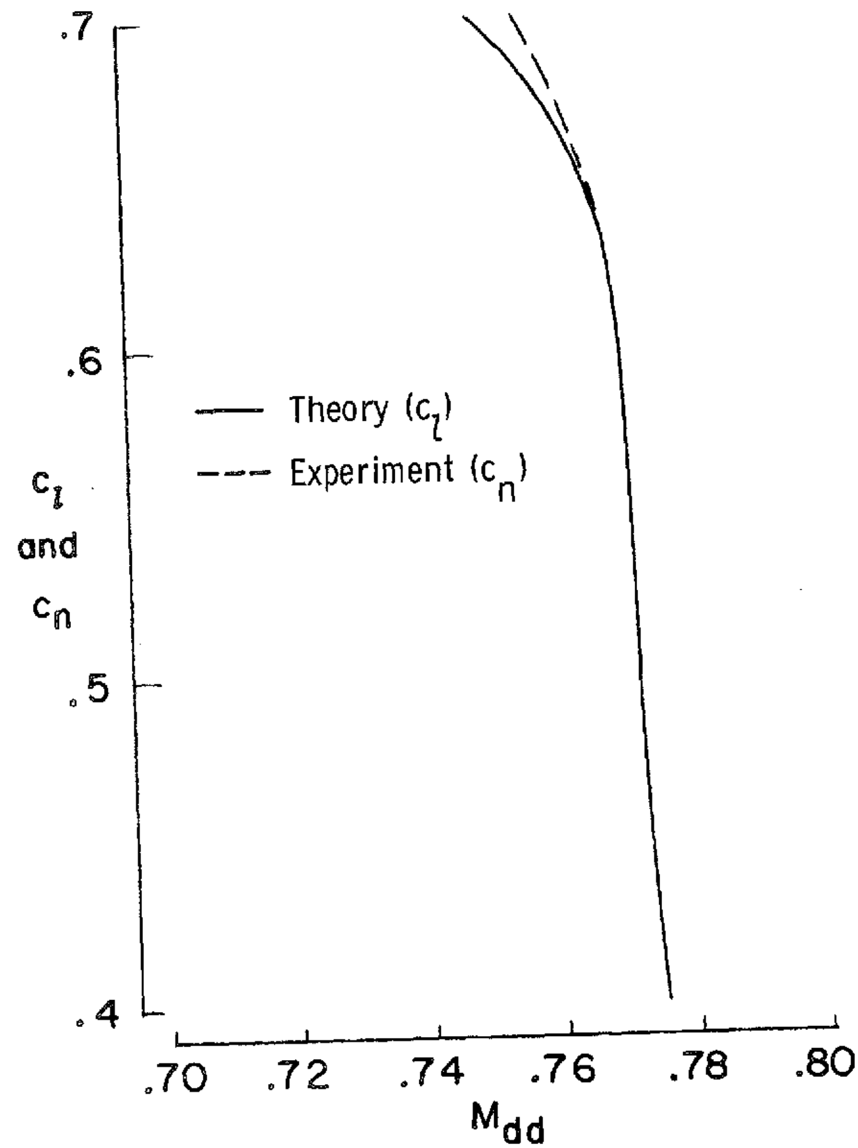
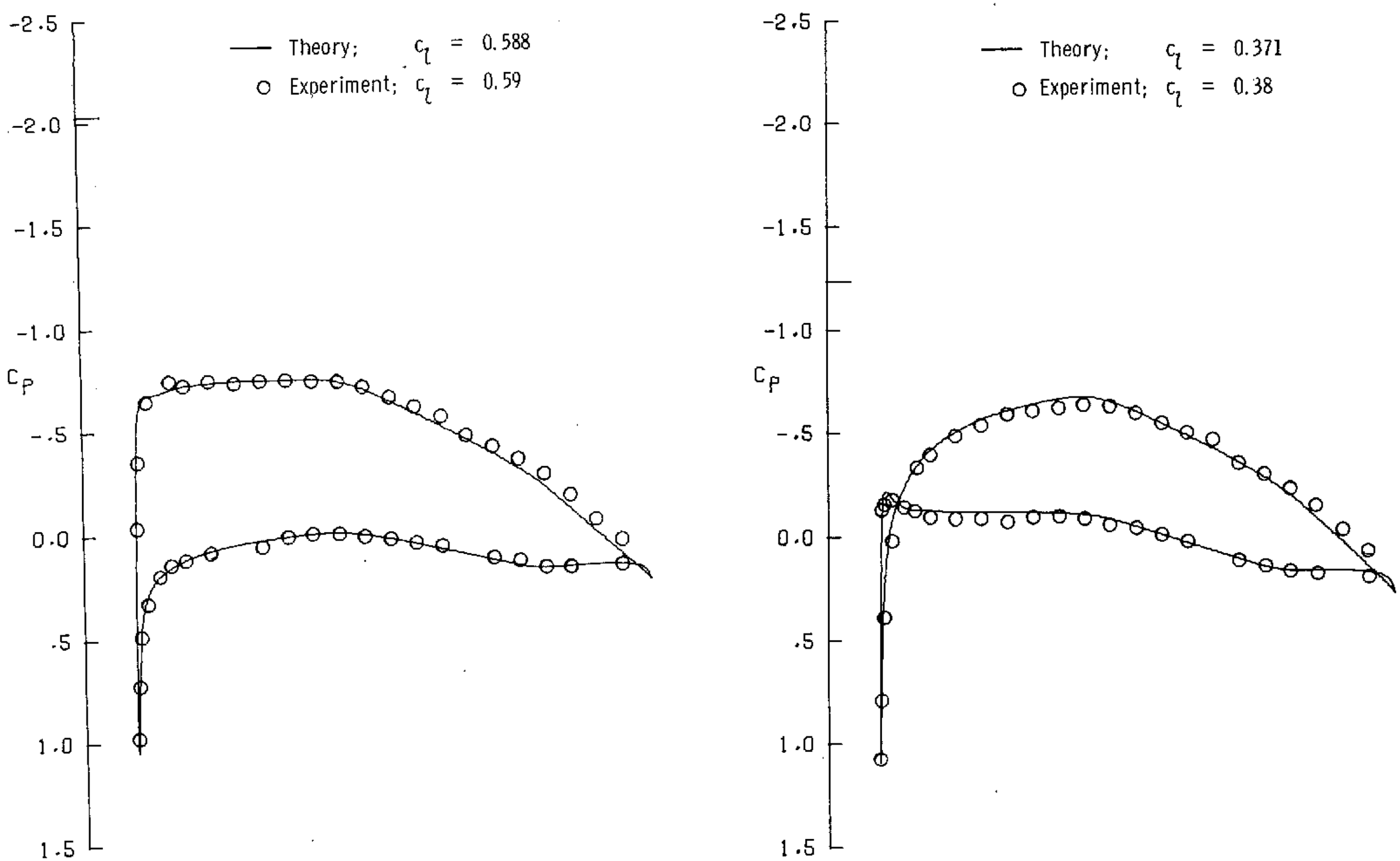


