

NASA
CR
3064
c.1

NASA Contractor Report 3064

TECH LIBRARY KAFB, NM



0061773

LOAN COPY: RETURN
AFWL TECHNICAL LIBR
KIRTLAND AFB, N. M.

Operational Manual for Two-Dimensional Transonic Code TSFOIL

Stephen S. Stahara

CONTRACT NAS2-8648
DECEMBER 1978





NASA Contractor Report 3064

Operational Manual for Two-Dimensional Transonic Code TSFOIL

Stephen S. Stahara
Nielsen Engineering & Research, Inc.
Mountain View, California

Prepared for
Ames Research Center
under Contract NAS2-8648



National Aeronautics
and Space Administration

**Scientific and Technical
Information Office**

1978

TABLE OF CONTENTS

<u>Section</u>	<u>Page No.</u>
1. SUMMARY	1
2. INTRODUCTION	2
3. LIST OF SYMBOLS	5
4. GENERAL USER'S MANUAL	8
4.1 Description of Program	8
4.1.1 Differential equations and transonic scaling	8
4.1.2 Geometry	9
4.1.3 Boundary conditions	9
4.1.4 Aerodynamic variables	13
4.1.5 Numerical formulation	13
4.1.6 Operating procedure	17
4.1.7 Overall program flow chart	19
4.2 Program Input	20
4.2.1 Dictionary of input variables	20
4.2.2 Control options	22
4.2.3 Default values	30
4.2.4 Input data format	31
4.3 Error and Information Messages	34
4.4 Program Output	38
4.4.1 General description of the output	38
4.4.2 Sample cases	41
4.5 Comparisons with Data and Other Methods	42
4.6 Program Run Times	44
4.7 Machine Compatability	46
5. PROGRAMMER'S MANUAL	102
5.1 Dictionary of Subroutine Variables in Common	102
5.2 Description of Individual Subroutines	116
5.2.1 Subroutine ANGLE	116
⋮	
⋮	
5.2.53 Subroutine VROOTS	145
APPENDIX - LISTING OF COMPUTER PROGRAM TSFOIL	146
REFERENCES	177

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page No.</u>
4-1	Coordinate system and airfoil geometry.	47
4-2	Summary of boundary conditions for free air flows.	48
4-3	Summary of boundary conditions for wind-tunnel simulations.	49
4-4	Sample test case 1 - program default option - subsonic, free air simulation (full output): Korn airfoil, $\delta = 0.115$, $\alpha = 0.15^\circ$, $M_\infty = 0.75$.	50
4-5	Sample test case 2 - subsonic, solid wall wind-tunnel simulation (full output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2^\circ$, $M_\infty = 0.75$, $H = 2$.	61
4-6	Sample test case 3 - subsonic, slotted tunnel wall simulation (abbreviated output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2^\circ$, $M_\infty = 0.75$, $H = 2$, $F = 0.066$.	73
4-7	Sample test case 4 - supersonic, free air simulation (abbreviated output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2^\circ$, $M_\infty = 1.1$.	78
4-8	Sample test case 5 - supersonic, free jet simulation (abbreviated output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2^\circ$, $M_\infty = 1.1$, $H = 2$.	84
4-9	Sample test case 6 - supersonic, perforated/porous wall simulation (abbreviated output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2^\circ$, $M_\infty = 1.1$, $H = 2$.	90
4-10	Comparison of hodograph and relaxation solutions for the surface pressure distribution on a NLR quasi-elliptical airfoil with $\delta = 0.1212$, $\alpha = 1.32^\circ$ and $M_\infty = 0.7557$.	97
4-11	Comparison of hodograph and relaxation solutions for the surface pressure distribution on a Korn airfoil with $\delta = 0.115$, $\alpha = 0.12^\circ$, and $M_\infty = 0.75$.	97
4-12	Comparison of relaxation solutions for the surface pressure distribution of a NACA 0012 airfoil at $\alpha = 2^\circ$, $M_\infty = 0.630$.	98
4-13	Comparison of conservative time-dependent and nonconservative relaxation solutions of exact isentropic equation and the small-disturbance conservative solution for the surface pressure distribution on an NACA 64A410 airfoil at $M_\infty = 0.735$ and $\alpha = 1^\circ$.	98
4-14	Comparison of FCR and NCR solutions for the similarity surface pressure distribution on a parabolic arc airfoil with $\delta = 0.06$ and $M_\infty = 0.872$. (19)	99
4-15	Comparison of various solutions for the detached bow shock wave location for a parabolic arc airfoil with $\delta = 0.06$ and $M_\infty = 1.15$. (19)	99

LIST OF ILLUSTRATIONS (CONCLUDED)

<u>Figure</u>		<u>Page No.</u>
4-16	Comparison of small disturbance FCR solutions with data for the surface pressure distribution on a Garabedian-Korn airfoil at $\alpha = 1.38^\circ$, $M_\infty = 0.768$. (19)	100
4-17	Comparison of small disturbance FCR solutions with data for the surface pressure on a NACA 0012 at $\alpha = 0^\circ$ and $M = 0.8$. (19)	101
4-18	Comparison of free-air and wind-tunnel small disturbance FCR solutions for the surface pressure distribution on a NACA 64A410 airfoil at $\alpha = 2.0^\circ$ and $M_\infty = 0.802$.	101

LIST OF TABLES

<u>Table</u>		<u>Page No.</u>
4-I	Exponents for transonic similarity parameters	47

OPERATIONAL MANUAL FOR TWO-DIMENSIONAL
TRANSONIC CODE TSFOIL

By Stephen S. Stahara
Nielsen Engineering & Research, Inc.

1. SUMMARY

A detailed operating manual is presented for the computer code TSFOIL recently developed by Murman, Bailey, and Johnson at NASA/Ames Research Center. This code solves the two-dimensional, transonic, small-disturbance equations for flow past lifting airfoils in both free air and various wind-tunnel environments by using a variant of the finite-difference method initially proposed by Murman and Cole. A description of the theoretical and numerical basis of the code is provided, together with complete operating instructions and sample cases for the general user. In addition, a programmer's manual is also presented to assist the user interested in modifying the code. Included in the programmer's manual are a dictionary of subroutine variables in common and a detailed description of each subroutine.

2. INTRODUCTION

During the past several years, the rapid growth and development of numerical techniques capable of treating the nonlinear equations of transonic flow past aerodynamic configurations has generally resulted in a wealth of research computer programs but a lack of adequately documented user-oriented codes. In light of this, the present work was undertaken with the dual objectives of:

- (1) providing a computer code which is capable of solving a general class of two-dimensional, transonic, small-disturbance flows, and which incorporates as many of the recent advances in the theory as possible and
- (2) providing a systematically organized, user-oriented code with complete documentation to allow easy extension and modification for new applications and improvements.

A summary of the features and capabilities of the computer program was given in reference 1. This document provides an operational manual of the code, and includes a full description of the method of solution.

Although the basic solution technique used in the program is based on the original work of Krupp and Murman (refs. 2 and 3) various improvements and extensions have been incorporated in the version presented here. These include a simplified procedure for converging to the proper circulation for lifting airfoils (refs. 4 and 5), the appropriate difference operator to be used at shock waves (ref. 6), the evaluation of inviscid drag using momentum contour methods (ref. 7), wind-tunnel wall simulations (ref. 8), calculation of $M_\infty > 1$ transonic flows (ref. 9), the accelerated spread of circulation throughout the flow field for $M_\infty < 1$ free air flows (ref. 10), an analytic mesh (ref. 11) as well as a mesh cutting option to speed convergence, a successive line overrelaxation procedure based on the correction potential (ref. 12), and a pseudo-time term (ref. 12) to aid stability.

The operating manual presented here has been subdivided into two general categories. The first category is presented in section 4 and is a General User's Manual. In that section, a description of the problem solved by the code is given, including a discussion of the differential equation, geometry, boundary conditions, and numerical formulation of the problem, together with a description of the general operating procedure of the code, overall program flow chart, input/output description, error

messages, and sample calculations. All the information necessary to run the code are provided in that section. In section 5, a Programmer's Manual is presented to assist the user who wishes to modify the code to incorporate additional features or to extend the code beyond its present capabilities. Included in that section is a dictionary of all subroutine variables which appear in common blocks and a detailed description of the individual subroutines. In order to provide a user-code that will readily accommodate modifications, the program has been modularized in the sense that numerous (53) special-purpose subroutines have been used, rather than one large main program. Consequently, should the user not require some of the special calculations or output provided by the code, deletions can be made simply and directly.

In reference 1, it was noted that a number of computational problems, which are repeated here for convenience, existed in the version of the code then available. Some of those problems have since been alleviated by improvements to that initial version. In particular, the computation of lifting airfoils in supersonic free streams, which had proved troublesome, has been improved by the addition of a pseudo-time term as discussed in detail in section 4.1.5, and also by the introduction of an analytical mesh suggested in reference 11. A characteristic difficulty with $M_\infty > 1$ flows is that the maximum iteration error may initially decrease monotonically to a certain level, and then oscillate about that level, or in more severe cases, diverge. The maximum error almost always occurs at points near the airfoil and close to the bow or trailing edge shocks, or the locations where those shocks cross the mesh outer boundary. Addition of the pseudo-time term has served to stabilize this feature of the calculation, particularly when mesh cutting has been employed and the switch is made to the next finer mesh. Use of the analytic mesh frequently has served to preserve the monotonic convergence behavior of the solution, presumably both by the clustering of points near the leading and trailing edges and the expansion of points near the mesh boundaries.

Addition of these two features has also served to enhance the convergence of $M_\infty < 1$ flows. For example, converged solutions were achieved for a number of transonic flows past the 64A410 airfoil which could not previously be obtained with the earlier version. Also, supercritical flows for which the shock wave on the airfoil upper surface reached the trailing edge had proven difficult to calculate. For these cases, use of the analytic mesh appears to aid convergence considerably.

However, the occurrence of irregularities in the solution near the leading edge, as noted in reference 1, remains in the present version. Although these also appear in Krupp's results (ref. 2), in the current program they are predicted to be somewhat stronger. These dissimilarities may be due to the slight differences in numerical procedures used. Of course, one of the inherent deficiencies in any first-order, small-disturbance theory is the inaccurate treatment of the leading edge. A more accurate treatment, such as those suggested in references 13 and 14, has not been incorporated in the program, although a Reigel's rule option has been added.

The program has been written in FORTRAN IV source language and developed for use on a CDC 7600 computer using the OPT=2 compiler. The memory requirements are $(57K)_8$ and $(30K)_8$ words for small core and large core, respectively.

3. LIST OF SYMBOLS

$A_{j,i}$	coefficient matrix, equation (36)
c	airfoil chord
\bar{C}_i	column vector containing values of the correction potential $C_{j,i}$ along the i^{th} column of the finite-difference mesh, equation (47)
C_D	drag coefficient, equation (34)
$C_{j,i}$	correction potential, equation (42)
C_m	pitching-moment coefficient, equation (33)
C_p	pressure coefficient, equation (28)
C_{p_u}	pressure coefficient associated with the airfoil upper surface
C_{p_l}	pressure coefficient associated with the airfoil lower surface
f_i	column vector associated with the i^{th} column of the finite-difference grid, equation (46)
F	nondimensional wind-tunnel slot parameter, equation (20)
$F_{u,l}$	dimensionless affine function describing, respectively, the ordinates of the upper and lower airfoil surface, equation (7)
H	ratio of wind-tunnel half height to airfoil chord
\tilde{H}	transonically scaled tunnel half height to airfoil chord ratio, equation (22)
i,j	indices identifying the grid point at the i^{th} column and j^{th} row
k	exponent of Mach number in similarity definition of the small-disturbance equation, equations (2) and (6)
K	transonic similarity parameter, equation (5)
L	nondimensional spacing of slot centerlines, normalized by c ; equation (20)
m	exponent of Mach number in similarity definition of transonically scaled y coordinate, equation (4)
M	local Mach number, equation (30)
M_∞	free-stream Mach number
n	exponent of Mach number in similarity definition of velocity potential, equation (1)

p	pressure
P	nondimensional experimentally determined wind-tunnel porosity parameter equal to $2\Phi_y(x, \pm H)/\Delta p_{\text{wall}}(x, \pm H)/q_\infty$, equation (19)
\tilde{P}	transonically scaled wind-tunnel porosity parameter, equation (23)
P_0	shock-wave total pressure loss
P_∞	free-stream pressure
q_∞	free-stream dynamic pressure
$R_{j,i}$	residual matrix, equation (45)
S	nondimensional slot width, normalized by c ; equation (20)
U_∞	free-stream velocity
(x,y)	nondimensional Cartesian coordinate system with x axis directed rearward and aligned with the free stream; coordinates normalized by c
(x',y')	equal to (cx,cy)
\tilde{y}	transonically scaled y coordinate, equation (4)
$y_{u,l}$	ordinates of the upper (u) and lower (l) airfoil surface, equation (7)
α	angle of attack
Γ	circulation, equation (12)
Γ_{ff}	far-field circulation, equation (13)
γ	ratio of specific heats
δ	airfoil thickness ratio
ϵ	parameter multiplying pseudo-time term $\Delta t \phi_{xt}/\Delta x$ in small-disturbance partial differential equation, equation (48)
θ	local flow angle, equation (29)
$\mu_{j,i}$	switching parameter, equation (37)
Φ	disturbance velocity potential, equation (1)
$\bar{\Phi}$	total velocity potential, equation (1)
ϕ	dimensionless perturbation velocity potential, equation (1)
ϕ_{ff}	asymptotic far-field solution of the perturbation velocity potential

$\Delta\phi(x)$ jump in perturbation velocity potential along trailing vortex wake, equations (14) and (16)

$\Delta\phi_{te}$ jump in perturbation velocity potential at the trailing edge, equation (12)

ω relaxation parameter, equation (44)

ω_e relaxation parameter for elliptic points, equation (44)

ω_Γ relaxation parameter for circulation, equation (13)

4. GENERAL USER'S MANUAL

4.1 Description of Program

This section provides an overall description of the program, a brief discussion of the theoretical and numerical formulation of the problem solved by the code, including a description of the differential equation, airfoil geometry, boundary conditions, and finite-difference method. Also, the general operating procedure of the code is reviewed and an overall program flow chart is provided.

4.1.1 Differential equations and transonic scaling.- The partial differential equation solved by TSFOIL is the transonic small-disturbance equation for the perturbation velocity potential cast in terms of transonic similarity variables. Because of the nonuniqueness of the scaling, various forms have been adopted by different authors. If $(\bar{\Phi}, \phi)$ are, respectively, the physically and transonically scaled perturbation potentials, the various forms can be combined in the following general manner:

$$\bar{\Phi}(x', y') = cU_{\infty}[x + \phi(x, y) + \dots] = cU_{\infty}\left[x + \frac{\delta^{2/3}}{M_{\infty}^n} \phi(x, \tilde{y}) + \dots\right] \quad (1)$$

where c is the airfoil chord, U_{∞} and M_{∞} , respectively, are the free-stream velocity and Mach number, δ is the airfoil thickness ratio, (x, y) are the physical Cartesian coordinates (x', y') normalized by airfoil chord, and \tilde{y} the transonically scaled y coordinate given below. Now, $(\bar{\Phi}, \phi)$ satisfy

$$\left[(1 - M_{\infty}^2) \bar{\Phi}_{xx} - \frac{\gamma + 1}{2} M_{\infty}^k \bar{\Phi}_x^2 \right]_x + [\bar{\Phi}_y]_y = 0 \quad (2)$$

$$\left[K \phi_x - \frac{\gamma + 1}{2} \phi_x^2 \right]_x + [\phi_{\tilde{y}}]_{\tilde{y}} = 0 \quad (3)$$

where

$$\tilde{y} = \delta^{1/3} M_{\infty}^m y \quad (4)$$

$$K = \frac{1 - M_{\infty}^2}{\delta^{2/3} M_{\infty}^{2m}} \quad (5)$$

and the exponents (k,m,n) are related by

$$k = 2m + n \quad (6)$$

The different forms of the transonic similarity variables depend essentially on the particular choice of the exponents (k,m,n), and Table I provides a summary of the particular values proposed by Cole, Spreiter, or Krupp which have been incorporated into the program. The partial differential equation actually solved by TSFOIL is then equation (3).

4.1.2 Geometry.- The coordinate system used in the program is the Cartesian system shown in figure 4-1, with the airfoil located along the x-axis between 0 and 1, at angle of attack, α , and with a shape given by

$$y_{u,\ell}(x) = \delta F_{u,\ell}(x) \quad 0 \leq x \leq 1 \quad (7)$$

where (u, ℓ) denotes the upper and lower surfaces, and $F_{u,\ell}(x)$ is a function of order one describing the ordinates of a family of affinely related airfoils. Default profiles incorporated into the program are the parabolic-arc

$$F_{u,\ell}(x) = \pm 2x(1-x) \quad (8)$$

the NACA four-digit series

$$F_{u,\ell}(x) = \pm(1.4845\sqrt{x} - 0.63x - 1.758x^2 + 1.415x^3 - 0.5075x^4) \quad (9)$$

and the Korn airfoil described in reference 15. Provision has been made in the program for the user to input a profile of his choice by either a formula or a table of ordinates.

4.1.3 Boundary conditions.- The boundary conditions applied to this problem consist of the flow tangency condition at the airfoil surface, a condition for lifting airfoils assuring the value of circulation be unique, and some far-field conditions to be satisfied on the outer boundaries of the computational mesh. Within the framework of small-disturbance theory, the tangency condition at the airfoil can be linearized and applied along the slit $\tilde{y} = 0$, $0 \leq x \leq 1$ as

$$\phi_{\tilde{y}}(x, 0^+) = \frac{dF_u(x)}{dx} - \frac{\alpha}{\delta} \quad (10)$$

$$\phi_{\tilde{y}}(x, 0^-) = \frac{dF_l(x)}{dx} - \frac{\alpha}{\delta} \quad (11)$$

For lifting airfoils, two options have been incorporated in the program to insure that the circulation is unique. The first is the usual Kutta condition which is satisfied by requiring that ϕ_x (pressure) be continuous along the line $\tilde{y} = 0$, $x \geq 1$, and that $\phi_{\tilde{y}}$ (flow angle) be continuous across $\tilde{y} = 0$, $x > 1$. The disturbance potential ϕ remains single-valued by introduction of a cut along the $\tilde{y} = 0$ axis downstream of the airfoil along which ϕ is discontinuous, jumping by the value of the circulation Γ defined by

$$\Gamma = - \oint d\phi = \Delta\phi_{te} \quad (12)$$

where the contour is taken around the airfoil surface, and the resultant integral is equal to the jump in potential at the trailing edge. The circulation Γ_{ff} used in the asymptotic far-field solutions to update the outer boundaries is determined from the relation

$$\Gamma_{ff}^{n+1} = \Gamma_{ff}^n - \omega_{\Gamma} (\Gamma_{ff}^n - \Delta\phi_{te}^n) \quad (13)$$

where ω_{Γ} is the relaxation factor for circulation and n is the iteration count. The updated jump in potential $\Delta\phi(x)$ along the slit $\tilde{y} = 0$, $x > 1$ is calculated by the program according to the formula

$$\Delta\phi(x) = \Delta\phi_{te}^n + \frac{x-1}{x_{max}-1} (\Gamma_{ff}^{n+1} - \Gamma_{ff}^n) \quad (14)$$

From the results of reference 4, a value of circulation relaxation factor $\omega_{\Gamma} = 1.0$ (default value) is recommended. This implies that the updated far-field circulation be set equal to the new jump in potential at the trailing edge,

$$\Gamma_{ff}^{n+1} = \Delta\phi_{te}^n \quad (15)$$

and that the jump in potential along the trailing vortex wake be held constant and equal to

$$\Delta\phi(x) = \Delta\phi_{te}^n \quad (16)$$

Numerical experimentation has shown that the simpler representation of equations (15) and (16) gives the same results and convergence rate as both the method of Krupp (ref. 2) and the closely related procedure of equations (13) and (14).

The second option available to insure uniqueness of circulation is to specify the lift. In this case, the Kutta condition is not satisfied, and the pressure becomes double-valued at the trailing edge. The jump in potential at the trailing edge, as well as the far-field circulation, and the potential jump along the trailing vortex wake are given by

$$\Delta\phi_{te} = \Gamma_{ff} = \Delta\phi(x) = \frac{(C_L)_{SET}}{2} \quad (17)$$

where $(C_L)_{SET}$ is the specified lift coefficient.