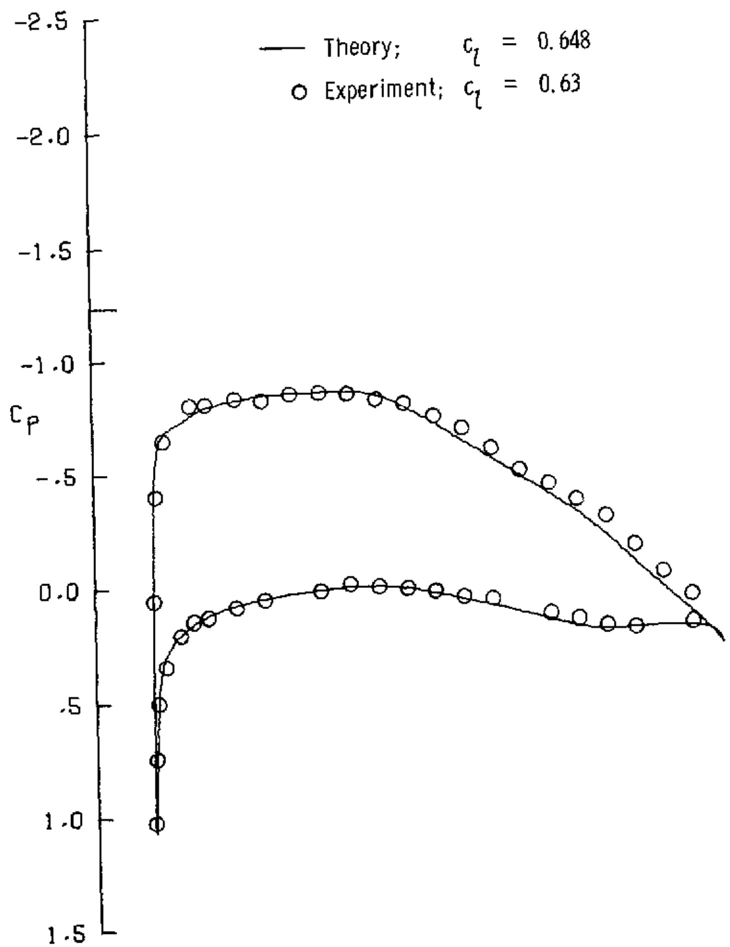
Figure 14.- Drag divergence boundary for a Korn supercritical airfoil.



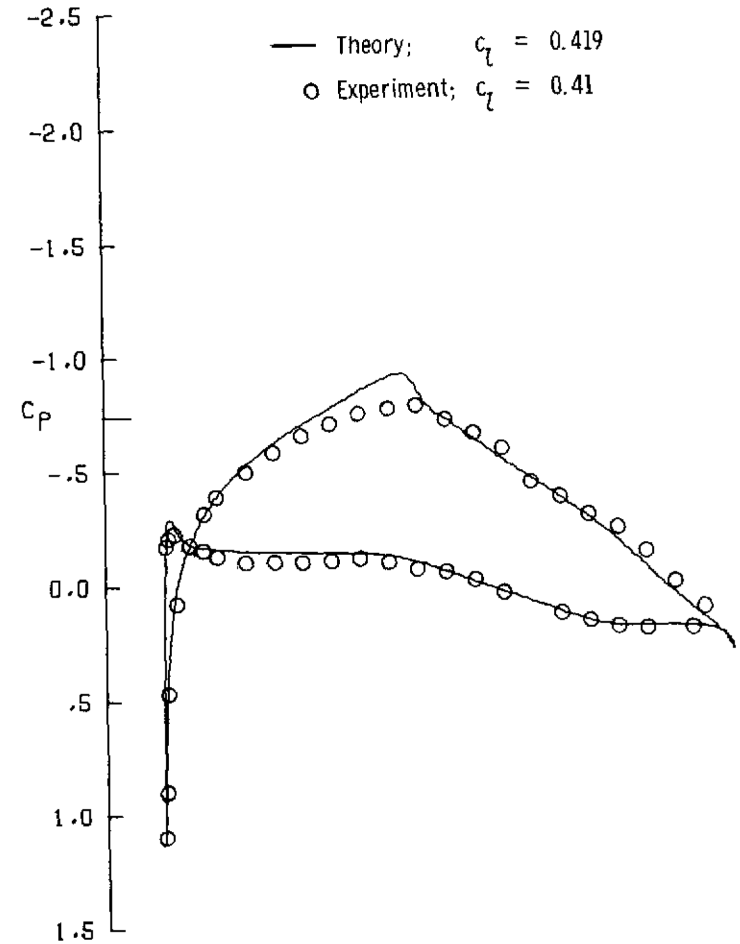
(a) M = 0.51; R = 1.45 × 10^6; α = 2.00°.

(b) M = 0.61; R = 1.65 × 10^6; α = 0.00°.

Figure 15.- Pressure distributions for a NACA 64A410 airfoil.

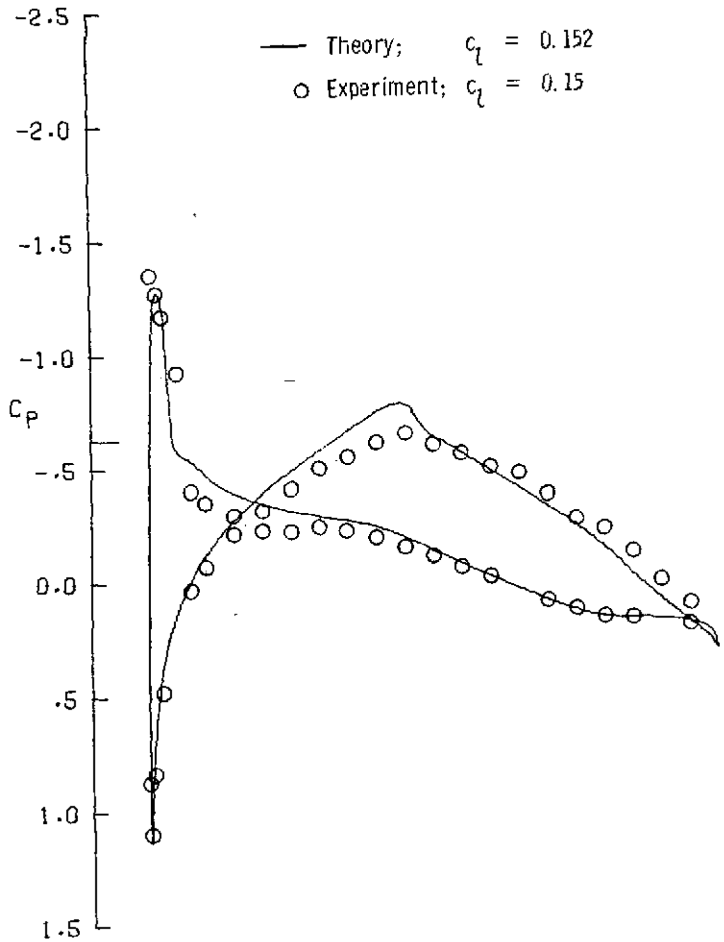


(c) M = 0.61; R = 1.65 × 10^6; α = 2.00°.