For conditions at the outer boundaries, provision is made in the program to represent free air flows and various wind-tunnel wall simulations for both $M_\infty < 1$ and $M_\infty > 1$. For subsonic free air flows, the program incorporates the asymptotic far-field solution ϕ_{ff} given by a compressible vortex and doublet, while for supersonic free air flows the appropriate far-field conditions incorporated are that the perturbation velocities ϕ_x, ϕ_y vanish at the upstream boundary and that the outgoing wave condition

$$\phi_y \sim = \pm \sqrt{K} \phi_x \quad (18)$$

be enforced at the top and bottom boundaries in order to prevent reflected waves. Strictly speaking, for $M_\infty > 1$, no condition is required at the downstream boundary if as we assume it is sufficiently far removed downstream so that the outflow is entirely supersonic. However, for actual numerical application, some condition is needed (ref. 8) to treat subsonic outflow which may develop during the course of the relaxation solution process prior to convergence. By numerical experimentation, the condition $\phi_x = 0$ was found to be satisfactory. Figure 4-2 summarizes these conditions for free air calculations.

For wind-tunnel simulations, the classical homogeneous tunnel wall boundary condition

$$\Phi_x(x, \pm H) \pm FH\Phi_{xy}(x, \pm H) \pm \frac{1}{P} \Phi_y(x, \pm H) = 0 \quad (19)$$

is applied at the tunnel walls $y = \pm H$. Here, H is the ratio of tunnel half height to airfoil chord, F is the tunnel slot parameter, and P is the porosity parameter which accounts for viscous effects in the slot and must be determined experimentally. A simplified analysis has been made by Baldwin, et al. (ref. 16) for the slot parameter F , which he gives as

$$F = \frac{L}{\pi H} \ln \left[\csc \left(\frac{\pi S}{2L} \right) \right] \quad (20)$$

where L is the spacing between slot centerlines and S is the slot width, both normalized by airfoil chord. The program uses equation (19) in the similarity form

$$\phi_{\tilde{x}}(x, \pm \tilde{H}) \pm F \tilde{H} \phi_{\tilde{x}\tilde{y}}(x, \pm \tilde{H}) \pm \frac{1}{\tilde{P}} \phi_{\tilde{y}}(x, \pm \tilde{H}) = 0 \quad (21)$$

where (\tilde{H}, \tilde{P}) are the transonically scaled tunnel half height and porosity given by

$$\tilde{H} = \delta^{1/3} M_{\infty}^m H \quad (22)$$

$$\tilde{P} = P / (\delta^{1/3} M_{\infty}^m) \quad (23)$$

The following subcases of equation (19) have been incorporated into the program as separate options

$$\phi_{\tilde{y}}(x, \pm \tilde{H}) = 0 \quad \text{Solid Wall} \quad (24)$$

$$\phi_{\tilde{x}}(x, \pm \tilde{H}) = 0 \quad \text{Free Jet} \quad (25)$$

$$\tilde{P} \phi_{\tilde{x}}(x, \pm \tilde{H}) \pm \phi_{\tilde{y}}(x, \pm \tilde{H}) = 0 \quad \text{Ideal Perforated/} \\ \text{Porous Wall} \quad (26)$$

$$\phi_{\tilde{x}}(x, \pm \tilde{H}) \pm F \tilde{H} \phi_{\tilde{x}\tilde{y}}(x, \pm \tilde{H}) = 0 \quad \text{Ideal Slotted Wall} \quad (27)$$

while the general case has not yet been included. For tunnel flows, the upstream and downstream conditions used are the following. For $M_{\infty} < 1$, the asymptotic far-field solution ϕ_{ff} which satisfies equation (19) and the linearized version of the differential equation (3) is used. The case of choked flow, as treated in reference 17, has not been included in the program. For $M_{\infty} > 1$, the conditions $\phi_x, \phi_{\tilde{y}} = 0$ are enforced at the

upstream boundary, while the flow is required to be supersonic at the downstream boundary and the condition $\phi_x = 0$ is employed, as in the supersonic free air calculation. These conditions are summarized in figure 4-3.

4.1.4 Aerodynamic variables.- The aerodynamic quantities of interest output by the program are given below in terms of the small-disturbance similarity potential $\phi(x, \tilde{y})$. These include pressure coefficient, C_p , flow angle, θ , local Mach number, M , shock wave total pressure loss, $[P_O]/P_\infty$, and coefficients of lift, C_L , pitching moment, C_m , and drag, C_D .

$$C_p = \delta^{2/3} M_\infty^{-n} (-2\phi_x) \quad (28)$$

$$\theta = \delta \phi_{\tilde{y}} \quad (29)$$

$$M = \left\{ 1 - \left[K - (\gamma + 1) \phi_x \right] \delta^{2/3} M_\infty^{2m} \right\}^{1/2} \quad (30)$$

$$\frac{[P_O]}{P_\infty} = 1 + \delta^2 \frac{\gamma(\gamma + 1)}{12} [\phi_x]_s^3 \quad (31)$$

$$C_L = - \int_0^1 (C_{pU} - C_{p\ell}) dx \quad (32)$$

$$C_m = \int_0^1 (x - x_m) (C_{pU} - C_{p\ell}) dx \quad (\text{positive nose-up}) \quad (33)$$

$$C_D = \delta^{5/3} M_\infty^{-n} \left\{ \oint_C \left[K\phi_x^2 - \phi_{\tilde{y}}^2 - \frac{2}{3} (\gamma + 1) \phi_x^3 \right] d\tilde{y} - \phi_x \phi_{\tilde{y}} dx - \frac{\gamma + 1}{6} \int_{SDC} [\phi_x]_s^3 d\tilde{y} \right\} \quad (34)$$

where $[\phi_x]_s$ is the jump in the streamwise perturbation velocity across a shock.

4.1.5 Numerical formulation.- The numerical formulation of the problem together with the techniques employed to solve it are presented in this section. The numerical solution of the problem is initiated by replacing equation (3) by the following finite-difference form given by Bailey and Ballhaus in reference 18.

$$\begin{aligned}
& \left\{ (1-\mu_{j,i}) A_{j,i} [(\phi_x)_{j,i+1/2} - (\phi_x)_{j,i-1/2}] \right. \\
& \quad \left. + \mu_{j,i-1} A_{j,i-1} [(\phi_x)_{j,i-1/2} - (\phi_x)_{j,i-3/2}] \right\} / (x_{i+1/2} - x_{i-1/2}) \\
& \quad + [(\phi_y)_{j+1/2,i} - (\phi_y)_{j-1/2,i}] / (y_{j+1/2} - y_{j-1/2}) = 0 \quad (35)
\end{aligned}$$

where

$$A_{j,i} = 1 - M_\infty^2 - (\gamma + 1) M_\infty^k (\phi_x)_{j,i} \quad (36)$$

$$\mu_{j,i} = \begin{cases} 0 & \text{for } A_{j,i} > 0 \\ 1 & \text{for } A_{j,i} < 0 \end{cases} \quad (37)$$

$$(\phi_x)_{j,i+1/2} = \frac{\phi_{j,i+1} - \phi_{j,i}}{x_{i+1} - x_i} \quad (38)$$

$$(\phi_y)_{j+1/2,i} = \frac{\phi_{j+1,i} - \phi_{j,i}}{y_{j+1} - y_j}, \text{ etc.} \quad (39)$$

and $(\phi_x)_{j,i}$ in equation (36) is approximated by

$$(\phi_x)_{j,i} = \frac{1}{2} [(\phi_x)_{j,i+1/2} + (\phi_x)_{j,i-1/2}] \quad (40)$$

This form of the difference equation with the switching parameter $\mu_{j,i}$ encompasses the four operators of Murman's fully conservative relaxation (FCR) method (ref. 15); that is, elliptic ($\mu_{j,i-1} = 0, \mu_{j,i} = 0$), hyperbolic ($\mu_{j,i-1} = 1, \mu_{j,i} = 1$), parabolic ($\mu_{j,i-1} = 0, \mu_{j,i} = 1$), and shock point ($\mu_{j,i-1} = 1, \mu_{j,i} = 0$), and for evenly spaced meshes is identical to that method. For unequally spaced meshes, the first term differs from Murman's (ref. 19) by a factor of $(x_i - x_{i-2}) / (x_{i+1} - x_{i-1})$ in hyperbolic regions. The finite-difference equation (35) can be cast in the following form

$$\begin{aligned}
& (1-\mu_{j,i})A_{j,i} (CXXL_i \phi_{j,i-1} - CXXC_i \phi_{j,i} + CXXR_i \phi_{j,i+1}) \\
& + \mu_{j,i-1} A_{j,i-1} (CXXL_{i-1} \phi_{j,i-2} - CXXC_{i-1} \phi_{j,i-1} + CXXR_{i-1} \phi_{j,i}) \\
& + CYYD_j \phi_{j-1,i} - CYYC_j \phi_{j,i} + CYYU_j \phi_{j+1,i} = 0
\end{aligned} \tag{41}$$

where the coefficients ($CXXL_i$, $CXXC_i$, etc.) depend on mesh geometry only and can be deduced from equation (35). Although equation (41) forms the basis of the numerical analysis, the solution technique actually employed in the program is similar to that recently suggested by Jameson (ref. 12) in which the difference equation (41) is recast in terms of a correction potential $C_{j,i}$ defined by

$$\phi_{j,i}^{n+1} = \phi_{j,i}^n + C_{j,i} \tag{42}$$

where n represents the current iteration cycle.

The governing equation for $C_{j,i}$ is determined in the following fashion. First, we consider the difference equation (41) in which the values $\phi_{j,i}$ of the potential along the current column are replaced by their provisional values $\tilde{\phi}_{j,i}$, while the values ($\phi_{j,i-1}^{n+1}$, $\phi_{j,i-2}^{n+1}$, $\phi_{j,i+1}^n$, $A_{j,i-1}^n$, $A_{j,i}^n$) are used to evaluate the remaining terms. Similarly, we define a residual $R_{j,i}$ by considering the same equation but with the result $\phi_{j,i}^n$ used for $\phi_{j,i}$. By subtracting those two equations and then eliminating $\tilde{\phi}_{j,i}$ by use of the overrelaxation formula

$$\phi_{j,i}^{n+1} = \phi_{j,i}^n + \omega (\tilde{\phi}_{j,i} - \phi_{j,i}^n) \tag{43}$$

where

$$\omega = \left\{ \begin{array}{ll} \omega_e, & 1 < \omega_e < 2 \quad \text{for locally subsonic flow } (A_{j,i} > 0) \\ 1 & \text{for locally supersonic flow } (A_{j,i} < 0) \end{array} \right\} \tag{44}$$

the following difference equation results for $C_{j,i}$:

$$\begin{aligned}
& \left\{ -(1-\mu_{j,i})A_{j,i}^n CXXC_i / \omega_e + \mu_{j,i-1} A_{j,i-1}^n CXXR_{i-1} \right\} C_{j,i} \\
& + CYYD_j C_{j-1,i} + CYYU_j C_{j+1,i} = -R_{j,i}
\end{aligned} \tag{45}$$

We note that in the derivation of equation (45), new values $\phi_{j,i}^{n+1}$ rather than provisional values $\tilde{\phi}_{j,i}$ are used to evaluate ϕ_{yy} at both subsonic and supersonic points. As pointed out in reference 12, this avoids a

discontinuity in ϕ_{yy} at a sonic line.

The solution of equation (45) then proceeds by using an iterative successive line over-relaxation (SLOR) algorithm as follows. The system of equations for each vertical line is written as

$$A\bar{C}_i = f_i \quad (46)$$

where \bar{C}_i is the J dimensional column vector

$$\bar{C}_i = \begin{pmatrix} C_{1,i} \\ \vdots \\ C_{j,i} \end{pmatrix} \quad (47)$$

A is a $J \times J$ dimensional, diagonally dominant, tridiagonal matrix, and f_i is a J dimensional column vector. The solution process consists of making an initial guess for ϕ , and then successively sweeping the grid from the upstream to downstream boundary, solving for new values of $C_{j,i}$ from equation (46) by direct elimination using the method of triangular decomposition, and then determining new values of ϕ from equation (42). The initial guess for ϕ may be selected from a uniform flow ($\phi = 0$) or a previous case stored in core or on a peripheral unit. With regard to the over-relaxation parameter ω_e , acceptable values on a fine mesh for ω_e are $1.8 < \omega_e < 1.95$, although optimum values for particular applications must be determined by numerical experimentation.

A mesh refinement option has been incorporated into the program to enhance the rate of convergence. Under this option, the user may elect to initiate the calculation either on a mesh (medium mesh) obtained by deleting every other mesh point of the input mesh (fine mesh), or on an even coarser mesh (coarse mesh) obtained by again deleting every other point on the medium mesh. In the latter case, the iterative solution is started on the coarse mesh, and after appropriate convergence is reached, continued on the medium mesh, and then completed on the input mesh. Also, in order to improve stability and aid convergence, an option has been provided in the program to add the pseudo-time term in the form

$$- \epsilon \frac{\Delta t}{\Delta x} \phi_{xt} \quad (48)$$

where

$$\frac{\Delta t}{\Delta x} \phi_{xt} = \frac{(\phi_{j,i}^{n+1} - \phi_{j,i}^n) - (\phi_{j,i-1}^{n+1} - \phi_{j,i-1}^n)}{(x_i - x_{i-1})^2} = \frac{C_{j,i} - C_{j,i-1}}{(x_i - x_{i-1})^2} \quad (49)$$

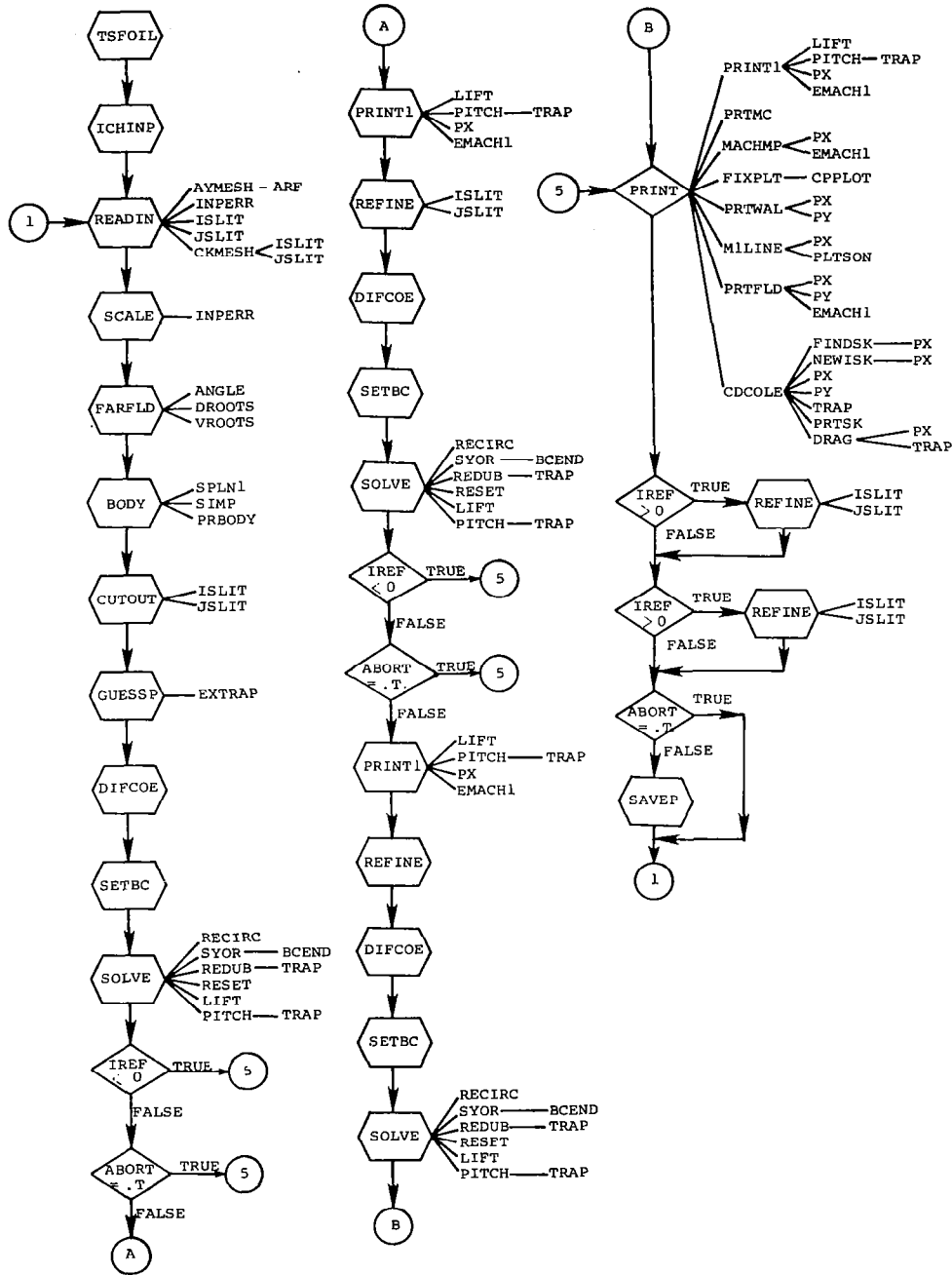
to the left-hand side of equation (45) as suggested in references 12 and 18. The parameter ϵ is an input parameter with a value in the range $0 \leq \epsilon \leq 1$. Considerable improvement in stability and convergence has been obtained in a number of both subsonic and supersonic free-stream calculations when this term is added. However, this result is not true for all cases and extensive numerical experimentation to determine optimum values for ϵ has not yet been carried out. Consequently, the user must, of necessity, determine suitable values for his particular application. An appropriate first guess would be $\epsilon = 0.2$ which is the default value.

Finally, we note that the numerical methods used in this program are of the shock-capturing, rather than shock-fitting type. This means that shock waves are spread over several mesh intervals and do not appear as sharp discontinuities. Calculations have shown (ref. 19) that for subsonic flow downstream of shock waves, four mesh points are sufficient to capture the shock jump; whereas for supersonic downstream flow, six to ten points are required. The latter is due to dissipation effects which dominate the first-order accurate hyperbolic operator in equation (35), and smooth out shocks to that extent. This is a weakness in the method if sharp supersonic oblique shocks are required, and necessitates special indentations of the momentum contours used to calculate the inviscid drag for all cases where $M_\infty > 1$, and also for cases where $M_\infty < 1$ when tail plus wake shocks occur. See subroutine CDCOLE in section 5.2 for a description of the momentum contours incorporated in the program.

4.1.6 Operating procedure.— The general operating procedure of the program is straightforward and is described in this section. First, the main program prints all of the input cards (ECHINP), then reads the input for the first case, checks it for errors, assigns default values to those parameters not specified, and then prints all of the input parameters (READIN). Next, if the input was in physical rather than transonic form, the appropriate scaling is applied to render all the necessary variables into transonic similarity form (SCALE). The unit-strength boundary data on the far-field mesh boundaries are then computed (FARFLD). Next, the geometrical characteristics of the airfoil that are required in the finite-difference calculations are determined (BODY). The medium and

coarse meshes are then established if the user has opted to use the mesh refinement option (CUTOOUT). Next, the ϕ array is initialized (GUESSP), and then the various difference coefficients that depend on mesh size are calculated for the current mesh (DIFCOE). The I and J indices for solving the difference equations are set, and the airfoil slope boundary condition is multiplied by mesh spacing constants for use in the finite-difference calculation (SETBC). The relaxation solution of the difference equations is then carried out in an iterative fashion by sweeping the flow field from upstream to downstream boundaries with convergence information being output after a specified number of sweeps (SOLVE). If the calculations have been carried out on an intermediate mesh, flow field information on the dividing streamline, including the airfoil, are printed (PRINT1), and then mesh points are added to obtain the next finer mesh (REFINE). The sequence of relaxation calculations are then repeated (DIFCOE, SETBC, SOLVE, PRINT1) until the final mesh is reached. At this point, the final flow information is output (PRINT), the solution stored for later use (SAVEP), and the calculation for the next case begun. If the solution should diverge during the course of the finite-difference calculation (SOLVE), the computation is then aborted, with portions of the final printout (PRINT) provided to assist the user in determining the cause of the difficulty.