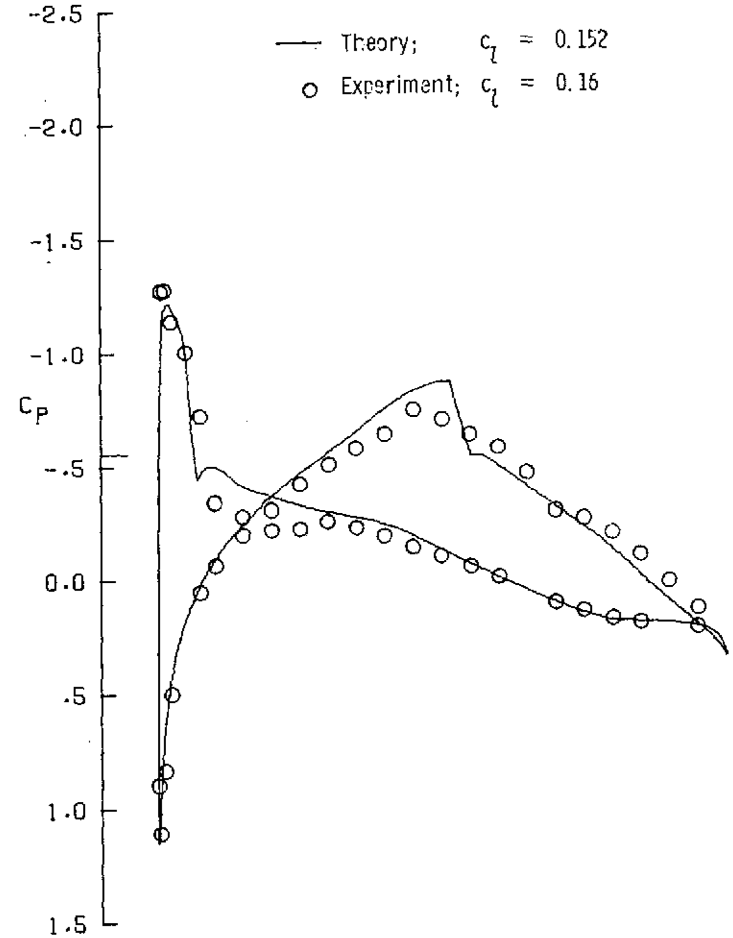


(d) M = 0.71; R = 1.73 × 10^6; α = -0.10°.

Figure 15.- Continued.



(e) M = 0.74; R = 1.77 × 10^6; α = -1.60°.



(f) M = 0.76; R = 1.78 × 10^6; α = -1.70°.

Figure 15.- Concluded.

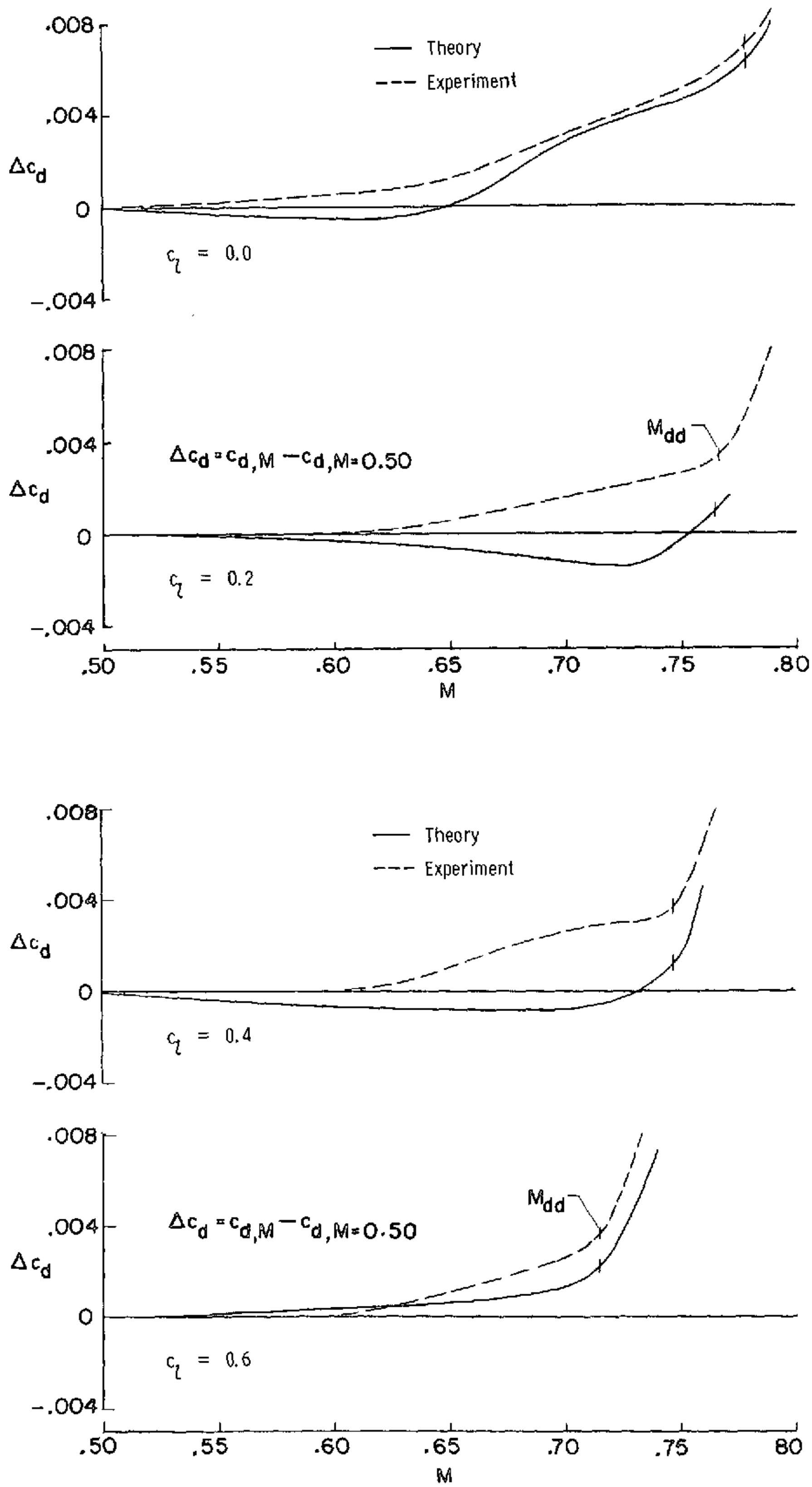


Figure 16.- Drag creep for a NACA 64A410 airfoil.

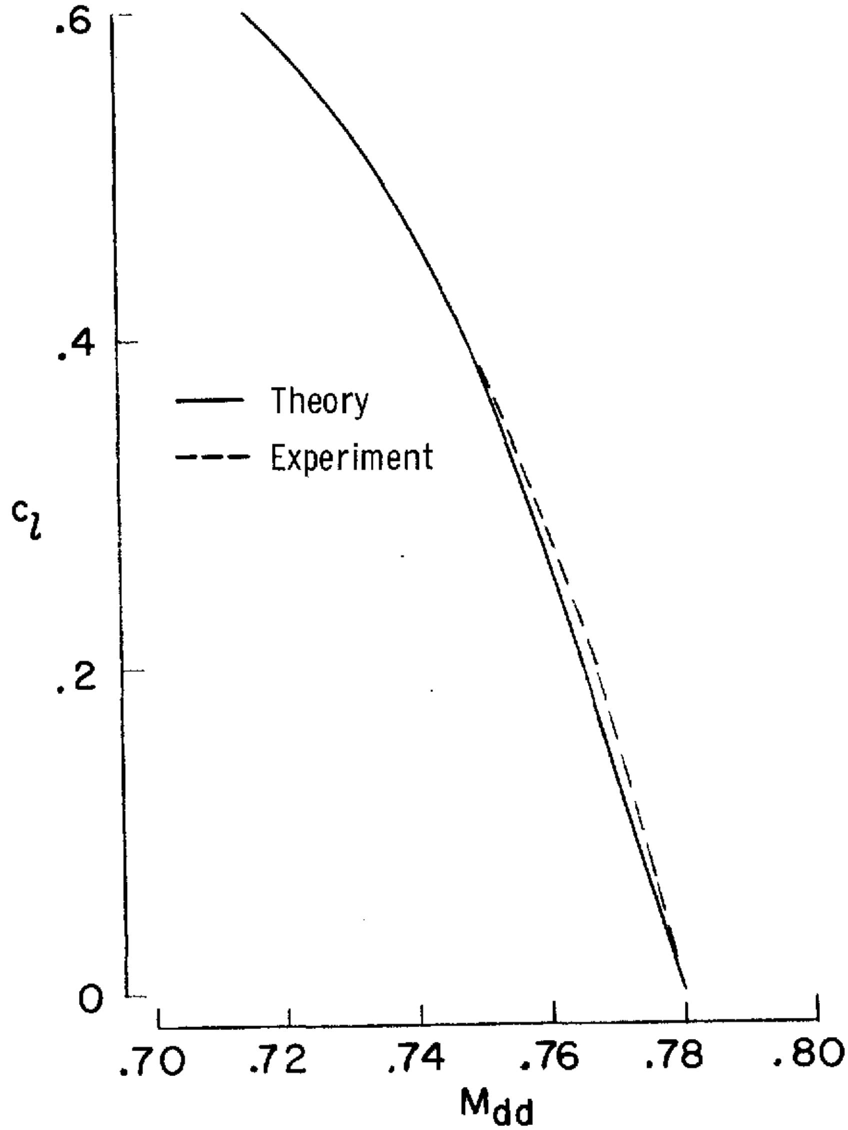
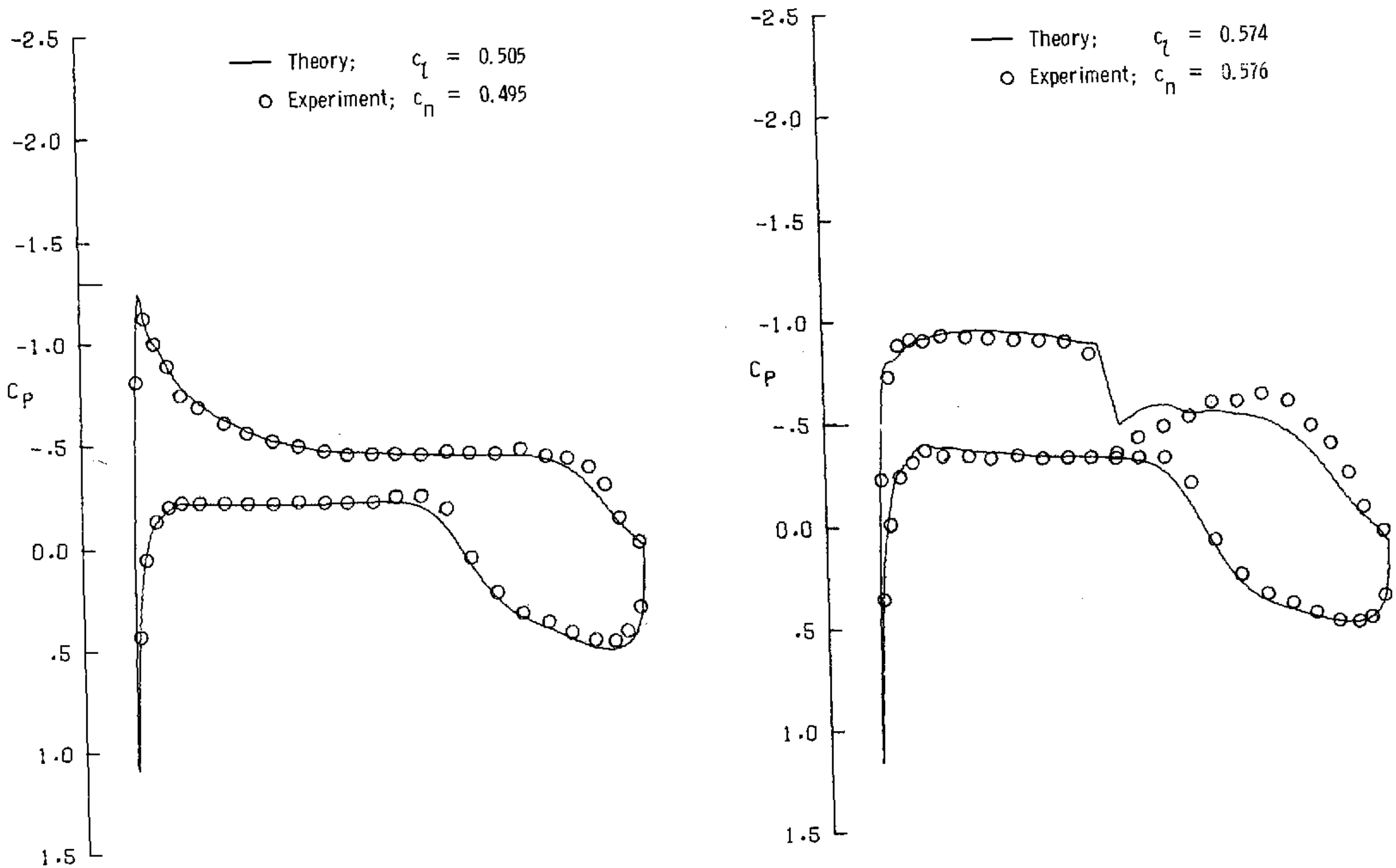