4.1.7 Overall program flow chart.- An overall program flow chart is provided below to illustrate the logical flow of the code.



4.2 Program Input

The input required by the program is described in this section. A dictionary of input variables is provided first, followed by a description of the various input control options available to the user. All of the default values of the input variables are given next, followed by a description of the input data format including preparation of a sample run.

4.2.1 Dictionary of input variables.- All variables that are input to this program are described in the following list.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
AK	transonic similarity parameter; equal to $1 - M_\infty^2 / M_\infty^2 m \delta^{2/3}$ where m is defined in Table 4-I according to the type of transonic scaling used; input required only if transonic scaling used in input/output; that is, PHYS = .F.
ALPHA	angle of attack, α , in degrees for PHYS = .T.
AMESH (logical)	control option for specification of analytical default mesh; equal to .T. or .F.
BCFOIL (integer)	control option for specification of airfoil geometry; equal to 1, 2, 3, or 4
BCTYPE	control option for type of flow environment; equal to 1, 2, 3, 4, or 5
CLSET	specified lift coefficient; input only if Kutta condition not enforced; that is, KUTTA = .F.
CVERGE	convergence criterion for maximum iteration error in ϕ
DELTA	airfoil thickness ratio δ
DVERGE	divergence criterion for maximum iteration error in ϕ
EMACH	free-stream Mach number, M_∞ ; because of the particular far-field formulation used in this program, M_∞ can be close but not exactly equal to 1
EPS	coefficient of pseudo-time term $\Delta t \phi_{xt} / \Delta x$ in differential equation
F	slot parameter for ideal slotted wind tunnel
FCR (logical)	control option for specification of type of differencing to be used at shock waves; equal to .T. or .F.
GAM	ratio of specific heats γ
H	ratio of tunnel half height/airfoil chord
ICUT	control option for mesh cutting and refinement; equal to 0, 1, or 2

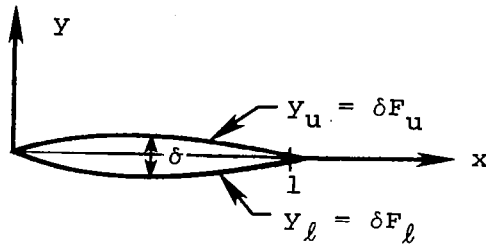
<u>VARIABLE</u>	<u>DESCRIPTION</u>
IMAXI	number of x-mesh points in input mesh; $IMAXI \leq 100$
IMIN	value of I index designating the x-mesh point where the finite-difference calculation is to start
IPRTER	control option for print frequency of intermediate output
JMAXI	number of y-mesh points in input mesh; $JMAXI \leq 100$
JMIN	value of J index designating the y-mesh point where the finite-difference calculation is to start
KUTTA (logical)	control for Kutta condition specification; equal to .T. or .F.
MAXIT	maximum allowed number of iteration sweeps through the flow field
NL	number of ordinate points used to describe airfoil lower surface; $NL \leq 100$
NU	number of ordinate points used to describe airfoil upper surface; $NU \leq 100$
PHYS (logical)	control for specification of the type of scaling (physical or transonic) used in input/output; equal to .T. or .F.
POR	porosity factor for ideal porous/perforated wind-tunnel wall
PRTFLO	control for printout of final flow field; equal to 1, 2, or 3
PSAVE (logical)	control for saving restart block of values on unit 3; equal to .T. or .F.
PSTART (integer)	control for initialization of ϕ array; equal to 1, 2, or 3
SIMDEF (integer)	control for definition of transonic similarity scaling; equal to 1, 2, 3, or 4
WCIRC	relaxation factor for circulation
WE(I)	vector array of length 3 representing values of the relaxation factor for elliptic points on the coarse, medium, and fine meshes, respectively; if specified, all three values must be given
XGRDIN	control for input of user-designated x-grid; equal to .T. or .F.

<u>VARIABLE</u>	<u>DESCRIPTION</u>
XIN(I)	vector array of length IMAXI of user-supplied x-mesh points; normalized by airfoil chord; maximum of 100
XL(I)	vector array of length NL of x-ordinates of the airfoil lower surface; normalized by airfoil chord; maximum of 100
XU(I)	vector array of length NU of x-ordinates of the airfoil upper surface; normalized by airfoil chord; maximum of 100
YGRDIN	control for input of user-designated y-grid; equal to .T. or .F.
YIN(I)	vector array of length JMAXI of user-supplied y-mesh points, normalized by airfoil chord for PHYS = .T.; maximum of 100
YL(I)	vector array of length NL of y-ordinates of the airfoil lower surface; normalized by airfoil chord times thickness ratio; maximum of 100
YU(I)	vector array of length NU of y-ordinates of the airfoil upper surface; normalized by airfoil chord times thickness ratio; maximum of 100

4.2.2 Control options.- Those input variables which allow the user to select the various options incorporated in this program are detailed in this section together with the options they control.

Airfoil geometry option: Four choices are available to the user for the specification of the airfoil geometry. Selection is made according to the value specified (1, 2, 3, or 4) for the integer index BCFOIL. For BCFOIL = 1, the program uses the symmetric NACA four-digit (00XX) series geometry; for BCFOIL = 2, a parabolic-arc profile is used; for BCFOIL = 3, user-supplied values of upper and lower airfoil ordinates (YU(I), YL(L)) versus abscissas (XU(I), XL(I)) are required by the program, which then calculates internally the ordinates and slopes at the input mesh points by cubic spline interpolation--under this geometry option, a default airfoil (Korn) is provided; for BCFOIL = 4, the user provides his own specification (i.e., ordinates and slopes at the input mesh points) of the airfoil geometry. We note that the last option (BCFOIL = 4) is, of course, open (inactive) and requires user-supplied information in subroutine BODY. If a formula is available for the airfoil profile desired, this option should be used. Implementation is straightforward and an illustrative

example is given below. Consider the symmetric profile formed by the sinusoidal curve given by:



$$Y_{u,l}(x) = \pm \frac{\delta}{2} \sin(\pi x)$$

so that

$$F_{u,l}(x) = \pm \frac{1}{2} \sin(\pi x)$$

$$\frac{dF_{u,l}(x)}{dx} = \pm \frac{\pi}{2} \cos(\pi x)$$

Thus, the following FORTRAN statement changes would be entered into subroutine BODY. Immediately after statement number 400 (see the listing of subroutine BODY in the Appendix), the abnormal stop message and STOP card would be removed and the following statements inserted:

```

PI = 3.1415927
IC = 0
DO 425 I = ILE, ITE
IC = IC + 1
Z = XIN(I)
Z1 = PI * Z
XFOIL(IC) = Z
FU(IC) = SIN(Z1)/2.
FL(IC) = - FU(IC)
FXU(IC) = PI * COS(Z1)/2.
FXL(IC) = - FXU(IC)
425 CONTINUE

```

We note that it is necessary that subroutine BODY define the arrays XFOIL(I), FU(I), FL(I), FXU(IC), and FXL(I) which are, respectively, the x-ordinates of the input x-mesh XIN(I) which fall on the airfoil, the y-ordinates of the airfoil upper and lower surface divided by the thickness ratio, and the slopes of the airfoil upper and lower surface divided by the thickness ratio, all calculated at the XFOIL(I) points.

Flow environment option: To specify the type of flow environment; that is, free air or tunnel simulation conditions, the integer index BCTYPE must be set. Six different choices have been incorporated in the program according to the following schedule:

<u>BCTYPE</u>	<u>Flow Condition</u>
1	Free air
2	Solid wall tunnel
3	Free jet
4	Ideal slotted wall tunnel
5	Ideal porous/perforated wall tunnel
6	General tunnel wall

The program default value is BCTYPE = 1.

Conservative differencing option: An option has been provided for the user to specify whether conservative or nonconservative differencing be used at shock waves. This choice is made according to the logical index FCR:

<u>FCR</u>	<u>Shock Difference Type</u>
.T.	Fully conservative
.F.	Nonconservative

Program default value is FCR = .T.

Mesh refinement option: A convergence acceleration feature has been incorporated in the program which provides the user the option to initiate the iterative solution on a mesh that is coarser than the input mesh (fine mesh) on which the solution is ultimately desired. Provision has been made for two automatic grid reductions (medium and coarse) by the program. Since the grid change is accomplished by the deletion of every other mesh point, the following simple rules should be followed in specifying the input (fine) grid to take full advantage (i.e., two cuts) of this feature.

These rules consist of making the following grid point differences a multiple of four,

IMAXI - ITE
ITE - IMIN
JMAXI - JUP + 1
JLOW - JMIN + 1

which insure that each of the above four grid intervals can be halved twice. See Section 5.1 under COM 1 for a description of these indices.

The control for mesh refinement is made by specification of the integer index ICUT according to:

<u>ICUT</u>	<u>Comment</u>
0	Input mesh used to convergence
1	Input mesh may be cut once
2	Input mesh may be cut twice

We note that for the cases ICUT = 1 or 2, if the grid criteria given above are not met, the program will try to adjust if possible the grid by adding (never deleting) points. However, since the stated conditions are so mild, it is recommended that they be followed directly if the grid refinement option is desired. Program default value is ICUT = 2.

Intermediate printout option: If the user wished to obtain intermediate information concerning the convergence of the solution, an option is provided which will cause a limited printout to occur each time a specified number of iteration sweeps through the flow field are completed. Output includes iteration count, lift and moment coefficients, iteration error in circulation, and absolute values and locations of the maximum iteration error in ϕ , the residual $R_{j,i}$, and the upper and lower surface pressure coefficients. Control is through specification of the integer index IPRTER which causes output print every IPRTERth iteration sweep through the field. Program default value is IPRTER = 0.

Kutta condition enforcement option: If, in lieu of enforcing the Kutta condition, the user wishes to specify instead the lift coefficient, an option has been provided in the program to accommodate this. Control for this option is through the logical index KUTTA where

KUTTAComment

.T.	Kutta condition enforced
.F.	Lift coefficient specified; trailing-edge pressure discontinuous

Program default value is KUTTA = .T.

Physical or transonic I/O scaling option: A user choice for the type of scaling used in the input/output is available through the logical index PHYS. The I/O can be in either the usual physical scaled form (PHYS = .T.); that is, lengths normalized by airfoil chord, c , velocities by free-stream velocity, U_∞ , pressure by free-stream dynamic pressure, q_∞ , etc., or in the appropriate transonically scaled form (PHYS = .F.). If transonic scaling is used, we note the following requirements: the lateral coordinate of the input grid must be scaled according to

$$\tilde{Y} = \delta^{1/3} M_\infty^m Y$$

the angle of attack by

$$\tilde{\alpha} = \alpha / (\delta \cdot 180^\circ / \pi)$$

the half-tunnel height by

$$\tilde{H} = \delta^{1/3} M_\infty^m H$$

the lift coefficient CLSET by

$$(\text{CL}\tilde{\text{SET}}) = \text{CLSET} / \delta^{2/3} M_\infty^{-n}$$

the porosity by

$$\tilde{P} = P / \delta^{1/3} M_\infty^m$$

the velocity potential by

$$\tilde{\phi} = \delta^{2/3} M_\infty^{-n} \phi$$

and instead of specifying the Mach number, M_∞ , the similarity parameter K must be given, where

$$K = \frac{1 - M_\infty^2}{M_\infty^{2m} \delta^{2/3}}$$

and where the exponents (m,n) depend upon the choice of scaling and are defined in Table 4-I. Program default value is PHYS = .T.

Flow field printout option: In the final printout of a completed calculation, the user has the choice of obtaining flow field information [i.e., pressure coefficient, flow angle, and either local Mach number (PHYS = .T.) or local similarity parameter (PHYS = .F.)], either throughout or in a limited region of the flow field. This is controlled through the integer index PRTFLO according to

<u>PRTFLO</u>	<u>Flow Field Printout</u>
1	None
2	All
3	Three J lines about $ \phi_{j,i}^{n+1} - \phi_{j,i}^n _{\max}$ line

Program default value is PRTFLO = 1.

Result storage option: If at the completion of a calculation the user wishes, in addition to obtaining the printed output, to store the results on tape for future use, an option has been provided whereby the program will write the following information on tape 3:

- (1) run title
- (2) grid dimensions; IMAXI, JMAXI, IMIN, JMIN
- (3) C_L , M_∞ , α , δ , airfoil volume, far field doublet strength
- (4) X and Y grid arrays
- (5) $\phi_{j,i}$

Control is through the logical index PSAVE according to

<u>PSAVE</u>	<u>Storage</u>
.T.	Results stored on tape 3
.F.	No storage

Program default value is PSAVE = .F.

Initialization of ϕ option: Three choices are available for initializing the ϕ array to start the calculation. Selection is made according to the integer index PSTART where

<u>PSTART</u>	<u>Initialization of ϕ</u>
1	Set to zero
2	Read from tape 7
3	Used from previous case

Program default value is $PSTART = 1$.

Transonic similarity definition option: Four choices are available to the user for the specification of the particular transonic similarity form of the small-disturbance equation to be solved. The first three of these choices ($SIMDEF = 1, 2, \text{ or } 3$) have been incorporated into the program and are detailed below. The fourth ($SIMDEF = 4$) is an open (inactive) option, requiring user-supplied definitions of the various scaling parameters calculated in subroutine SCALE.

The three active options reflect the different scaling rules of Cole, Spreiter, or Krupp and depend essentially upon the choice of values for the exponents (k, m, n) in the following representations of the small-disturbance perturbation velocity potential equation, stretched-lateral coordinate, and transonic similarity parameter; that is,

$$\left[(1 - M_\infty^2) \phi_x - \frac{\gamma + 1}{2} M_\infty^k \phi_x^2 \right]_x + \phi_{yy} = 0$$

or

$$\left[K \phi_x - \frac{\gamma + 1}{2} \phi_x^2 \right]_x + \phi_{\tilde{y}\tilde{y}} = 0$$

where

$$\Phi(x, y) = \delta^{2/3} M_\infty^{-n} \phi(x, \tilde{y})$$

$$Y = \delta^{-1/3} M_\infty^{-m} \tilde{y}$$

$$K = \frac{1 - M_\infty^2}{M_\infty^{2m} \delta^{2/3}}$$

The values of the exponents (k, m, n) for each of the three cases are given in Table 4-I. Selection of the particular scaling rule according to the value of $SIMDEF$ is

<u>SIMDEF</u>	<u>Scaling</u>
1	Cole
2	Spreiter
3	Krupp
4	User defined

Program default value is SIMDEF = 3.

Default grid option: Two different sets of default grids have been incorporated in the program. These are the Krupp mesh defined in reference 2 and an analytical mesh based on a modified version of the grid discussed in reference 11. The program default x-mesh is the Krupp x-mesh XKRUPP(I), given in reference 2, while two default y-meshes are incorporated; that is, the Krupp y-mesh for free air calculations YFREE(I) and a modified y-mesh YTUN(I) for tunnel simulations. To use the analytical mesh, control is through the logical variable AMESH according to

<u>AMESH</u>	<u>Grid Specification</u>
.T.	Analytical mesh used
.F.	Either Krupp or user-specified mesh used

Program default value is AMESH = .F.

Input grid option: If the user wishes not to employ the default grids incorporated in the program, an option has been provided whereby an arbitrary mesh can be conveniently supplied to the program. Control is through the logical variables XGRDIN and YGRDIN according to

<u>XGRDIN</u>	<u>Grid Specification</u>
.T.	x-mesh provided by user in input data
.F.	Default x-mesh used

<u>YGRDIN</u>	<u>Grid Specification</u>
.T.	y-mesh provided by user in input data
.F.	Default y-mesh used

When the grids are user supplied, each element of the grid is defined in the input data, as illustrated in section 4.2.4. We note that the program default x-mesh is the Krupp x-mesh XKRUPP(I) defined in reference 2, while two default y-meshes are incorporated; that is, the Krupp y-mesh for free air calculations YFREE(I) and a modified y-mesh YTUN(I) for tunnel simulations. Program default values are XGRDIN = .F. and YGRDIN = .F.

4.2.3 Default values.- Each input variable has been assigned a default value that is stored in the BLOCK DATA subroutine. Consequently, unless the user wishes to change a variable from its default value, it is not necessary to enter it in the input data block. For convenience, the default values of all the input variables are tabulated below:

<u>Variable</u>	<u>Default Value</u>
AK	0.
ALPHA	0.12
AMESH	.F.
BCFOIL	3
BCTYPE	1
CLSET	0.
CVERGE	0.00001
DELTA	0.115
DVERGE	10.
EMACH	0.75
EPS	0.2
F	0.
FCR	.T.
GAM	1.4
H	0.
ICUT	2
IMAXI	77 (Krupp grid)
IMIN	1
IPRTER	10
JMAXI	56 (Krupp grid)
JMIN	1
KUTTA	.T.
MAXIT	500
NL	75 (Korn airfoil)
NU	100 (Korn airfoil)
PHYS	.T.
POR	0.
PRTFLO	1
PSAVE	.F.
PSTART	1
RIGF	0.
SIMDEF	3

<u>Variable</u>	<u>Default Value</u>
WCIRC	1.
WE(I), (I = 1,2,3)	1.8,1.9,1.95
XGRDIN	.F.
XIN(I)	XKRUPP(I)
XL(I)	(Korn airfoil)
XU(I)	(Korn airfoil)
YGRDIN	.F.
YIN(I)	YFREE(I) or YTUN(I)
YL(I)	(Korn airfoil)
YU(I)	(Korn airfoil)

It is important to realize that the above variables assume their default values only before the first case is run. Input variables, once user specified, are not changed internally by the program; thus, it is unnecessary to respecify them in subsequent cases if their values are to remain constant.

4.2.4 Input data format.-- With the exception of a run title card, which is the first card of each case and on which the user can provide any alphanumeric information desired in columns 1 to 80, all of the input data for the program are read in under a NAMELIST format called INP. Consequently, the input format is most easily demonstrated by an example. Consider two cases of simulating the flow of air ($\gamma = 1.4$, default) past an NACA 0010 airfoil in an ideal slotted wall wind tunnel where the slot parameter is 0.07, and the tunnel half-height-to-chord ratio is 6.5. For the first case, the free-stream Mach number is 0.70, the angle of attack is 1.5° , and it is decided to match the measured tunnel lift coefficient of 0.187. The fully conservative equations (default) will be solved on the YTUN (tunnel default) y-grid and the basic Krupp XKRUPP (default) x-grid, employing the mesh refinement option with two grid halvings (default). The maximum number of iteration sweeps allowed will be 250, with intermediate printout desired after every fifth sweep, a convergence criterion on ϕ of 1×10^{-4} , a divergence criterion of 10. (default), with final flow field printout confined to three rows centered about the maximum iteration error in ϕ , and with the final results stored on tape 3. The transonic equation will be used in the similarity form as given by Spreiter, with the input/output in physical scaling terms (default), with the initial guess of ϕ equal to zero (default), and with the circulation relaxation factor equal to 0.9, the three elliptic relaxation factors

equal to 1.3, 1.6, and 1.9 for the coarse, medium, and fine grids respectively. The pseudo-time term will be omitted from the differential equation ($\epsilon = 0.$).

For the second case, all of the input parameters will be identical to the first except that the Mach number will be equal to 0.72 and the measured lift coefficient equal to 0.195.

Thus, the input data cards would read (note that with a NAMELIST format, input variable sequencing is arbitrary):

CARD NO. 1 - RUN TITLE CARD

COLUMN NO.	1	80
	SAMPLE CALCULATION NO. 1	

CARD NO. 2

COLUMN NO.	2	6	80
	\$INP EMACH=.7,F=.07,ALPHA=1.5,H=6.5,DELTA=.1,BCFOIL=1,		

CARD NO. 3

COLUMN NO.	2	80
	KUTTA=.F.,CLSET=.187,MAXIT=250,IPRTER=5,CVERGE=.0001,	

CARD NO. 4

COLUMN NO.	2	80
	PRTFLO=3,PSAVE=.T.,SIMDEF=2,WCIRC=.9,WE(1)=1.3,WE(2)=1.6,	

CARD NO. 5

COLUMN NO.	2	80
	WE(3)=1.9,EPS=0.,	

CARD NO. 6

COLUMN NO.	2	80
	\$END	

CARD NO. 7 - RUN TITLE CARD

COLUMN NO.	1	80
	SAMPLE CALCULATION NO. 2	

CARD NO. 8

COLUMN NO.	2	6	80
	\$INP EMACH=.72,CLSET=.195,		

CARD NO. 9

COLUMN NO.	2	80
	\$END	

CARD NO. 10

COLUMN NO.	1	80
	FINISHED	

We comment that the last card in the input data, following the final data card for the last case, must contain the word FINISHED in the first eight columns. This signals the program that there are no further cases to calculate and that the program is to stop.