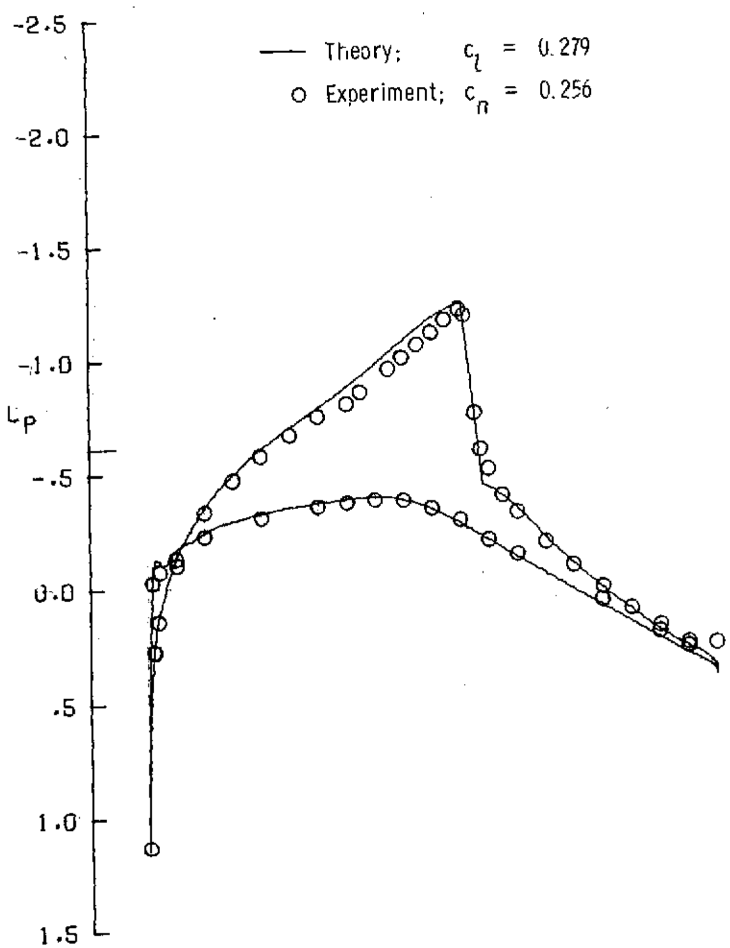
Figure 17.- Drag divergence boundary for a NACA 64A410 airfoil.



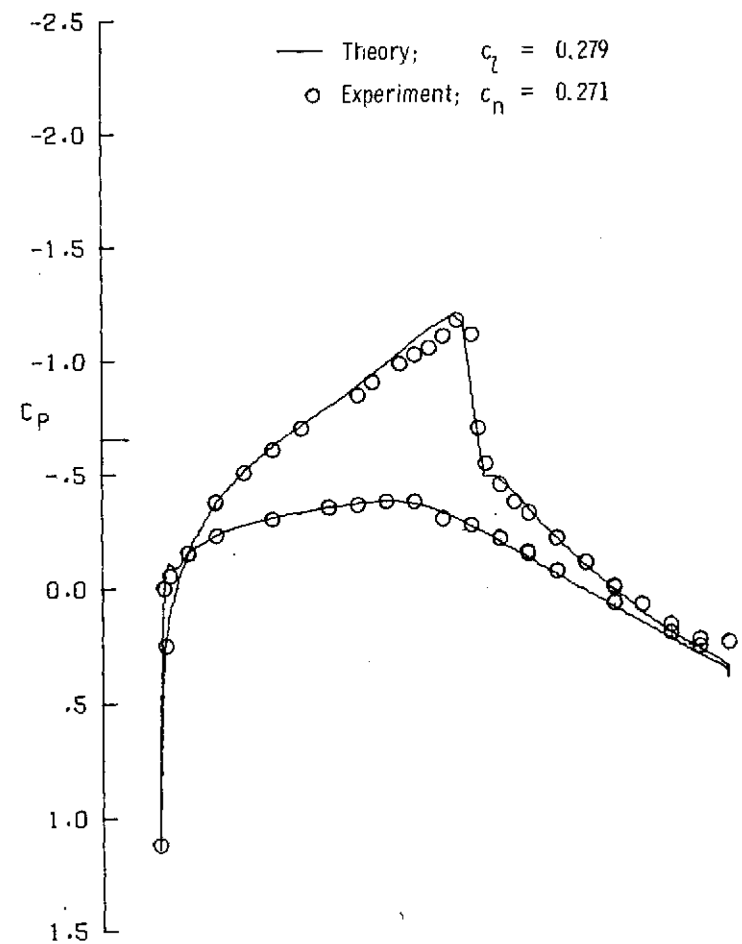
(a) M = 0.600; R = 6.60 × 10^6; α = 0.10°.

(b) M = 0.780; R = 7.76 × 10^6; α = 0.00°.

Figure 18.- Pressure distributions for an early NASA supercritical airfoil.

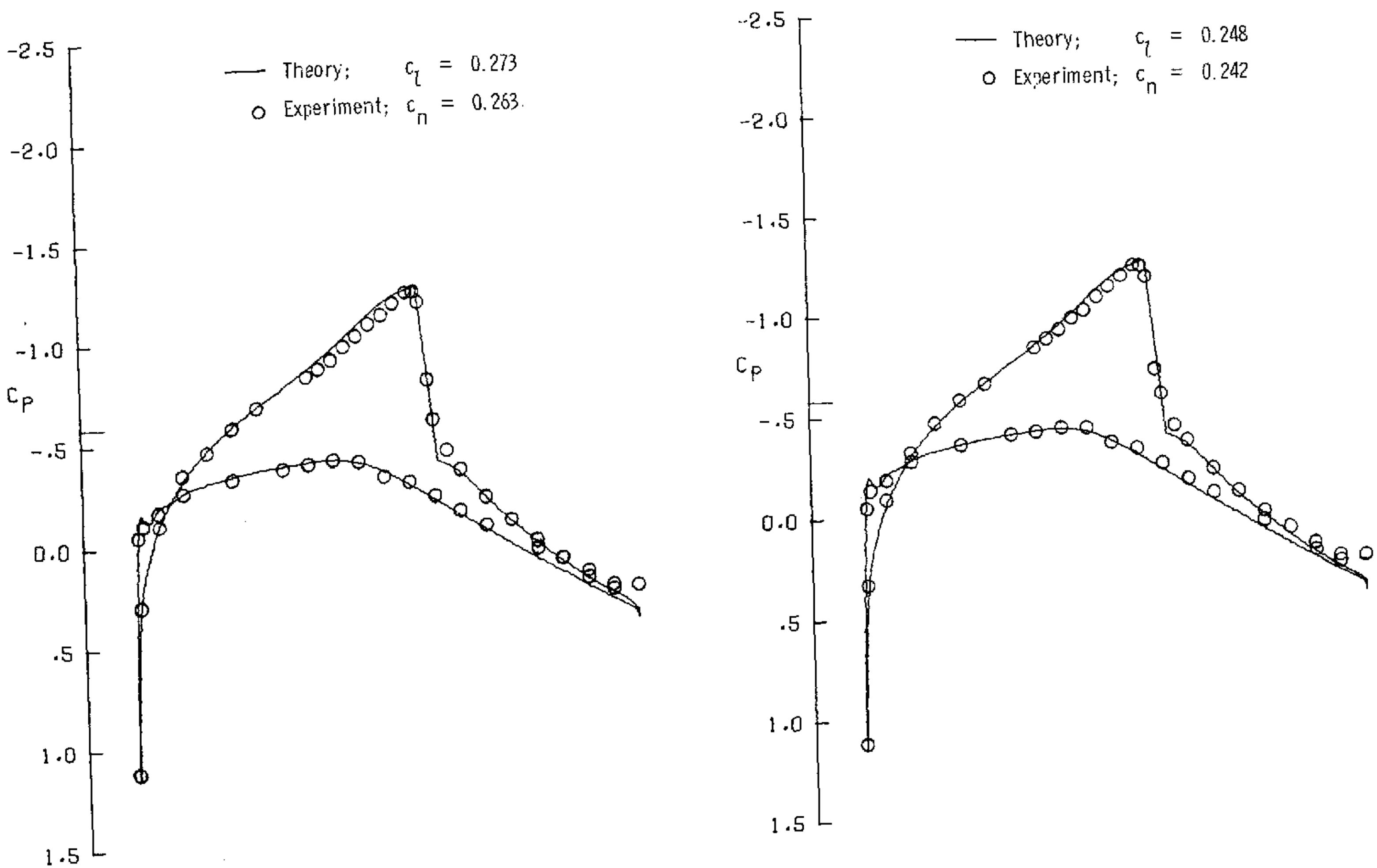


(a) M = 0.744; R = 25.00 × 10^6; α = 0.10°.