If for the above cases the user wished to supply his own x- and y-grids, this would have been done by setting the grid input controls XGRDIN and YGRDIN equal to TRUE, specifying in the input the integers IMAXI and JMAXI, subject to the restriction $IMAXI, JMAXI \leq 100$, and then identifying all of the IMAXI values of XIN(I) and JMAXI values of YIN(I) in the input data. For example, say that the user has an 81x60 (x,y) grid. Then, the card data to be included in the input for the first case would be (using some fictitious values for XIN, YIN):

COLUMN NO.	2	80
	XGRDIN=.T.,IMAXI=81,XIN(1)=-3.0,XIN(2)=2.9,...	

⋮ etc.

COLUMN NO.	2	80
	XIN(80)=1.9,XIN(81)=2.0,YGRDIN=.T.,JMAXI=60,YIN(1)=-6.5,	

⋮ etc.

COLUMN NO.	2	80
	YIN(59)=6.4,YIN(60)=6.5,	

A similar procedure would apply if the user wished to supply his own description of the airfoil surface; that is, arrays XL(I), YL(I), XU(I), and YU(I).

4.3 Error and Information Messages

The error and information messages printed by the program are listed in this section together with a brief explanation of why they occur and what to do when they are encountered. The first group of messages (1 to 12) is concerned with input quantities while the second group (13 to 21) occur during the execution of the program and involve certain limitations inherent in the methods used in this program.

(1) IMAX OR JMAX IS GREATER THAN 100, NOT ALLOWED

This message indicates that the input mesh size limitation IMAX, $JMAX \leq 100$ has been violated.

(2) X MESH POINTS NOT MONOTONIC INCREASING

This message indicates an error in the input x-mesh that violates the condition $X_i > X_{i-1}$.

(3) Y MESH POINTS NOT MONOTONIC INCREASING

This message indicates an error in the input y-mesh that violates the condition $Y_j > Y_{j-1}$.

(4) MACH NUMBER NOT IN PERMITTED RANGE (0.5,2.0)

This message indicates that the input free-stream Mach number M_∞ is outside the nominally accepted transonic regime. A scheme solving the linear small-disturbance equation, rather than the nonlinear transonic equation solved in this program, should be employed.

(5) ALPHA NOT IN PERMITTED RANGE (-9.0,9.0)

This message indicates that the input angle of attack is beyond the range in which small-disturbance theory is considered valid; that is, $-9^\circ < \alpha < 9^\circ$.

(6) DELTA NOT IN PERMITTED RANGE (0.0,1.0)

This message indicates that the restriction on thickness ratio, $0. < \delta < 1.$, has been violated.

(7) AK = 0. VALUE OF AK MUST BE INPUT SINCE PHYS = .F.

This message indicates that the transonic similarity parameter $K = (1 - M_\infty^2)/M_\infty^{2m} \delta^{2/3}$ has not been input as is required when transonic rather than physical scaling; that is, PHYS = .F., is used in the input/output.

(8) MACH NUMBER TOO CLOSE TO 1., NOT ALLOWED

This message indicates that the input free-stream Mach number M_∞ is too close to one; that is, $|M_\infty - 1| < 0.001$. This is unallowed since the program incorporates a representation of the far field solution containing the factor $1/\sqrt{K} \sim 1/|M_\infty^2 - 1|^{1/2}$. To correct this a special sonic free-stream far field solution would be required.

(9) ABNORMAL STOP IN SUBROUTINE BCEND

BCTYPE = 6 IS NOT USABLE

This message indicates to the user that the general tunnel wall boundary condition BCTYPE = 6 is an inactive option for which conditions have not yet been worked out.

(10) ABNORMAL STOP IN SUBROUTINE BODY

BCFOIL = 4 IS NOT USABLE

This message indicates to the user that the airfoil geometry specification BCFOIL = 4 is an inactive option requiring user definition. For details on necessary user input, see section 4.2.2.

(11) ABNORMAL STOP IN SUBROUTINE FARFLD

BCTYPE = 6 IS NOT USABLE

See message (9).

(12) ABNORMAL STOP IN SUBROUTINE SCALE

SIMDEF = 4 IS NOT USABLE

This message indicates to the user that the transonic similarity specification SIMDEF = 4 is an inactive option requiring user definition. For details on necessary user input, see section 4.2.2.

(13) NOTE - ONE OR MORE SHOCKS EXTEND OUTSIDE OF CONTOUR

CDWAVE DOES NOT EQUAL TOTAL WAVE DRAG

This message indicates that some of the shock waves present for the case at hand extend beyond the appropriate drag contour incorporated in the program, so that the wave drag calculated by the program does not equal the total wave drag. For details on the drag contours available, see the description of subroutine CDCOLE in section 5.2.

(14) SHOCK WAVE ATTACHED TO BODY. MOMENTUM INTEGRAL CANNOT BE DONE. DRAG OBTAINED FROM SURFACE PRESSURE INTEGRAL.

This message indicates that, for reasons explained in section 5.2 under subroutine CDCOLE, the momentum integral method for calculating drag cannot be used. The drag is then calculated by integrating the surface pressures on the airfoil.

(15) * * * * * CAUTION * * * * *
SOLUTION MAY BE INVALID
DETACHED SHOCK WAVE IS TOO CLOSE TO UPSTREAM BOUNDARY.
MOMENTUM INTEGRAL CANNOT BE DONE. DRAG OBTAINED FROM
SURFACE PRESSURE INTEGRAL.

This message indicates that the bow shock is too close (within three mesh points) of the upstream boundary for the momentum integral along the bow shock to be performed, and that the drag is obtained by integrating the surface pressures on the airfoil. Also, because the bow shock is so close to the upstream boundary, the solution may be inaccurate and it is recommended that the upstream boundary be displaced further from the airfoil and the calculation repeated.

(16) THE INPUT MESH CANNOT BE ADJUSTED FOR CUTOFF, BECAUSE
IMAX OR JMAX IS TOO CLOSE TO THE LIMIT OF 100.

This message is written by subroutine CKMESH and indicates that, because the input mesh requirements for grid cutting described in section 4.2.2 under Mesh Refinement Option were not met, subroutine CKMESH has attempted to correct this by adding points. However, either the input X or Y grid is too close to the limit of 100 to allow this, with the result that no points are added, the mesh refinement option is bypassed, and the input mesh is used to convergence.

(17) ABNORMAL STOP IN SUBROUTINE DROOTS. (0)
NONCONVERGENCE OF ITERATION FOR ALPHA (1)
(2)

This message identifies one of the constants ALPHA0, ALPHA1, or ALPHA2 required in the subsonic far-field doublet solution in an ideal slotted wind tunnel, and indicates that the iterative solution for that constant has failed to converge. This error message should not normally occur. If it does, the input values for the tunnel slot parameter F and wall porosity factor P should be checked. See subroutine DROOTS in section 5.2 for details.

(18) * * * * * CAUTION * * * * *

SONIC LINE HAS REACHED A BOUNDARY
THIS VIOLATES ASSUMPTIONS USED TO DERIVE BOUNDARY CONDITIONS.
SOLUTION IS PROBABLY INVALID

This message indicates that the sonic line has extended out to the top and/or bottom grid boundaries, and that the solution is probably invalid since the assumptions under which the boundary conditions were derived have been violated. For free-stream flows, the difficulty can be remedied by moving the lateral boundaries out further, while for tunnel simulation calculations, this condition indicates that the flow has become choked. For the latter case, the details have not yet been worked out.

(19) * * * * * CAUTION * * * * *

NUMBER OF SONIC POINTS EXCEEDED 200. ARRAY DIMENSION
EXCEEDED. EXECUTION OF SUBROUTINE MILINE TERMINATED

This message indicated that the number of sonic points has exceeded 200 and subroutine MILINE, which determines the locations of the sonic lines, has terminated calculation. For the maximum grid size (100x100) allowed in this program, this error message should not occur. If it does, the remainder of the output should be checked for unusual behavior.

(20) * * * * * CAUTION * * * * *

MAXIMUM MACH NUMBER EXCEEDS 1.3
SHOCK JUMPS IN ERROR IF UPSTREAM NORMAL MACH NUMBER GREATER
THAN 1.3

This message is written by subroutine PRINT1 and indicates that the maximum Mach number along the dividing streamline, which includes the airfoil upper and lower surface, has exceeded 1.3. Consequently, if the normal Mach number just upstream of the shock wave is greater than 1.3 (this can be checked from the output of PRINT1) the shock jumps, as calculated by the program using isentropic small-disturbance theory, will be in error with the true Rankine-Hugoniot value since the usually accepted upper bound for validity of the isentropic assumption has been exceeded.

(21) ABNORMAL STOP IN SUBROUTINE VROOTS (0)
NONCONVERGENCE OF ITERATION FOR BETA(1)
(2)

This message identifies one of the constants BETA0, BETA1, or BETA2 required in the subsonic far-field vortex solution in an ideal slotted wind tunnel, and indicates that the iterative solution for that constant

has failed to converge. This error message should not normally occur. If it does, the input values for the tunnel slot parameter F and wall porosity factor P should be checked. See subroutine VROOTS in section 5.2 for details.

4.4 Program Output

The output generated by the program is described in this section. A general description of the format is provided in the first subsection, followed by a subsection containing several sample cases for user reference.

4.4.1 General description of the output.- The output format of the program is as follows. On the first page a header is printed giving the program name, some general information concerning the problem solved and method used, and identification of the program's authors. On the next page, all of the input cards for the entire run sequence are printed. On the first line of the third page, the information input by the user on the first run title card is printed. Next, all of the input variables described in section 4.2.1 are printed. This includes, of course, variables not specified in the input data and which then assume their default values. On the fourth page the airfoil geometry information required by the program is printed. First, a header is printed to inform the user whether the printout is in physical variables (PHYS = .T.), in which case all of the airfoil geometric characteristics are normalized by the chord; or if the printout is in transonic similarity variables (PHYS = .F.), for which the geometric characteristics are normalized by chord length times thickness ratio. Next, the maximum thickness, airfoil volume (per unit span), and maximum camber are output. Finally, the ordinates and slopes of the upper and lower surface, together with the thickness and camber distributions are provided in tabular form at the input x -mesh points. On the fifth page, intermediate output information for the coarse mesh (if a coarse grid is used) is provided. This includes the elliptic relaxation factor, the value of ϵ (EPS) used, and the maximum number of iterations for this mesh. This is followed by a heading which contains, from left to right, the iteration count (ITERATION), lift coefficient (CL), moment coefficient (CM), I and J locations (IERR,JERR) where the maximum iteration error in ϕ occurs, the absolute magnitude of the maximum iteration error in ϕ (ERROR), the I and J locations (IRL,JRL) of the maximum residual, the absolute magnitude of the maximum residual (BIGRL), the absolute magnitude of the iteration error in circulation (ERCIRC), the I location (ICPU) of

the point on the airfoil where the maximum iteration error in upper surface pressure coefficient occurs, the absolute magnitude of the maximum iteration error in upper surface pressure coefficient (CPERRU), and the corresponding results (ICPL,CPERRL) for the maximum iteration error in surface pressure coefficient. A printout of these quantities is provided at every IPRTERth iteration until the solution either converges or diverges, or the iteration limit is reached; at this point, a message is printed indicating which of these three alternatives occurred. On the sixth page, the final output for the coarse mesh is given. This consists of the final coarse-mesh values of the lift and moment coefficients and tabulated values of the pressure coefficient and local Mach number (or local similarity parameter if PHYS = .F.) along the entire length of the dividing ($y = 0$) streamline, including both the upper and lower surface of the airfoil. In addition, the x - and y -coarse-mesh ordinates are also provided. The format is as follows. First, the lift and moment coefficients are given, together with the value of the critical pressure coefficient. Next, a header is printed which contains, from left to right, the I index of the current x -mesh point (I), the x value of the mesh point (X), the pressure coefficient (CP), and local Mach number ($M1$) just below the dividing streamline with the additional title ($\begin{matrix} \text{LOWER} \\ y = 0- \end{matrix}$), and then the pressure coefficient (CP) and local Mach number ($M1$) just above the dividing streamline with the additional title ($\begin{matrix} \text{UPPER} \\ y = 0+ \end{matrix}$). Tabulated values of these quantities follow and are provided for each x -mesh point of the coarse grid. Locations of the airfoil leading and trailing edges are indicated in the output. Alongside the table of values, a concomitant printer plot of the tabulated pressure coefficients is given. Finally, all the y -values of the coarse grid y mesh are printed. On the seventh and eighth pages, the intermediate and final output, respectively, for the medium grid (if a medium grid is used) are provided in the same format as described above for the coarse grid. On the eight page, the intermediate output for the fine mesh is given. On the following page, the final print output begins for the fine mesh. On this page, information is provided informing the user whether the printout is in physical or transonic variables, and if it is in physical variables which particular definition of similarity parameters was used. Also indicated is the type of boundary condition specified (free air or type of wind tunnel), whether the difference equations were in conservative form or not, and whether the Kutta condition was enforced or the lift coefficient specified. Next,

some limited airfoil geometry and flow field information is given (i.e., free-stream Mach number and airfoil thickness ratio if physical scaling is used, (PHYS = .T.), plus angle of attack and transonic similarity parameter in all cases, and far-field doublet strength and airfoil volume per unit span when $M_\infty < 1$. Finally, if PHYS = .T., the transonic scaling parameters used by the program to scale the pressure, lift, drag, and moment coefficient, the y-ordinate, and the flow angle are printed. On the next page, the fine grid lift and moment coefficients, pressure coefficients and local Mach number on the dividing streamline together with the x- and y-fine mesh ordinates, and a printer plot of the pressure coefficients is given in the format previously described for the coarse grid. On the next page, a flow character map is provided. Each point in the mesh is characterized according to whether it is a parabolic point (P), hyperbolic point (H), shock point (S), or an elliptic point (-). The following page provides a Mach number map of the flow field. Each grid point receives either a number or letter letter designation according to the symbol table provided in the plot. On the next page, a laterally expanded printer plot of the pressure coefficients on the dividing streamline and airfoil surface is provided. For wind-tunnel wall simulation calculation (i.e., BCTYPE \neq 1 or 3), tabulated values of the pressure coefficient and flow angle along the entire length of the wind-tunnel walls ($y = H\pm$) are given next. The format is similar to that provided for output of the pressure distributing along the dividing streamline discussed previously. This consists of tabulated values of the pressure coefficient (CP) and the flow angle (THETA) along the bottom ($\begin{matrix} \text{LOWER} \\ y = -H \end{matrix}$ and top ($\begin{matrix} \text{UPPER} \\ y = +H \end{matrix}$) surfaces of the wind tunnel at each of the fine mesh x-grid (X) locations. A printer plot of the pressure coefficients is also provided alongside the tabulated values. On the next two pages, output describing the sonic lines is provided. The first page gives tabulated values of the sonic line coordinates, while the second page provides a printer plot of the sonic lines throughout the flow field. On the following page, flow field output, if requested by the user, is printed. This consists of tabulated values of the pressure coefficient, flow angle, and local Mach number (or local similarity parameter if PHYS = .F.) along the entire length of y = constant lines on the fine mesh. The format is as follows. First, a header is printed which contains, in the first two columns, the I index of the current x-mesh point (I) and the x value of the x-mesh point (X) The next nine columns contain three sets of pressure coefficients (CP), flow angles (THETA), and local Mach numbers (M1) along three different y = constant rows, with an additional title above each of the sets indicating both the

J index and associated y-value of the row. Tabulated values of these quantities follow for each x-mesh point of the fine grid. If the flow field print option PRTFLO = 3 was chosen, the flow field output is confined to three J lines about the location of maximum iteration error in ϕ , and all the flow field output is contained across one page. However, if the entire flow field is to be printed (PRTFLO = 2), then the above format is repeated for three successive rows at a time until the entire field is covered. Following this, the shock wave drag profiles and total pressure loss profiles are given. For each shock wave within the momentum contour, the total wave drag is given together with tabulated values of the shock wave drag ($CD(y)$) and the total pressure ($PO/POINF$) as functions of the lateral coordinate (y). On the next page, information regarding the drag coefficient is given. If the drag was calculated by the momentum integral method, the locations of the contour boundaries are given together with the individual contributions to the drag from the integrations of the drag integral along each of these boundaries. In addition, information regarding the number of shock waves inside the contour and the total wave drag are printed. Should any of the shocks extend beyond the momentum contour, a message is printed indicating this and the fact that the wave drag calculated does not equal the total wave drag. Finally, the total of all the separate drag contributions is given. If the momentum integral method could not be used to calculate the drag--either because the bow shock was attached or too close to the nose or because the bow shock was too close to the upstream boundary--a message to that effect is printed and the drag is then calculated by an integration of surface pressures. The time to run the case in seconds is provided on the next page. This would conclude the output for an individual case. The above format is repeated until all cases are run. Following this, a final page of output is printed which consists simply of the word FINISHED, indicating a normal exit from the program.

4.4.2 Sample cases.-- In order to provide reference checks on the programs, six sample test cases have been run exercising a representative sample of the various options available. The results are provided in figures 4-4 through 4-9. In each case the input data are provided together with the corresponding output. Because of the large amount of output information, full program output is provided only for the first two cases (figs. 4-4,4-5); however, the abbreviated output given for the remaining cases is sufficient to verify the correctness of the program under those options selected.

4.5 Comparisons with Data and Other Methods

A variety of comparisons of the results predicted by the methods employed in this program have been presented in the literature (refs. 1, 2, 3, 6, 7, 8, 9, and 20). For convenience, several of those comparisons are briefly repeated here in order to provide the user both with a general assessment of the accuracy of the methods used and some idea of the typical solution behavior that can be expected. For further information, the user is referred to the excellent review article by Bailey (ref. 21).

Basically, the comparisons presented here fall into three categories; that is, (1) comparisons with exact (hodograph) inviscid solutions, (2) comparisons with more accurate (full inviscid, full potential) numerical solutions, and (3) comparisons with data. The first two categories serve to assess the accuracy of the numerical technique and the small-disturbance potential approximation, while the third serves to demonstrate the influence of tunnel interference and viscous effects.

In figure 4-10 a comparison is made of the small-disturbance relaxation solutions calculated by the Krupp-Murman techniques (ref. 3) employed in this program with the exact hodograph solutions (ref. 22) for a NLR quasi-elliptical airfoil at $\alpha = 1.32^\circ$, $M_\infty = 0.7557$. A similar result is shown in figure 4-11 for a Garabedian-Korn airfoil at $\alpha = 0.12^\circ$, $M_\infty = 0.76$. Also provided in those two figures are relaxation solutions (refs. 23 and 24) for the full potential equation. We note that in both of those cases, the small-disturbance relaxation solutions and the full potential relaxation solutions show a weak shock as compared to the exact hodograph solution, which is for the shock-free design condition. For the Garabedian-Korn airfoil, the small-disturbance solution also predicts a somewhat smaller expansion beyond the nose. The overall agreement, however, between the small-disturbance solutions and the exact and full potential solutions is quite good, particularly in light of the fact that flows past such shock-free airfoils are extremely sensitive to small changes in geometry and flow conditions, and therefore provide an exacting test of the accuracy of any transonic predictive method. Figure 4-12 shows a further comparison of the small-disturbance results predicted by the Krupp-Murman method and a relaxation solution of the full inviscid equations for a subcritical flow (ref. 25). Again, the agreement is very good, even at this low ($M_\infty = 0.63$) a Mach number.

The results presented in the preceding three figures are for shock free flows. For flows containing shock waves, the relaxation methods used above for the full equations do not employ conservative difference operators and consequently do not satisfy the proper shock jump conditions. Some idea of the error involved in solving the exact isentropic equation in nonconservative form (ref. 26) and the calculation of the isentropic solution using a conservative time-dependent procedure (ref. 27) is given in figure 4-13. Also provided is the small-disturbance result predicted by the present program when using the fully conservative (FCR) option. We note that the shock location predicted by the two conservative calculations is essentially in agreement, while the nonconservative result predicts a weaker shock displaced forward of the conservative location. We note that the differences in shock jump values observed above between the two conservative calculations is not due to an accuracy problem in the above computations but is attributable to the difference in pressure jump between the isentropic approximation to the Rankine-Hugoniot relations, which is the jump condition satisfied in the fully conservative time-dependent calculation, and the small-disturbance approximation to the R-H conditions, which is the corresponding shock jump condition satisfied when using the conservative (FCR) option in this program.

Comparisons between the fully conservative (FCR) and not fully conservative (NCR) small-disturbance solutions predicted by the present finite-difference scheme are of interest in order to assess the differences between such calculations. Such results have been discussed previously in reference 19, and figures 4-14 and 4-15 summarize the essential conclusions of that study and provide a further comparison with a more exact result. Figure 4-14 compares the FCR and NCR solutions for the similarity surface pressure distributions for a subsonic free-stream supercritical flow as the finite-difference grid is refined. We note that while the FCR solution approaches the theoretical, small-disturbance shock jump, the NCR solution does not. Figure 4-15 shows the corresponding result for a supersonic free stream and a detached bow wave, and also includes the result of a conservative time-dependent calculation for the exact isentropic equations (ref. 28). While the FCR and the time-dependent method produce essentially the same result, the NCR method yields too great a detachment distance. Furthermore, it was found (ref. 19) that the NCR method predicts shock jumps which are considerably in error for strong oblique shocks, although the results for normal and weak oblique shocks remain reasonable. We note,

as pointed out previously (refs. 1 and 21), in many cases of $M_\infty < 1$ transonic flows, the NCR predicted surface pressure distributions are usually in better agreement with experimental data than the FCR results, due to boundary-layer weakening of the shock wave. Unfortunately, this agreement is clearly fortuitous, and because the NCR method does not yield a unique solution nor account for viscous effects in any rational manner, it is not recommended for general use.

In the absence of shock wave-boundary layer separation, it has been suggested (refs. 19 and 29) that the principal viscous effect to account for is the reduction in circulation from the inviscid value determined by the Kutta condition. This effect is illustrated in figure 4-16 where the data from reference 15 is compared with the two FCR small-disturbance solutions found by: (a) specifying the Kutta condition and (b) fixing the circulation at the value determined by the experimental lift. Considerably better agreement with data results in the latter case except, of course, for points near the trailing edge.

Finally, some indication of the effects of wind-tunnel interference at transonic speeds is given in figures 4-17 and 4-18. Figure 4-17 shows the FCR solutions for an NACA 0012 airfoil at $\alpha = 0^\circ$ in free air and for an ideal slotted wall compared to the data taken in the RAE slotted wind tunnel (ref. 30). A forward movement of the shock due to tunnel interference is indicated. This effect is again shown by the results presented in figure 4-18, where data taken from the NASA/Ames Research Center 2- by 2-Foot Wind Tunnel (ref. 1) is compared with the FCR solutions for free air, an ideal slotted tunnel, and an ideal slotted tunnel with the lift fixed at the experimental value. In this example, the effects of wind-tunnel interference on lift and shock position are of the same magnitude as the viscous effects as evident by their respective changes from the free air to the slotted tunnel Kutta condition result (interference effect), and from the slotted tunnel Kutta condition result to the slotted tunnel fixed C_L result (viscous effect).

4.6 Program Run Times

The development of the structure of the present code as well as the actual debugging process have been carried out on a Control Data Corporation 7600 computer. Therefore, the majority of run time information and experience is for that machine. However, the most crucial time information required by the general user is simply the CPU time required for one

iteration sweep of the flow field on a typical fine grid, since this will provide directly an estimate of the maximum run time required for a particular case. The resolution of the default free air grid (77x56) and default tunnel grid (77x48) used in the program is generally sufficient for most engineering applications, so that the time per iteration sweep on these grids should provide a fairly general and useful result. We have determined that the sweep time per iteration on these grids using the FTN, OPT = 2 compiler on the CDC 7600 is approximately 0.04 sec/sweep. Since the medium and coarse meshes contain approximately one-fourth and one-sixteenth as many points, the approximate maximum CPU execution time per case when employing the grid refinement option is

$$\begin{aligned} \text{Maximum CPU Execution Time} &\doteq \text{MAXIT} \cdot (1 + 1/4 + 1/16) \cdot (0.04) \text{ sec} \\ (\text{CDC 7600; FTN, OPT} = 2) & \\ &\doteq 0.0525 \cdot \text{MAXIT} \text{ sec} \end{aligned}$$

The analogous result for the CDC 6600 is

$$\begin{aligned} \text{Maximum CPU Execution Time} &\doteq 0.310 \text{ MAXIT sec} \\ (\text{CDC 6600; FTN, OPT} = 2) & \end{aligned}$$

We note that on the analytical mesh (81x64), the corresponding CDC 7600 time is approximately 0.055 sec/sweep.

With regard to improving run times, numerical experimentation has shown that by using the mesh refinement option execution times can be reduced by 30 percent to 60 percent in most cases, in comparison to the same calculation started and carried to completion on the fine grid alone. In addition, it was found that a number of cases which diverged when started on the fine mesh, converged smoothly on the final mesh when the mesh refinement option was used. Consequently, the additional programming required for this option appears well worthwhile in terms of enhanced stability and computational time. Finally, we note that a number of proven convergence acceleration techniques, such as the extrapolation methods used by Hafez and Cheng (ref. 31) have not yet been incorporated into the present version of the code. Future use of these improvements, as well as others currently under development--such as direct or semidirect solvers (refs. 32 and 33) and shock fitting (ref. 34)--can undoubtedly augment the methods used in the current program.

4.7 Machine Compatability

The present code has been written in FORTRAN and primarily tested out on a Control Data Corporation 7600 using the FTN, OPT = 2 compiler. Care has been taken in the programming to insure that the code can be easily adapted for use on other machines. All header output statements are written in the universal Hollerith format. Double precision arithmetic is not used. Storage requirements have been reduced as much as possible by the liberal use of common blocks to pass information internally. The code has been successfully check tested on a CDC 6600 and required no changes.

	k	n	m
Cole	0	0	0
Spreiter	2	2/3	2/3
Krupp	7/4	3/4	1/2

Table 4-I.- Exponents for transonic similarity parameters.

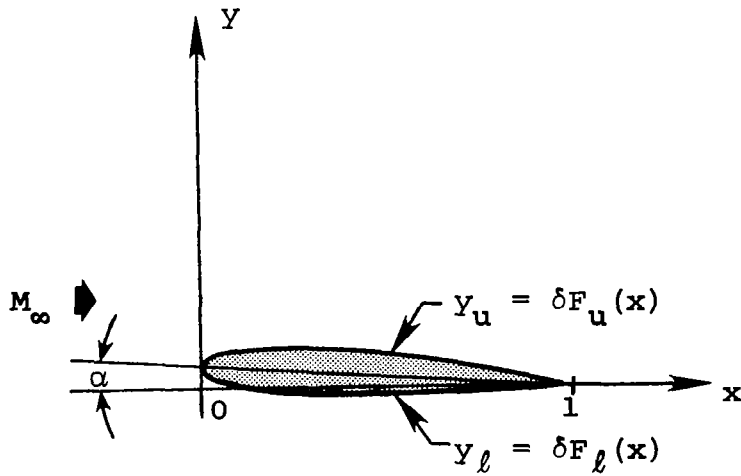


Figure 4-1.- Coordinate system and airfoil geometry.

$$\begin{array}{c}
 \phi_{ff} \\
 \hline
 \phi_{ff} \quad \left[\phi_x \right] = \left[\phi_{\tilde{y}} \right] = 0 \\
 \hline
 \phi_{ff} \quad \left[\phi \right] = \Gamma \\
 \hline
 \phi_{ff} = \phi_{\text{far field}}
 \end{array}$$

$\phi_{ff} = \phi_{\text{far field}}$

(a) Subsonic free stream ($M_\infty < 1$).

$$\begin{array}{c}
 \phi_{\tilde{y}} = + \sqrt{K} \phi_x \\
 \hline
 \phi_x = 0 \quad \left[\phi_x \right] = \left[\phi_{\tilde{y}} \right] = 0 \\
 \hline
 \phi_{\tilde{y}} = 0 \quad \left[\phi \right] = \Gamma \\
 \hline
 \phi_{\tilde{y}} = - \sqrt{K} \phi_x
 \end{array}$$

(b) Supersonic free stream ($M_\infty > 1$).

Figure 4-2.- Summary of boundary conditions for free air flows.

$$\phi_x + F\tilde{H}\phi_{x\tilde{y}} + \frac{1}{\tilde{p}}\phi_{\tilde{y}} = 0$$

$$\tilde{y} = \tilde{H}$$

$$\frac{\phi_{\tilde{y}}}{\tilde{y}} = \frac{dF_u}{dx} - A \quad [\phi_x] = [\phi_{\tilde{y}}] = 0$$

$$\frac{\phi_{\tilde{y}}}{\tilde{y}} = \frac{dF_l}{dx} - A \quad [\phi] = \bar{\Gamma}$$

$$\tilde{y} = -\tilde{H}$$

$$\phi_x - F\tilde{H}\phi_{x\tilde{y}} - \frac{1}{\tilde{p}}\phi_{\tilde{y}} = 0$$

(a) Subsonic free stream ($M_\infty < 1$).

$$\phi_x + F\tilde{H}\phi_{x\tilde{y}} + \frac{1}{\tilde{p}}\phi_{\tilde{y}} = 0$$

$$\tilde{y} = \tilde{H}$$

$$\frac{\phi_{\tilde{y}}}{\tilde{y}} = \frac{dF_u}{dx} - A \quad [\phi_x] = [\phi_{\tilde{y}}] = 0$$

$$\frac{\phi_{\tilde{y}}}{\tilde{y}} = \frac{dF_l}{dx} - A \quad [\phi] = \bar{\Gamma}$$

$$\tilde{y} = -\tilde{H}$$

$$\phi_x - F\tilde{H}\phi_{x\tilde{y}} - \frac{1}{\tilde{p}}\phi_{\tilde{y}} = 0$$

(b) Supersonic free stream ($M_\infty > 1$).

Figure 4-3.- Summary of boundary conditions for wind-tunnel simulations.

SAMPLE CASE 1 - PROGRAM DEFAULT OPTION - KORN AIRFOIL
 \$IMP
 \$END

(a) Input.

SAMPLE CASE 1 - PROGRAM DEFAULT OPTION - KORN AIRFOIL

INPUT PARAMETERS							

EMACH = .75000	POR = 0.00000	IMIN = 1	BCTYPE = 1	AMESH = F			
DELTA = .11500	CLSET = 0.00000	IMAXI = 77	BCFOIL = 3	PHYS = T			
ALPHA = .12000	EPS = .20000	JMIN = 1	PSTART = 1	PSAVE = F			
AK = 0.00000	RIGF = 0.00000	JMAXI = 56	PRTFLO = 1	KUTTA = T			
GAM = 1.40000	WCIRC = 1.00000	MAXIT = 500	IPRTER = 10	FCR = T			
F = 0.00000	CVERGE = .00001	NU = 100	SIMDEF = 3				
H = 0.00000	DVERGE = 10.0	NL = 75	ICUT = 2				
WE = 1.00000, 1.90000, 1.95							
XIN							
-1.075000	-.950000	-.825000	-.700000	-.575000	-.450000	-.325000	-.200000
-1.175000	-.125000	-.075000	-.052500	-.035000	-.022500	-.015000	-.007500
-.002500	.002500	.007500	.012500	.017500	.022500	.027500	.032500
.037500	.045000	.055000	.065000	.075000	.085000	.095000	.105000
.140625	.171875	.203125	.234375	.265625	.296875	.328125	.359375
.390625	.421875	.453125	.484375	.515625	.546875	.578125	.609375
.640625	.671875	.703125	.734375	.765625	.796875	.828125	.859375
.890000	.905000	.915000	.930000	.945000	.960000	.975000	.990000
1.000000	1.010000	1.025000	1.035000	1.045000	1.055000	1.065000	1.075000
1.400000	1.500000	1.625000	1.750000	1.875000			
YIN							
-5.200000	-4.400000	-3.600000	-3.000000	-2.400000	-1.950000	-1.600000	-1.350000
-1.150000	-.950000	-.800000	-.650000	-.550000	-.450000	-.390000	-.340000
-.300000	-.270000	-.240000	-.210000	-.180000	-.150000	-.125000	-.100000
-.075000	-.050000	-.030000	-.010000	.010000	.030000	.050000	.075000
.100000	.125000	.150000	.180000	.210000	.240000	.270000	.300000
.340000	.390000	.430000	.500000	.550000	.600000	.650000	.700000
1.350000	1.600000	1.950000	2.400000	3.000000	3.600000	4.400000	5.200000
XU							
.000008	.000167	.000331	.000799	.001477	.002153	.003331	.005336
.008648	.014583	.023461	.033891	.046887	.053973	.076921	.108450
.059966	.061445	.062909	.065925	.068785	.071482	.074007	.075322
.076603	.077862	.079112	.080445	.081819	.083269	.084841	.086702
.088848	.091378	.094413	.098308	.103104	.109010	.118244	.125292
.136635	.150037	.165853	.184499	.195177	.206361	.218244	.230813
.244047	.257917	.272371	.287410	.302990	.319057	.335355	.352421
.369591	.386995	.404133	.421391	.438708	.456013	.473246	.491343
.507242	.523881	.539536	.554867	.569823	.584351	.598405	.611930
.624904	.637273	.648435	.659016	.669987	.678321	.687112	.695300
.706936	.723406	.738649	.761390	.777010	.792241	.809008	.824992
.836953	.857188	.875621	.893268	.913086	.927686	.939004	.952002
.971789	.989100	.997860	1.000000				
YU							
.000787	.003392	.004538	.006137	.007683	.009056	.010575	.012013
.015607	.019624	.024441	.029035	.033198	.035966	.038637	.041277
.037700	.038103	.038497	.039276	.039986	.040625	.041195	.041833
.041756	.042019	.042274	.042539	.042804	.043079	.043368	.043700
.044072	.044497	.044989	.045595	.046312	.047156	.048132	.049304
.050626	.052089	.053663	.055351	.056210	.057068	.057918	.058751
.059559	.060335	.061068	.061751	.062381	.062947	.063445	.063867
.064213	.064473	.064646	.064733	.064735	.064651	.064487	.064248
.063871	.063438	.062945	.062376	.061731	.061014	.060232	.059389
.058496	.057562	.056650	.055721	.054791	.053867	.052965	.052180
.050722	.048045	.046680	.044341	.041053	.038606	.035768	.032428
.030775	.026954	.023361	.018848	.015750	.012954	.010067	.006213
.004559	.001620	.002293	0.000000				

Figure 4-4.- Sample test case 1 - program default option - subsonic, free air simulation (full output): Korn airfoil, $\delta_n = 0.115$, $\alpha = 0.15^\circ$, $M_\infty = 0.75$.

XL							
0.000000	.000012	.000043	.000183	.000249	.000348	.000455	.000680
.001011	.001481	.001875	.002316	.002755	.003201	.003647	.004179
.007035	.008265	.009969	.012286	.015346	.019276	.024035	.029379
.039095	.052516	.062469	.073329	.085290	.099822	.118663	.140987
.167184	.202933	.228511	.247895	.263995	.282047	.297745	.311147
.324075	.344872	.363522	.387544	.404492	.426308	.450316	.475370
.521837	.549843	.578612	.605305	.623479	.642152	.657943	.671212
.690340	.708891	.726684	.746683	.768502	.784892	.801149	.819187
.838548	.858817	.879431	.903723	.926504	.943652	.955663	.973623
.986187	.996582	1.000000					
YL							
0.000000	-.000700	-.001385	-.002168	-.003330	-.003880	-.004479	-.005199
-.006133	-.007183	-.007933	-.008675	-.009776	-.011204	-.012405	-.012861
-.013983	-.014962	-.016175	-.017636	-.019336	-.021288	-.023496	-.025937
-.028634	-.032423	-.034840	-.037182	-.039456	-.041862	-.044483	-.047317
-.049298	-.051443	-.052406	-.052859	-.053062	-.053117	-.053027	-.052849
-.052562	-.051951	-.051218	-.050013	-.049004	-.047495	-.045671	-.043288
-.038336	-.034916	-.031104	-.027333	-.024661	-.021854	-.019317	-.017249
-.014527	-.011771	-.009228	-.006937	-.003868	-.002080	-.000524	-.000351
.002227	.003224	.003885	.004212	.004067	.003657	.003067	.002242
.001329	.000376	0.000000					

AINFOIL GEOMETRY OUTPUT

PRINTOUT IN PHYSICAL VARIABLES NORMALIZED BY CHORD LENGTH
AIRFOIL VOLUME= .07556017

MAX THICKNESS = .11569133
MAX CAMBER = .1154307

INPUT GRID X	UPPER SURFACE Y	UPPER SURFACE DY/DX	LOWER SURFACE Y	LOWER SURFACE DY/DX	THICKNESS	CAMBER
.00250000	.00959171	1.45523113	-.50895517	-1.54389477	.71927844	.00000000
.00750000	.01470383	.01267528	-.01430528	-.00527524	.31453455	.00000000
.01250000	.01831605	.05037444	-.11776258	-.08763322	.01803922	.00027064
.01750000	.02132582	.56142563	-.02741959	-.40511405	.02187270	.00043312
.02250000	.02395903	.49640664	-.02267782	-.42226950	.02331842	.00040000
.02750000	.02631987	.44556493	-.02467419	-.47871654	.02549773	.00022204
.03250000	.02846690	.41150450	-.02648065	-.34492811	.02747482	.00049416
.03750000	.03144665	.37490980	-.02813392	-.31735008	.02929028	.00110030
.04250000	.03312592	.33599923	-.03036772	-.28473871	.03175682	.00130940
.04750000	.03627290	.29719151	-.03300553	-.24997457	.03466322	.00109968
.05250000	.03904311	.25543566	-.03541151	-.22262497	.03722031	.00131480
.05750000	.04141318	.21765425	-.03751777	-.19049957	.03946048	.00149769
.06250000	.04239674	.18054845	-.03940423	-.17878220	.04140049	.00149028
.06750000	.04547125	.15571564	-.04155113	-.15725689	.04348617	.00149007
.11500000	.04796777	.13267017	-.04402275	-.13157240	.04599526	.00149050
.14062500	.05117517	-.1107138	-.04693157	-.1153417	.04902792	.00244725
.17187500	.05422336	.09136735	-.04940890	-.07604901	.05193110	.00222271
.20312500	.05682572	.07560296	-.05140188	-.04017982	.05413880	.0021052
.23437500	.05897522	.06223596	-.05256572	-.02573214	.05577137	.00232580
.26562500	.06073549	.05053493	-.05307432	-.00710822	.05694491	.00230359
.29687500	.06214320	.03994811	-.05302871	.00999124	.05788500	.00240724
.32812500	.06323300	.02994726	-.05246013	.02617109	.05784500	.00253853
.35937500	.06401751	.02066257	-.05130534	.04194706	.05770642	.00231199
.39062500	.06451653	.01150033	-.04984448	.05725378	.05710051	.00230602
.42187500	.06473420	.00241416	-.04781989	.07246666	.05627715	.00240715
.45312500	.06467130	-.00050335	-.04533303	.08089782	.05500219	.00249011
.48437500	.06421853	-.01594853	-.04240441	.10705528	.05335947	.00249090
.51562500	.06368431	-.02063701	-.03905118	.11389338	.05135775	.00230627
.54687500	.06268379	-.04090603	-.03529278	.12682358	.04898827	.00150908
.57812500	.06133234	-.04979547	-.03117483	.13715976	.04625159	.00150879
.60937500	.05955542	-.06436061	-.02674112	.14591750	.04314827	.00160073
.64062500	.05724524	-.08146377	-.02220895	.15119319	.03969010	.00170315
.67187500	.05451546	-.09802562	-.01732781	.15260505	.03591913	.001809132
.70312500	.05117116	-.11046994	-.01261849	.14780410	.03189432	.00192003
.73437500	.04725652	-.13370925	-.00815935	.13883154	.02771293	.001954359
.76562500	.04280623	-.15103101	-.00420143	.11693724	.02350383	.00190224
.79687500	.03783843	-.16679849	-.00091374	.09298191	.01937609	.001846235
.82812500	.03239201	-.18132023	.00157884	.06659416	.01544658	.00180542
.85937500	.02653266	-.19298838	.00324655	.04018931	.01164305	.001740931
.89062500	.02154184	-.19909862	.00400325	.01893405	.00874929	.00170254
.90000000	.01850618	-.20087884	.00419381	.00644729	.00712319	.00160471
.91500000	.01548642	-.20052560	.00419572	.00238834	.00565535	.001498157
.93000000	.01249528	-.19797396	.00400511	-.01924820	.00424508	.00125019
.94500000	.00955711	-.19341345	.00361309	-.03322763	.00297211	.00102800
.96000000	.00676666	-.18611192	.00300379	.04814189	.00185143	.00060298
.97500000	.00399227	-.17519895	.00215209	.06622514	.00092000	.00037208
.99000000	.00147612	-.15923679	.00109946	.08737525	.00023333	.000124279
1.00000000	0.00000000	-.13691589	0.00000000	-.11000585	0.00000000	0.00000000

Figure 4-4.- Continued.