(b) M = 0.732; R = 34.10 × 10^6; α = 0.10°.

Figure 19.- Pressure distributions for a NACA 65_1-213 (a = 0.5) airfoil.



(c) M = 0.752; R = 43.90 × 10^6; α = 0.00°.

(d) M = 0.753; R = 52.60 × 10^6; α = -0.10°.

Figure 19.- Concluded.

NASA-Langley, 1975 L-9708

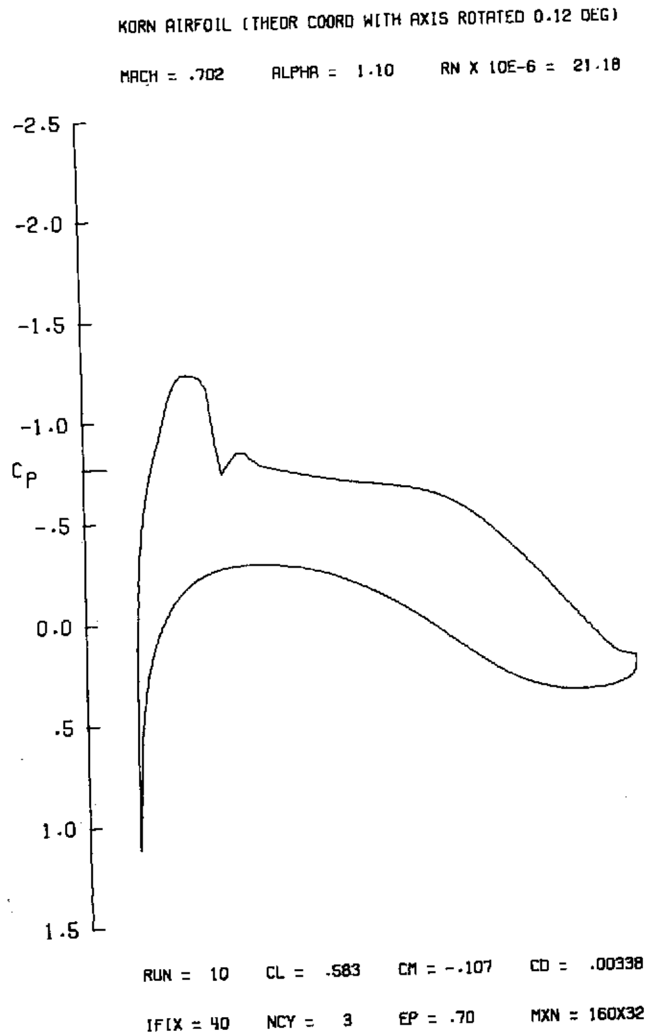


Figure 20.- Sample of CalComp plot generated by computer program.

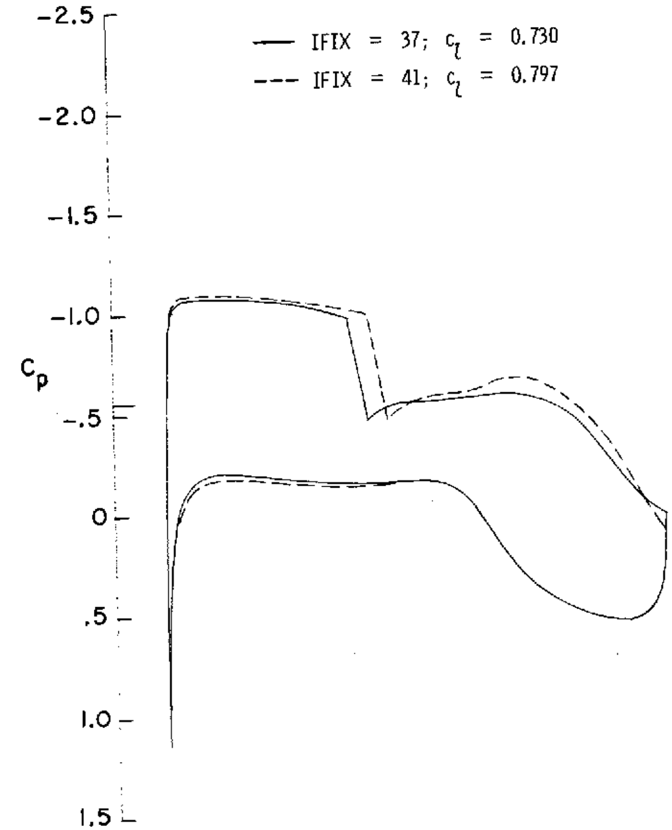


Figure 21.- Effect of IFIX on pressure distribution for a typical supercritical airfoil. M = 0.760; $R = 7.66 \times 10^6$; $\alpha = 0.00^{\circ}$.