INTERMEDIATE OUTPUT FOR COARSE MESH

WE = 1.0000 EPS = .2000 MAXIT FOR THIS MESH = 125

ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERRU	ICPL	CPERRL
10	.45126	-.16416	12	7	.4684E-01	6	7	.9957E+02	.4366E-01	12	.5703E+00	16	.5936E+00
20	.52852	-.15065	5	4	.1506E-01	6	7	.1756E+03	.7888E-02	16	.9515E-01	6	.2033E+00
30	.56601	-.15205	5	7	.7516E-02	6	7	.9360E+02	.5803E-02	8	.1324E+00	10	.3931E-01
40	.59563	-.15427	10	7	.4266E-02	6	7	.4973E+02	.2151E-02	9	.7349E-01	7	.2837E-01
50	.60322	-.15424	5	7	.2120E-02	6	7	.2546E+02	.1601E-02	9	.3169E-01	7	.1275E-01
60	.60992	-.15472	11	7	.1166E-02	6	7	.1350E+02	.7924E-03	9	.1804E-01	7	.7328E-02
70	.61343	-.15498	5	7	.6631E-03	6	7	.7928E+01	.5202E-03	9	.9775E-02	7	.2819E-02
80	.61554	-.15512	6	7	.3844E-03	6	7	.4595E+01	.2905E-03	9	.5994E-02	7	.1781E-02
90	.61678	-.15520	5	7	.2250E-03	6	7	.2686E+01	.1740E-03	9	.3547E-02	7	.1018E-02
100	.61751	-.15525	5	7	.1318E-03	6	7	.1573E+01	.1014E-03	9	.2103E-02	7	.5931E-03
110	.61793	-.15528	5	7	.7715E-04	6	7	.9203E+00	.5952E-04	9	.1241E-02	7	.3496E-03
120	.61818	-.15530	5	7	.4521E-04	6	7	.5391E+00	.3489E-04	9	.7299E-03	7	.2041E-03

***** ITERATION LIMIT REACHED *****

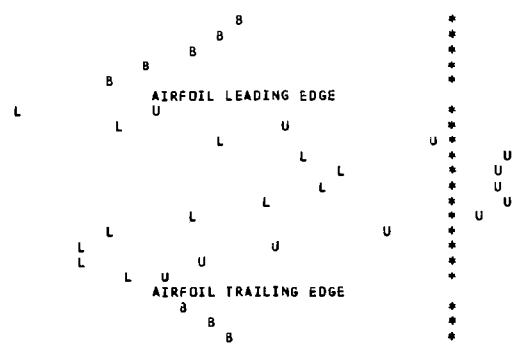
FINAL OUTPUT FOR COARSE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER (OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.

CL = .618266
CM = -.155303
CP* = -.603170

B = BEFORE OR BEHIND AIRFOIL
U = UPPER AIRFOIL SURFACE
L = LOWER AIRFOIL SURFACE
* = CRITICAL PRESSURE

I	X	LOWER Y=0-		UPPER Y=0+	
		CP	M1	CP	M1
1	-1.075000	.003088	.748505	.003088	.748505
2	-.575000	.043372	.728725	.043372	.728725
3	-.175000	.129825	.684349	.129825	.684349
4	-.035000	.264500	.608810	.264500	.608810
5	-.002500	.360806	.548448	.360806	.548448
AIRFOIL LEADING EDGE					
6	.017500	.630016	.324850	.249591	.617627
7	.037500	.337262	.563801	-.143732	.816550
8	.075000	.054363	.723235	-.562652	.985196
9	.140625	-.182317	.833511	-.760830	1.055631
10	.263625	-.278642	.874419	-.738598	1.047965
11	.390625	-.230724	.854314	-.739994	1.048448
12	.518625	-.086058	.790520	-.768144	1.057655
13	.640625	.141348	.678215	-.682882	1.028503
14	.765625	.364488	.546008	-.419234	.930906
15	.885000	.460487	.478009	-.118344	.805195
16	.945000	.447856	.487498	.108112	.699918
17	1.000000	.324958	.568480	.208969	.641253
AIRFOIL TRAILING EDGE					
18	1.090000	.147296	.675027	.147296	.675027
19	1.400000	.076301	.712149	.076301	.712149
20	1.875000	.029479	.735607	.029479	.735607



Y-GRID VALUES Y(J) J= 1 TO 14
 -5.200000 -2.400000 -1.150000 1.550000 -3.000000 -1.800000 -0.750000 .075000
 .180000 .300000 .550000 1.150000 2.400000 5.200000

Figure 4-4.- Continued.

INTERMEDIATE OUTPUT FOR MEDIUM MESH

WE = 1.9000 EPS = .2000 MAXIT FOR THIS MESH = 250

ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERRU	ICPL	CPERRL
10	.61625	-.15287	26	13	.2078E-02	10	11	.5886E+02	.1262E-02	17	.1267E+01	10	.1255E+31
20	.60949	-.15190	6	14	.6059E-03	19	14	.3003E+02	.1228E-02	19	.1433E-01	16	.1908E-01
30	.60690	-.15213	19	8	.2952E-03	10	15	.8248E+01	.2461E-03	18	.6059E+02	12	.1703E-01
40	.60546	-.15221	20	14	.2015E-03	10	6	.4499E+01	.1875E-03	27	.2313E-02	12	.7866E-02
50	.60507	-.15238	9	7	.1035E-03	10	14	.3935E+01	.5786E-04	14	.2075E-02	11	.2943E-02
60	.60464	-.15242	23	14	.5786E-04	10	14	.2842E+01	.7117E-04	14	.1498E-02	17	.2463E-03
70	.60432	-.15242	22	14	.4998E-04	10	14	.2046E+01	.4340E-04	14	.7971E-03	19	.1173E-03
80	.60414	-.15243	22	14	.3398E-04	10	14	.1612E+01	.2780E-04	19	.4925E-03	10	.1189E-03
90	.60400	-.15243	22	14	.2403E-04	10	14	.1294E+01	.2180E-04	19	.3968E-03	10	.8284E-04
100	.60390	-.15242	22	14	.1822E-04	10	14	.1007E+01	.1537E-04	19	.3038E-03	13	.4506E-04
110	.60382	-.15242	22	14	.1397E-04	10	14	.7818E+00	.1088E-04	19	.2256E-03	12	.4264E-04
120	.60377	-.15242	11	14	.1063E-04	10	14	.6022E+00	.8491E-05	19	.1736E-03	12	.4414E-04
130	.60373	-.15241	10	14	.8144E-05	10	14	.4618E+00	.6354E-05	19	.1337E-03	12	.3968E-04

***** SOLUTION CONVERGED *****

FINAL OUTPUT FOR MEDIUM MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.

CL = .603729
CM = -.152415
CP* = -.603170

B = BEFORE OR BEHIND AIRFOIL
U = UPPER AIRFOIL SURFACE
L = LOWER AIRFOIL SURFACE
* = CRITICAL PRESSURE

I	X	LOWER Y=0-		UPPER Y=+			
		CP	M1	CP	M1		
1	-1.075000	.021647	.739498	.021647	.739498		*
2	-.825000	.033335	.733764	.033335	.733704		*
3	-.575000	.051951	.724443	.051951	.724443		*
4	-.350000	.092701	.703748	.092701	.703748		*
5	-.175000	.177867	.658397	.177867	.658397		*
6	-.075600	.315444	.577665	.315444	.577665		*
7	-.035000	.458494	.479519	.458494	.479519		*
8	-.015000	.567892	.388057	.567892	.388057		*
9	-.002500	.654248	.295566	.654248	.295566		*
10	.007500	.942391	0.000000	.553319	.401445	L	*
11	.017500	.592448	.364366	.172386	.661409	U	*
12	.027500	.381819	.534372	-.087333	.788073		*
13	.037500	.205370	.643069	-.308037	.886527		*
14	.055000	.098007	.731391	-.389290	.994589		*
15	.075000	-.094830	.794536	-.836156	1.081200		U*
16	.097500	-.204465	.843093	-.907755	1.104459		*
17	.140625	-.294229	.687861	-.863369	1.090290		U
18	.203125	-.337801	.898621	-.812684	1.073293		*
19	.289625	-.346768	.899815	-.773972	1.060136		U
20	.328125	-.320584	.891840	-.750194	1.051971		*
21	.390625	-.276890	.874522	-.748073	1.048752		U
22	.453125	-.213683	.847049	-.746913	1.050839		U
23	.515625	-.125367	.808352	-.768921	1.058477		U
24	.578125	-.013389	.756447	-.797181	1.069045		U
25	.640625	.120502	.689272	-.756741	1.054225		U
26	.703125	.263402	.639463	-.590553	.995414		U*
27	.765625	.385401	.531935	-.408063	.926543		*
28	.828125	.455697	.481479	-.251294	.863002		U
29	.885000	.476548	.465664	-.091152	.792855		*
30	.915000	.473480	.468047	.032375	.734178		*
31	.945000	.459460	.478787	.132021	.683184		*
32	.975000	.426120	.503409	.223371	.632836		*
33	1.000000	.345091	.558744	.282392	.598056		*
34	1.025000	.229962	.629047	.229962	.629047		*
35	1.090000	.165655	.665090	.165655	.665090		*
36	1.225000	.110601	.694462	.110601	.694462		*
37	1.406000	.072623	.712494	.072623	.712494		*
38	1.625000	.059638	.722896	.059638	.722896		*
39	1.875000	.040203	.736301	.040203	.736301		*

Y-GRID VALUES Y(J) J= 1 TO 28

-5.200000	-3.600000	-2.400000	-1.600000	-1.150000	-.800000	-.550000	-.390000
-.300000	-.240000	-.180000	-.120000	-.075000	-.030000	.030000	.075000
.125000	.180000	.240000	.300000	.390000	.550000	.800000	1.150000
1.600000	2.400000	3.600000	5.200000				

Figure 4-4.- Continued.

INTERMEDIATE OUTPUT FOR FINE MESH

WE = 1.9500 EPS = .2000 MAXIT FOR THIS MESH = 500

ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERKU	ICPL	CPKRL
10	.60707	-.15260	9	29	.2196E-02	17	28	.3073E+03	.2779E-03	18	.1740E+01	19	.1765E+01
20	.60871	-.15277	16	21	.8121E-03	17	28	.1820E+03	.2453E-03	38	.3434E-01	21	.2270E-01
30	.61016	-.15295	16	31	.4264E-03	17	28	.1302E+03	.2539E-03	43	.2680E-01	20	.1020E-01
40	.61131	-.15288	37	28	.3469E-03	17	28	.1029E+03	.1811E-03	47	.2130E-01	24	.6088E-02
50	.61210	-.15264	16	31	.2650E-03	17	28	.8335E+02	.1167E-03	41	.1387E-01	21	.4292E-02
60	.61290	-.15290	18	29	.2146E-03	17	28	.7003E+02	.1274E-03	42	.1184E-01	25	.2479E-02
70	.61355	-.15288	18	29	.1854E-03	17	28	.5968E+02	.1127E-03	41	.8645E-02	24	.1804E-02
80	.61424	-.15294	18	29	.1566E-03	17	28	.5998E+02	.1145E-03	41	.7518E-02	24	.1483E-02
90	.61480	-.15293	18	29	.1345E-03	17	28	.4363E+02	.8579E-04	42	.6988E-02	24	.1285E-02
100	.61527	-.15293	18	29	.1160E-03	17	28	.3769E+02	.7737E-04	42	.6063E-02	24	.1066E-02
110	.61568	-.15294	19	29	.1032E-03	17	28	.3246E+02	.6443E-04	42	.5177E-02	24	.8096E-03
120	.61601	-.15293	18	29	.8708E-04	17	28	.2829E+02	.5434E-04	42	.4272E-02	24	.7525E-03
130	.61631	-.15294	18	29	.7521E-04	17	28	.2448E+02	.4896E-04	42	.3694E-02	24	.5851E-03
140	.61658	-.15294	18	29	.6548E-04	17	28	.2131E+02	.4467E-04	42	.2864E-02	24	.5467E-03
150	.61682	-.15294	18	29	.5699E-04	17	28	.1850E+02	.3881E-04	42	.2400E-02	24	.4749E-03
160	.61703	-.15295	18	29	.4939E-04	17	28	.1606E+02	.3344E-04	42	.2009E-02	24	.3959E-03
170	.61721	-.15295	18	29	.4298E-04	17	28	.1398E+02	.2903E-04	42	.2152E-02	24	.3431E-03
180	.61736	-.15295	18	29	.3742E-04	17	28	.1218E+02	.2523E-04	42	.1854E-02	24	.2944E-03
190	.61750	-.15295	18	29	.3261E-04	17	28	.1062E+02	.2218E-04	42	.1593E-02	24	.2502E-03
200	.61762	-.15295	18	29	.2842E-04	17	28	.9258E+01	.1955E-04	42	.1397E-02	24	.2188E-03
210	.61773	-.15295	18	29	.2478E-04	17	28	.8072E+01	.1714E-04	42	.1243E-02	24	.1897E-03
220	.61782	-.15295	18	29	.2251E-04	17	28	.7040E+01	.1495E-04	42	.1105E-02	24	.1665E-03
230	.61790	-.15295	18	29	.1895E-04	17	28	.6142E+01	.1302E-04	42	.9200E-03	24	.1429E-03
240	.61797	-.15295	18	29	.1645E-04	17	28	.5399E+01	.1135E-04	42	.7977E-03	24	.1241E-03
250	.61803	-.15295	18	29	.1435E-04	17	28	.4677E+01	.9918E-05	42	.6726E-03	24	.1061E-03
260	.61808	-.15295	18	29	.1253E-04	17	28	.4082E+01	.8674E-05	42	.6029E-03	24	.9429E-04
270	.61813	-.15295	18	29	.1093E-04	17	28	.3562E+01	.7578E-05	42	.5290E-03	24	.8231E-04
280	.61817	-.15295	18	29	.9538E-05	17	28	.3199E+01	.6611E-05	42	.4582E-03	24	.7188E-04

***** SOLUTION CONVERGED *****

 * FINAL OUTPUT *
 * * *

PRINTOUT IN PHYSICAL VARIABLES.
 DEFINITION OF SIMILARITY PARAMETERS BY KRUPP
 BOUNDARY CONDITION FOR FREE AIR
 DIFFERENCE EQUATIONS ARE FULLY CONSERVATIVE.
 KUTTA CONDITION IS ENFORCED.

MACH = .750000
 DELTA = .115000
 ALPHA = .120000
 K = 2.4667108
 DOUBLET STRENGTH = .0725602
 AIRFOIL VOLUME = .6755602

PARAMETERS USED TO TRANSFORM VARIABLES
 TO TRANSONIC SCALING

CPFACT = .2934287
 CDFACT = .0337443
 CMFACT = .2934287
 CLFACT = .2934287
 YFACT = 2.3744886
 VFACT = 6.5890146

Figure 4-4.- Continued.

FINAL OUTPUT FOR FINE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.

CL = .618170
CM = -.192947
CP* = -.603170

B = BEFORE OR BEHIND AIRFOIL
U = UPPER AIRFOIL SURFACE
L = LOWER AIRFOIL SURFACE
* = CRITICAL PRESSURE

I	X	LOWER Y=0-		UPPER Y=0+			
		CP	M1	CP	M1		
1	-1.075000	.034632	.733359	.034032	.733359		a
2	-.950000	.038228	.731281	.034228	.731281		B
3	-.825000	.043603	.728611	.043603	.728611		B
4	-.700000	.051881	.724478	.051881	.724478		B
5	-.575000	.064692	.718036	.064692	.718036		d
6	-.450000	.084149	.708141	.084149	.708141		d
7	-.350000	.112631	.693421	.112631	.693421		d
8	-.250000	.156104	.670278	.156104	.670278		B
9	-.175000	.217644	.638452	.217644	.638452		B
10	-.125000	.297629	.588744	.297629	.588744		B
11	-.075000	.419957	.514216	.419957	.514216		B
12	-.052500	.536533	.416334	.536533	.416334		B
13	-.035000	.662740	.285993	.662740	.285993		B
14	-.022500	.794714	0.000000	.794714	0.000000		B
15	-.015000	.918801	0.000000	.918801	0.000000		B
16	-.007500	1.041777	0.000000	1.041777	0.000000		B
17	-.002500	1.163438	0.000000	1.163438	0.000000		B
18	.002500	1.407124	0.000000	.929811	0.000000	L	U
19	.007500	.847589	0.000000	.364417	.546055		L
20	.012500	.545956	.408049	.069857	.715423		L
21	.017500	.358899	.549707	-.133933	.812166		L
22	.022500	.229092	.629549	-.282545	.876638		L
23	.027500	.134081	.682090	-.393788	.923439		L
24	.032500	.061736	.719528	-.483822	.954598		L
25	.037500	-.031223	.752591	-.567411	.986966		L
26	.045000	-.065930	.781231	-.646691	1.015661		L
27	.055000	-.126525	.808871	-.741993	1.049139		L
28	.065000	-.175851	.836693	-.849193	1.035564		L
29	.075000	-.215274	.847730	-.938462	1.114969		L
30	.085000	-.248385	.861778	-1.005396	1.136651		L
31	.097500	-.281089	.875431	-1.023934	1.124245		L
32	.115000	-.315831	.889710	-1.008197	1.137445		L
33	.140625	-.346290	.902845	-.979322	1.128201		L
34	.171875	-.364996	.909530	-.949933	1.118713		L
35	.203125	-.371656	.912179	-.923959	1.110261		L
36	.234375	-.371149	.911985	-.907749	1.102653		L
37	.265625	-.365818	.909857	-.879032	1.099487		L
38	.296875	-.355271	.905644	-.859039	1.088532		L
39	.328125	-.341392	.899666	-.836623	1.081155		L
40	.359375	-.320565	.891637	-.811060	1.072749		L
41	.390625	-.295284	.881294	-.781224	1.062614		L
42	.421875	-.264446	.868532	-.743156	1.049541		L
43	.453125	-.227512	.852949	-.701566	1.033495		L
44	.484375	-.184257	.834790	-.658262	1.023284		L
45	.515625	-.138117	.814052	-.610964	1.034859		L
46	.546875	-.084282	.789704	-.574064	1.048455		L
47	.578125	-.023989	.761512	-.542223	1.062955		L
48	.609375	.042142	.729337	-.518561	1.075282		L
49	.640625	.113148	.693140	-.483669	1.081472		L
50	.671875	.187714	.652951	-.473383	1.059917		L
51	.703125	.263167	.609639	-.491990	.995937		L
52	.734375	.333402	.566279	-.467771	.949627		L
53	.765625	.392678	.522951	-.417877	.930377		L
54	.796875	.436745	.495695	-.342798	.900619		L
55	.828125	.464815	.474867	-.257138	.858454		L
56	.859375	.478299	.464259	-.166233	.826483		L
57	.885000	.482335	.461135	-.086461	.790744		L
58	.900000	.481972	.461421	-.026327	.762624		L
59	.915000	.479614	.463270	.025433	.737599		L
60	.930000	.475295	.466639	.076993	.711797		L
61	.945000	.468316	.472032	.128491	.685055		L
62	.960000	.457556	.481228	.180440	.656978		L
63	.975000	.438907	.494111	.232955	.627320		L
64	.990000	.407740	.516972	.282117	.598223		L
65	1.000000	.353399	.553324	.314941	.577981		L
66	1.010000	.285581	.596123	.285581	.596120		L
67	1.025000	.248178	.618456	.248178	.618456		L
68	1.050000	.269303	.644847	.269303	.640847		L
69	1.090000	.170951	.662196	.173951	.662196		L
70	1.150000	.136726	.680682	.136726	.680682		L
71	1.225000	.110219	.694661	.110219	.694661		L
72	1.300000	.090361	.704952	.090361	.704952		L
73	1.400000	.074751	.712938	.074751	.712938		L
74	1.500000	.062705	.719039	.062705	.719039		L
75	1.625000	.053453	.723691	.053453	.723591		L
76	1.750000	.046883	.727002	.046883	.727062		L
77	1.875000	.041477	.729668	.041477	.729668		L

Figure 4-4.- Continued.

Y-GRID VALUES		Y(J) J= 1 TO 56					
-5.200000	-4.400000	-3.600000	-3.000000	-2.400000	-1.950000	-1.600000	-1.350000
-1.150000	-.950000	-.800000	-.650000	-.550000	-.450000	-.390000	-.340000
-.300000	-.270000	-.240000	-.210000	-.180000	-.150000	-.125000	-.100000
-.075000	-.050000	-.030000	-.010000	.010000	.030000	.050000	.075000
.100000	.125000	.150000	.180000	.210000	.240000	.270000	.300000
.340000	.390000	.450000	.550000	.650000	.800000	.950000	1.150000
1.350000	1.600000	1.950000	2.400000	3.000000	3.600000	4.400000	5.200000

FLOW CHARACTER MAP

P = PARABOLIC
H = HYPERBOLIC
S = SHOCK
- = ELLIPTIC

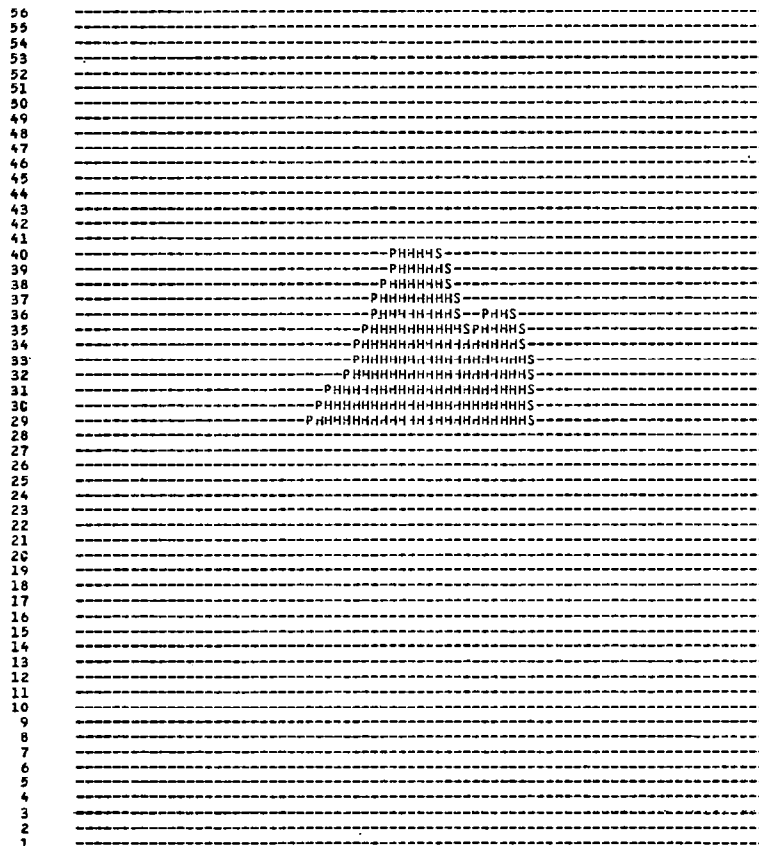
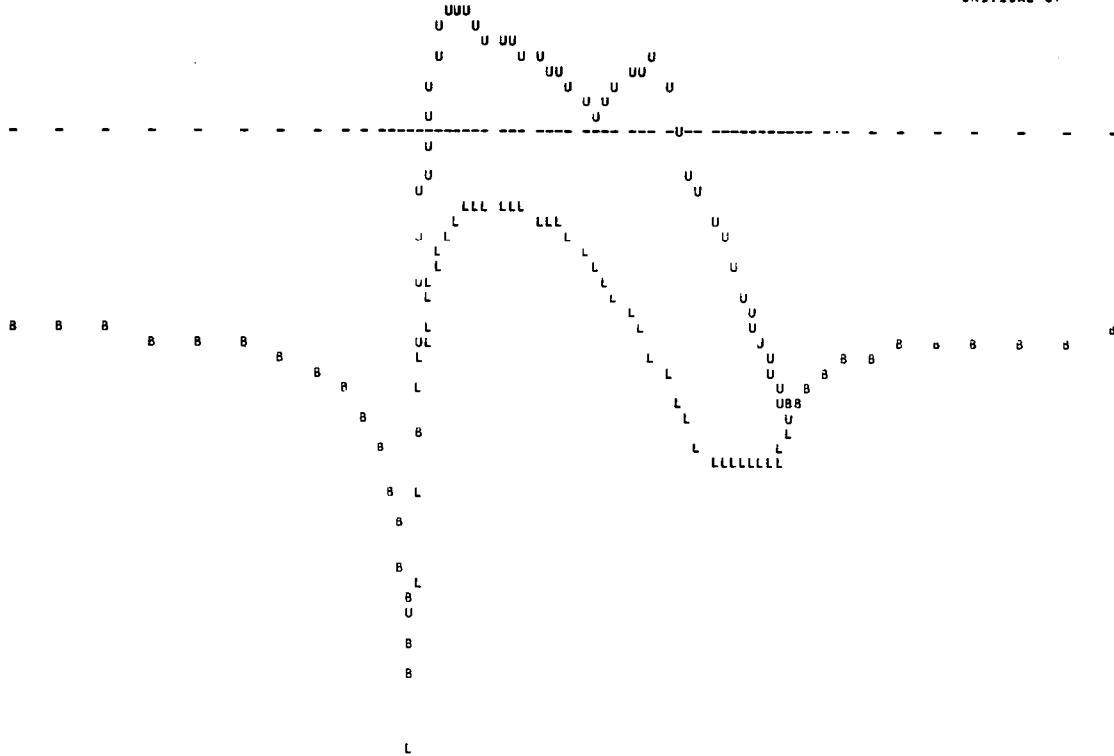


Figure 4-4.- Continued.

PRINTER PLOT OF CP ON BODY AND DIVIDING STREAMLINE

U = CP(UPPER)
 L = CP(LOWER)
 B = CP(UPPER) = CP(LOWER)
 --- = CRITICAL CP



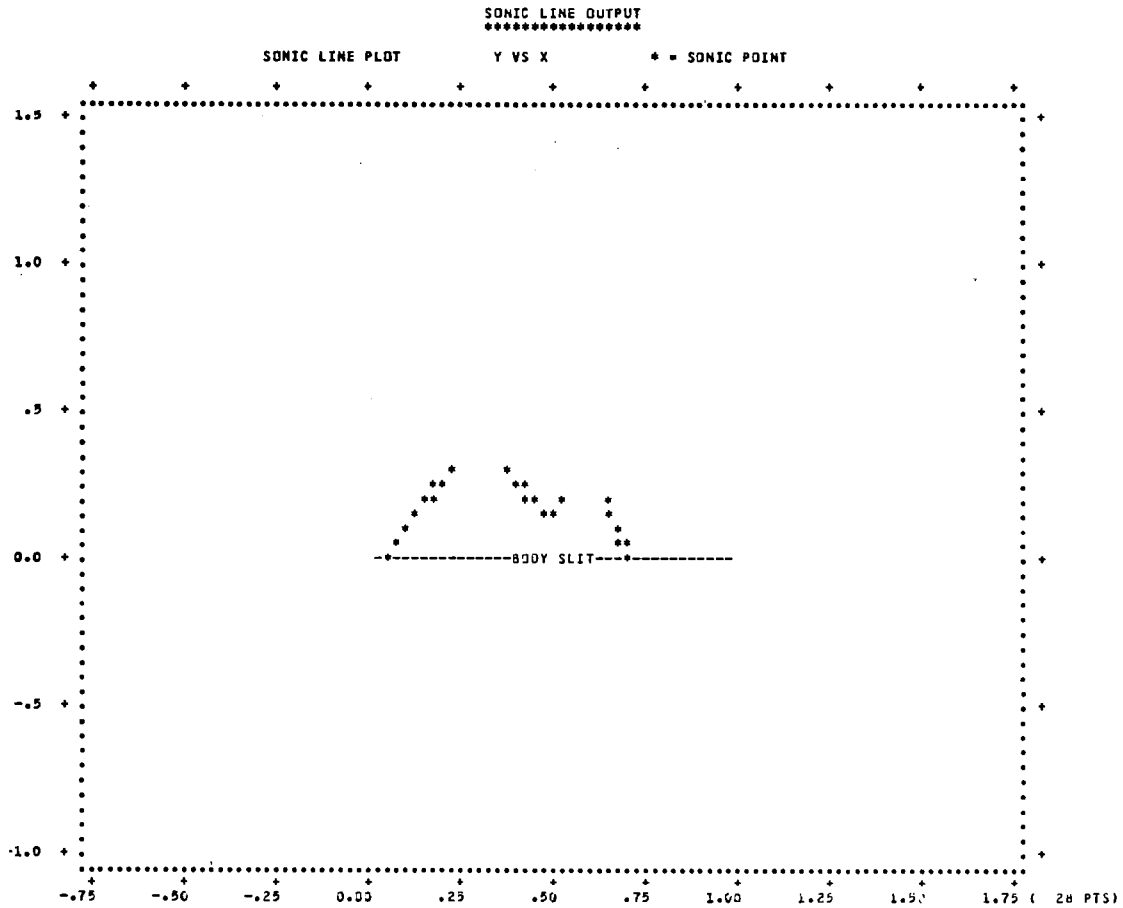
SONIC LINE COORDINATES

Y	X-SONIC	
.30000	.23428	.37126
.27000	.20723	.39561
.24000	.18509	.41554
.21000	.16381	.43445
.18000	.14349	.45086
.15000	.12669	.48161
.12500	.11184	.66557
.10000	.09882	.67579
.07500	.08642	.68326
.05000	.07393	.68982
.03000	.06287	.69457
.01000	.04913	.69902

BODY LOCATION

SONIC LINE OUTPUT

Figure 4-4.- Continued.



SHOCK WAVE DRAG AND TOTAL PRESSURE PROFILE OUTPUT

INVISCID WAKE PROFILES FOR INDIVIDUAL SHOCK WAVES WITHIN MOMENTUM CONTOUR

SHOCK 1

WAVE DRAG FOR THIS SHOCK=	THIS SHOCK=	.000149
Y	CD(Y)	PO/POINF
0.0000000	.00141537	.99997800
.01000000	.00138826	.99999566
.03000000	.00133509	.99913025
.05000000	.00123760	.99919381
.07500000	.00107696	.99929848
.10000000	.00089976	.99941387
.12500000	.00033395	.99978246
.15000000	.00030076	.99980412
.18000000	.00003761	.99997550

Figure 4-4.- Continued.

DRAG COEFFICIENT OUTPUT

CALCULATION OF DRAG COEFFICIENT BY MOMENTUM INTEGRAL METHOD

BOUNDARIES OF CONTOUR USED		CONTRIBUTION TO CD	
UPSTREAM	X = -0.175000	CDUP	= -0.002898
DOWNSTREAM	X = 1.225000	CDOWN	= 0.000679
TOP	Y = 4.400000	CDTP	= 0.000024
BOTTOM	Y = -4.400000	CDBT	= -0.000008
TOTAL CONTRIBUTIONS AROUND CONTOUR		=	-0.003203

THERE ARE 1 SHOCKS INSIDE CONTOUR. TOTAL CDWAVE = 0.003149

NOTE - ALL SHOCKS CONTAINED WITHIN CONTOUR
CDWAVE EQUALS TOTAL WAVE DRAG

DRAG CALCULATED FROM MOMENTUM INTEGRAL CD = -0.000054

TIME TO RUN CASE WAS 16.45 SECONDS.

(b) Output.
Figure 4-4.- Concluded

.20312500	.02877607	.02769939	-.02877607	-.02769939	.02877607	0.00003000
.23437500	.02947346	.01723458	-.02947346	-.01723458	.02947346	0.00003000
.26562500	.02986957	.00834494	-.02986957	-.00834494	.02986957	0.00003000
.29687500	.03006785	.0067163	-.03006785	-.0067163	.03006785	0.00003000
.32812500	.02992166	-.00602902	-.02992166	.00602902	.02992166	0.00003000
.35937500	.02963913	-.01193168	-.02963913	.01193168	.02963913	0.00003000
.39062500	.02918287	-.01716743	-.02918287	.01716743	.02918287	0.00000000
.42187500	.02857204	-.02183858	-.02857204	.02183858	.02857204	0.00003000
.45312500	.02782297	-.02602777	-.02782297	.02602777	.02782297	0.00003000
.48437500	.02694961	-.02980377	-.02694961	.02980377	.02694961	0.00003000
.51562500	.02596392	-.03322534	-.02596392	.03322534	.02596392	0.00003000
.54687500	.02487618	-.03634383	-.02487618	.03634383	.02487618	0.00003000
.57812500	.02369511	-.03920504	-.02369511	.03920504	.02369511	0.00003000
.60937500	.02242811	-.04185046	-.02242811	.04185046	.02242811	0.00003000
.64062500	.02108131	-.04431829	-.02108131	.04431829	.02108131	0.00003000
.67187500	.01965969	-.04664409	-.01965969	.04664409	.01965969	0.00003000
.70312500	.01816718	-.04886135	-.01816718	.04886135	.01816718	0.00003000
.73437500	.01660666	-.05100187	-.01660666	.05100187	.01660666	0.00003000
.76562500	.01498005	-.05309608	-.01498005	.05309608	.01498005	0.00003000
.79687500	.01328833	-.05517327	-.01328833	.05517327	.01328833	0.00003000
.82812500	.01153163	-.05726182	-.01153163	.05726182	.01153163	0.00003000
.85937500	.00970906	-.05938931	-.00970906	.05938931	.00970906	0.00003000
.89062500	.00816435	-.06118182	-.00816435	.06118182	.00816435	0.00003000
.92187500	.00723859	-.06252749	-.00723859	.06252749	.00723859	0.00003000
.95312500	.00629651	-.06335622	-.00629651	.06335622	.00629651	0.00003000
.98437500	.00533777	-.06448088	-.00533777	.06448088	.00533777	0.00003000
.99562500	.00436194	-.06563430	-.00436194	.06563430	.00436194	0.00003000
.99687500	.00336858	-.06661931	-.00336858	.06661931	.00336858	0.00003000
.99750000	.00235719	-.06863871	-.00235719	.06863871	.00235719	0.00003000
.99800000	.00132724	-.06992927	-.00132724	.06992927	.00132724	0.00003000
1.00000000	.00063000	-.07015500	-.00063000	.07015500	.00063000	0.00000000

INTERMEDIATE OUTPUT FOR COARSE MESH

WE = 1.8000 EPS = .2000 MAXIT FOR THIS MESH = 125

ITERATION	CL	CM	IERR	JERR	ERRDR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERRU	ICPL	CPERRL
10	.15585	.00554	18	6	.3094E-01	6	3	.1787E+03	.2912E-01	17	.9191E+01	17	.9617E+01
20	.23112	-.00164	16	6	.1675E-01	6	1	.1617E+03	.1632E-01	6	.9342E-01	9	.9380E-01
30	.28076	-.00255	5	1	.1595E-01	6	6	.2157E+03	.1172E-01	15	.9416E-01	6	.9696E-01
40	.31605	-.00266	4	1	.1155E-01	6	6	.1703E+03	.8326E-02	8	.9809E-01	17	.1973E-01
50	.34353	-.00316	6	1	.7618E-02	6	6	.1203E+03	.5963E-02	7	.9577E-01	8	.1352E-01
60	.36199	-.00335	5	1	.5974E-02	6	6	.9540E+02	.4317E-02	8	.9773E-01	6	.8552E-02
70	.37539	-.00345	5	1	.4401E-02	6	6	.7173E+02	.3147E-02	8	.9873E-01	10	.4243E-02
80	.38526	-.00348	5	1	.3338E-02	6	6	.5429E+02	.2329E-02	8	.9152E-01	16	.3706E-02
90	.39258	-.00354	4	1	.2445E-02	6	6	.4931E+02	.1739E-02	8	.8123E-01	16	.2152E-02
100	.39797	-.00360	5	1	.1804E-02	6	6	.2988E+02	.1272E-02	7	.1513E-01	16	.1873E-02
110	.40201	-.00363	5	1	.1322E-02	6	6	.2262E+02	.9612E-03	7	.1222E-01	16	.1899E-02
120	.40501	-.00364	5	1	.9915E-03	6	6	.1653E+02	.7084E-03	7	.9687E-02	16	.1242E-03

***** ITERATION LIMIT REACHED *****

Figure 4.5.- Continued

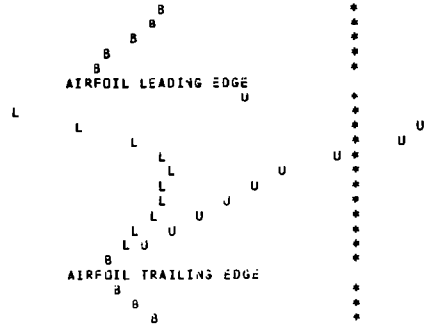
FINAL OUTPUT FOR COARSE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.

CL = .406213
CM = -.003640
CP* = -.603170

B = BEFORE OR BEHIND AIRFOIL
U = UPPER AIRFOIL SURFACE
L = LOWER AIRFOIL SURFACE
* = CRITICAL PRESSURE

I	X	LOWER Y=0-		UPPER Y=0+	
		CP	M1	CP	M1
1	-1.075000	-.018671	.758688	-.018071	.758688
2	-.975000	.003535	.748289	.003535	.748289
3	-.175000	.053214	.723811	.053214	.723811
4	-.035000	.126463	.686128	.126463	.686128
5	-.002500	.167177	.664260	.167177	.664260
AIRFOIL LEADING EDGE					
6	.017500	.622808	.332800	-.281678	.875648
7	.037500	.414301	.511853	-.592233	.995297
8	.075000	.219522	.635038	-.782709	1.063121
9	.140625	.054571	.723130	-.721086	1.041887
10	.265625	-.033574	.766063	-.534411	.974745
11	.390625	-.049898	.773792	-.390402	.919604
12	.515625	-.039715	.769065	-.293284	.888471
13	.640625	-.019512	.759376	-.208677	.844903
14	.765625	.008226	.746012	-.127227	.809185
15	.885000	.045206	.727812	-.045285	.771587
16	.945000	.086691	.706838	.030511	.735098
17	1.000000	.147981	.674659	.129331	.684611
AIRFOIL TRAILING EDGE					
18	1.090000	.101108	.699402	.101108	.699402
19	1.400000	.043505	.720659	.043505	.728659
20	1.875000	.012993	.743691	.012993	.743691



Y-GRID VALUES Y(J) J= 1 TO 12
 -2.000000 -1.200000 -.550000 -.300000 -.180000 -.075000 .075000 .180000
 .300000 .550000 1.200000 2.000000

INTERMEDIATE OUTPUT FOR MEDIUM MESH

WE = 1.9000 EPS = .2000 MAXIT FOR THIS MESH = 250

ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERRU	ICPL	CPERRL
10	.40865	.00109	9	10	.2390E-02	9	10	.1194E+03	.1412E-23	33	.1184E+01	10	.9603E+00
20	.40958	.00085	16	13	.2956E-02	10	12	.7445E+02	.4803E-03	11	.5313E-01	11	.1081E-01
30	.41075	.00088	16	13	.3343E-02	10	12	.5778E+02	.4971E-03	14	.7831E-02	10	.1852E-02
40	.41180	.00091	16	13	.3685E-02	10	12	.5043E+02	.5936E-03	10	.8490E-02	15	.1417E-02
50	.41275	.00093	16	13	.4000E-02	10	12	.4428E+02	.7167E-03	18	.7259E-02	12	.1376E-02
60	.41357	.00093	16	13	.4271E-02	10	12	.3488E+02	.8125E-03	18	.5412E-02	12	.9930E-03
70	.41433	.00095	15	13	.4503E-02	10	12	.3070E+02	.8965E-03	18	.4472E-02	12	.5217E-03
80	.41500	.00096	17	13	.4711E-02	10	12	.2635E+02	.9789E-03	18	.4304E-02	12	.3527E-03
90	.41558	.00097	17	13	.4918E-02	10	12	.2318E+02	.1051E-02	18	.3345E-02	12	.3272E-03
100	.41610	.00098	17	13	.5091E-02	14	13	.2091E+02	.1115E-02	18	.2848E-02	11	.1526E-03
110	.41655	.00098	17	13	.5233E-02	14	13	.2128E+02	.1169E-02	18	.2338E-02	18	.2814E-03
120	.41695	.00099	17	13	.5380E-02	14	13	.2161E+02	.1216E-02	18	.2029E-02	18	.1228E-03
130	.41729	.00099	17	13	.5471E-02	14	13	.2189E+02	.1260E-02	18	.1947E-02	18	.9542E-04
140	.41759	.00100	17	13	.5568E-02	14	13	.2214E+02	.1298E-02	18	.1871E-02	18	.9442E-04
150	.41785	.00100	17	13	.5650E-02	14	13	.2235E+02	.1336E-02	18	.1772E-02	18	.7375E-04
160	.41808	.00100	17	13	.5718E-02	14	13	.2252E+02	.1378E-02	18	.1122E-02	18	.1564E-03
170	.41828	.00100	17	13	.5789E-02	14	13	.2268E+02	.1382E-02	18	.9912E-03	18	.5764E-04
180	.41845	.00101	17	13	.5836E-02	14	13	.2282E+02	.1405E-02	18	.9893E-03	10	.3538E-04
190	.41860	.00101	17	13	.5885E-02	14	13	.2295E+02	.1424E-02	18	.7393E-03	18	.6389E-04
200	.41873	.00101	17	13	.5926E-02	14	13	.2305E+02	.1441E-02	18	.5804E-03	18	.4837E-04
210	.41885	.00101	17	13	.5958E-02	14	13	.2313E+02	.1454E-02	18	.5411E-03	29	.1443E-03
220	.41895	.00101	17	13	.5988E-02	14	13	.2321E+02	.1466E-02	18	.5147E-03	25	.4029E-04
230	.41904	.00101	17	13	.6019E-02	14	13	.2328E+02	.1478E-02	18	.5364E-03	18	.4943E-04
240	.41911	.00101	17	13	.6044E-02	14	13	.2335E+02	.1488E-02	18	.3695E-03	18	.5441E-04
250	.41917	.00101	17	13	.6064E-02	14	13	.2340E+02	.1496E-02	18	.2408E-03	18	.3650E-04

***** ITERATION LIMIT REACHED *****

Figure 4.5.- Continued

FINAL OUTPUT FOR MEDIUM MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.

CL = .419173
CM = .001013
CP* = -.603170

B = BEFORE OR BEHIND AIRFOIL
U = UPPER AIRFOIL SURFACE
L = LOWER AIRFOIL SURFACE
* = CRITICAL PRESSURE

I	X	LOWER Y=0-		UPPER Y=0+			
		CP	M1	CP	M1		
1	-1.075000	.002005	.749030	.002005	.749030		
2	-.825000	.007070	.746573	.007070	.746573		
3	-.575000	.016663	.741899	.016663	.741899		
4	-.350000	.040483	.730162	.040483	.730162		
5	-.175000	.093139	.703522	.093139	.703522		
6	-.075000	.181941	.656149	.181941	.656149		
7	-.035000	.273822	.603231	.273822	.603231		
8	-.015000	.338124	.563247	.338124	.563247		
9	-.002500	.376340	.538078	.376340	.538078		
AIRFOIL LEADING EDGE							
10	.007500	.918258	0.000000	-.153460	.820859	L	
11	.017500	.667224	.280249	-.482750	.955330	L	
12	.027500	.508325	.440221	-.797783	1.068251	L	
13	.037500	.376881	.537713	-.980251	1.128499	L	
14	.045000	.253899	.615092	-.963940	1.123116	L	
15	.075000	.158315	.669080	-.966368	1.124029	L	
16	.097500	.080143	.710190	-.965500	1.123748	L	
17	.140625	.010780	.744769	-.932189	1.112946	L	
18	.203125	-.034680	.766586	-.756319	1.054080	L	
19	.265625	-.055329	.776294	-.518140	.968672	L	
20	.328125	-.061852	.779335	-.415366	.929397	L	
21	.390625	-.060357	.778732	-.381778	.8916197	L	
22	.453125	-.054680	.775990	-.336338	.898030	L	
23	.515625	-.045948	.771899	-.291638	.879793	L	
24	.578125	-.035322	.766890	-.248923	.861837	L	
25	.640625	-.023147	.761111	-.207236	.844284	L	
26	.703125	-.009215	.754443	-.165908	.826340	L	
27	.765625	.007169	.746525	-.123632	.807573	L	
28	.828125	.027703	.736482	-.077478	.786573	L	
29	.885000	.050578	.725130	-.030166	.764448	L	
30	.915000	.074440	.713096	.008169	.746039	L	
31	.945000	.100658	.699635	.051955	.724441	L	
32	.975000	.141768	.677979	.109049	.693272	L	
33	1.000000	.199429	.646411	.192314	.650391	L	
AIRFOIL TRAILING EDGE							
34	1.025000	.153756	.671547	.153756	.671547	B	
35	1.090000	.092927	.703631	.092927	.703631	B	
36	1.225000	.052008	.724415	.052008	.724415	B	
37	1.400000	.030676	.735017	.030676	.735017	B	
38	1.625000	.019647	.740438	.019647	.740438	B	
39	1.875000	.011119	.744604	.011119	.744604	B	

Y-GRID VALUES		Y(J) J= 1 TO 24	
-2.000000	-1.600000	-1.200000	-.800000
-.180000	-.125000	-.075000	-.030000
.240000	.300000	.390000	.550000
			.800000
			1.200000
			1.600000
			2.000000

INTERMEDIATE OUTPUT FOR FINE MESH

ME = 1.9500		EPS = .2000		MAXIT FOR THIS MESH = 500											
ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERRU	ICPL	CPERRL		
10	.42002	.00324	9	25	.2361E-02	17	24	.4833E+03	.8166E-04	33	.1201E+01	18	.1378E+01		
20	.42058	.00327	17	17	.9564E-03	17	24	.2836E+03	.1431E-03	36	.5844E-01	23	.1734E-01		
30	.42107	.00340	72	36	.4896E-03	17	24	.1970E+03	.1395E-03	36	.3232E-01	19	.7999E-02		
40	.42159	.00343	16	28	.3429E-03	17	24	.1457E+03	.1307E-03	36	.2675E-01	21	.5662E-02		
50	.42210	.00340	16	27	.2993E-03	17	24	.1133E+03	.1346E-03	36	.1063E-01	23	.2689E-02		
60	.42258	.00342	16	28	.2195E-03	17	24	.9633E+02	.1238E-03	36	.1309E-01	22	.1462E-02		
70	.42302	.00343	16	27	.1850E-03	17	24	.8260E+02	.1140E-03	36	.1178E-01	20	.1052E-02		
80	.42343	.00342	21	1	.1725E-03	17	24	.7436E+02	.1075E-03	36	.8355E-02	24	.1060E-02		
90	.42383	.00341	18	1	.1598E-03	17	24	.6547E+02	.1002E-03	36	.6372E-02	21	.6330E-03		
100	.42419	.00342	12	1	.1416E-03	17	24	.5967E+02	.9275E-04	36	.6639E-02	28	.2572E-03		
110	.42453	.00342	16	4	.1294E-03	17	24	.5527E+02	.8777E-04	36	.5201E-02	24	.2063E-03		
120	.42485	.00342	14	4	.1171E-03	17	24	.5109E+02	.8375E-04	36	.4903E-02	57	.2704E-03		
130	.42516	.00343	15	6	.1083E-03	17	24	.4761E+02	.7901E-04	36	.4828E-02	24	.1983E-03		
140	.42549	.00343	18	1	.1033E-03	17	24	.4397E+02	.7460E-04	36	.4103E-02	62	.2563E-03		
150	.42572	.00344	18	3	.9330E-04	17	24	.4063E+02	.7007E-04	36	.4205E-02	52	.1146E-03		
160	.42598	.00344	16	2	.8833E-04	17	24	.3785E+02	.6584E-04	36	.3810E-02	24	.9666E-04		

Figure 4.5.- Continued

170	.42627	.00345	14	1	.8200E-04	17	24	.3500E+02	.6218E-04	36	.3572E-02	57	.1592E-03
180	.42645	.00345	15	1	.7606E-04	17	24	.3257E+02	.5819E-04	36	.3298E-02	24	.1096E-03
190	.42666	.00345	18	1	.7052E-04	17	24	.3017E+02	.5459E-04	36	.2904E-02	62	.1790E-03
200	.42686	.00345	18	1	.6571E-04	17	24	.2802E+02	.5117E-04	36	.2866E-02	52	.7496E-04
210	.42705	.00345	16	1	.6247E-04	17	24	.2629E+02	.4805E-04	36	.2544E-02	43	.4590E-04
220	.42722	.00346	14	1	.5839E-04	17	24	.2449E+02	.4525E-04	36	.2460E-02	57	.1685E-03
230	.42739	.00346	15	1	.5414E-04	17	24	.2293E+02	.4229E-04	36	.2269E-02	18	.5778E-04
240	.42754	.00346	18	1	.5041E-04	17	24	.2134E+02	.3968E-04	36	.2027E-02	62	.1241E-03
250	.42769	.00346	18	1	.4707E-04	17	24	.1988E+02	.3713E-04	36	.2047E-02	18	.5211E-04
260	.42782	.00346	16	1	.4479E-04	17	24	.1871E+02	.3483E-04	36	.1807E-02	18	.3307E-04
270	.42795	.00346	14	1	.4169E-04	17	24	.1746E+02	.3278E-04	36	.1750E-02	57	.7649E-04
280	.42807	.00347	21	1	.3889E-04	17	24	.1638E+02	.3062E-04	36	.1619E-02	18	.4585E-04
290	.42818	.00347	18	1	.3623E-04	17	24	.1527E+02	.2871E-04	36	.1448E-02	62	.8897E-04
300	.42828	.00347	18	1	.3386E-04	17	24	.1425E+02	.2686E-04	36	.1465E-02	18	.4630E-04
310	.42838	.00347	16	1	.3224E-04	17	24	.1343E+02	.2519E-04	36	.1297E-02	18	.2604E-04
320	.42847	.00347	14	1	.3003E-04	17	24	.1255E+02	.2370E-04	36	.1258E-02	57	.5507E-04
330	.42856	.00347	21	1	.2802E-04	17	24	.1178E+02	.2213E-04	36	.1164E-02	18	.3435E-04
340	.42864	.00347	18	1	.2612E-04	17	24	.1099E+02	.2075E-04	36	.1041E-02	62	.6419E-04
350	.42872	.00347	16	1	.2441E-04	17	24	.1026E+02	.1941E-04	36	.1055E-02	18	.2982E-04
360	.42879	.00347	16	1	.2325E-04	17	24	.9671E+01	.1820E-04	36	.9338E-03	18	.1934E-04
370	.42885	.00347	14	1	.2166E-04	17	24	.9042E+01	.1713E-04	36	.9058E-03	57	.3978E-04
380	.42892	.00347	21	1	.2022E-04	17	24	.8494E+01	.1599E-04	36	.8393E-03	18	.2516E-04
390	.42897	.00347	18	1	.1884E-04	17	24	.7924E+01	.1499E-04	36	.7513E-03	62	.4641E-04
400	.42903	.00348	18	1	.1762E-04	17	24	.7400E+01	.1403E-04	36	.7616E-03	18	.2174E-04
410	.42908	.00348	16	1	.1678E-04	17	24	.6977E+01	.1315E-04	36	.6742E-03	18	.1416E-04
420	.42913	.00348	14	1	.1563E-04	17	24	.6525E+01	.1237E-04	36	.6541E-03	57	.2877E-04
430	.42917	.00348	21	1	.1460E-04	17	24	.6130E+01	.1155E-04	36	.6462E-03	18	.1825E-04
440	.42922	.00348	18	1	.1360E-04	17	24	.5720E+01	.1083E-04	36	.5426E-03	62	.3394E-04
450	.42925	.00348	18	1	.1272E-04	17	24	.5342E+01	.1013E-04	36	.5502E-03	18	.1575E-04
460	.42929	.00348	16	1	.1212E-04	17	24	.5037E+01	.9499E-05	36	.4870E-03	18	.1021E-04
470	.42933	.00348	14	1	.1129E-04	17	24	.4711E+01	.8938E-05	36	.4726E-03	57	.2082E-04
480	.42936	.00348	21	1	.1054E-04	17	24	.4426E+01	.8347E-05	36	.4380E-03	18	.1321E-04
490	.42939	.00348	18	1	.9824E-05	17	24	.4130E+01	.7826E-05	36	.3920E-03	62	.2430E-04

***** SOLUTION CONVERGED *****

*
* FINAL OUTPUT *
*

PRINTOUT IN PHYSICAL VARIABLES.

DEFINITION OF SIMILARITY PARAMETERS BY KRUPP

BOUNDARY CONDITION FOR SOLID WALL

DIFFERENCE EQUATIONS ARE FULLY CONSERVATIVE.

KUTTA CONDITION IS ENFORCED.

MACH = .7500000
DELTA = .0600000
ALPHA = 2.0000000
K = 3.8061213
DOUBLET STRENGTH = .0516114
AIRFOIL VOLUME = .0410899

PARAMETERS USED TO TRANSFORM VARIABLES
TO TRANSONIC SCALING

CPFACT = .1901683
CDFACT = .0114101
CMFACT = .1901683
CLFACT = .1901683
YFACT = 2.9493264
VFACT = 3.4377467

Figure 4.5.- Continued

FINAL OUTPUT FOR FINE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.

CL = .429390
CM = .003479
CP* = -.603170

B = BEFORE OR BEHIND AIRFOIL
U = UPPER AIRFOIL SURFACE
L = LOWER AIRFOIL SURFACE
* = CRITICAL PRESSURE

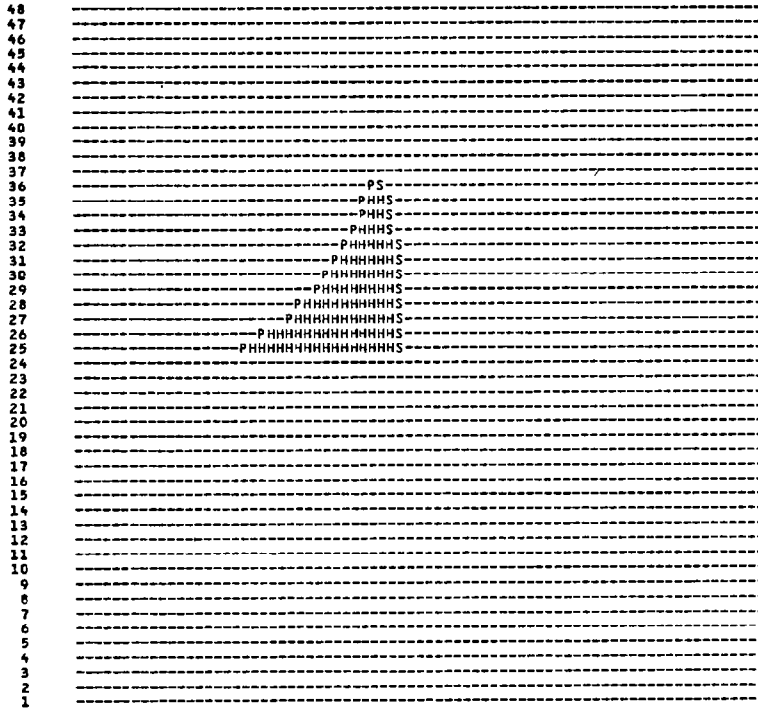
I	X	LOWER Y=0-		UPPER Y=0+			
		CP	M1	CP	M1		
1	-1.075000	.008699	.745782	.008699	.745782		
2	-.950000	.010789	.744765	.010789	.744765		
3	-.825000	.013555	.743416	.013555	.743416		
4	-.700000	.018026	.741232	.018026	.741232		
5	-.575000	.025241	.737693	.025241	.737693		
6	-.450000	.036605	.732086	.036605	.732086		
7	-.325000	.053754	.723549	.053754	.723549		
8	-.200000	.080668	.709922	.080668	.709922		
9	-.175000	.119291	.689909	.119291	.689909		
10	-.125000	.171516	.661886	.171516	.661886		
11	-.075000	.246613	.619373	.246613	.619373		
12	-.052500	.331554	.567461	.331554	.567461		
13	-.035000	.419085	.508452	.419085	.508452		
14	-.022500	.511797	.437351	.511797	.437351		
15	-.015000	.597950	.358868	.597950	.358868		
16	-.007500	.678353	.265456	.678353	.265456		
17	-.002500	.748611	.139665	.748611	.139665		
AIRFOIL LEADING EDGE							
18	.002500	1.322901	0.000000	.076052	.712276	L	U
19	.007500	.946407	0.000000	-.456347	.945253	L	U
20	.012500	.729237	.183192	-.807212	1.071447	L	U
21	.017500	.579391	.377158	-1.005689	1.136645	L	U
22	.022500	.470207	.470577	-1.024078	1.142497	L	U
23	.027500	.388169	.530045	-1.046277	1.149522	L	U
24	.032500	.325669	.571591	-1.074236	1.158309	L	U
25	.037500	.269801	.605644	-1.080325	1.163277	L	U
26	.045000	.213693	.638358	-1.077035	1.159185	L	U
27	.055000	.162263	.666937	-1.071090	1.157324	L	U
28	.065000	.122016	.688475	-1.067760	1.156280	L	U
29	.075000	.091169	.704537	-1.066786	1.155974	L	U
30	.085000	.065166	.717797	-1.0665945	1.155711	L	U
31	.097500	.039543	.730629	-1.066280	1.154686	L	U
32	.115000	.012083	.744134	-1.054815	1.152213	L	U
33	.140625	-.015048	.757241	-1.042097	1.148202	L	U
34	.171875	-.037047	.767705	-1.025352	1.142901	L	U
35	.203125	-.052108	.774788	-1.001581	1.135333	L	U
36	.234375	-.061703	.779266	-.845911	1.084467	L	U
37	.265625	-.067407	.781916	-.515745	.967774	L	U
38	.296875	-.070239	.783228	-.346674	.92194	L	U
39	.328125	-.070966	.783537	-.370809	.911845	L	U
40	.359375	-.069920	.782081	-.371587	.912154	L	U
41	.390625	-.067660	.782033	-.360922	.907904	L	U
42	.421875	-.064411	.780525	-.344785	.901435	L	U
43	.453125	-.060386	.778053	-.326131	.893886	L	U
44	.484375	-.055747	.774489	-.306299	.885815	L	U
45	.515625	-.050613	.774087	-.286087	.877501	L	U
46	.546875	-.045066	.771484	-.265806	.869079	L	U
47	.578125	-.039155	.768700	-.245599	.860605	L	U
48	.609375	-.032899	.765743	-.225498	.852092	L	U
49	.640625	-.026289	.762606	-.205458	.843520	L	U
50	.671875	-.019282	.759267	-.185381	.834843	L	U
51	.703125	-.011860	.755685	-.165116	.825993	L	U
52	.734375	-.003719	.751796	-.144455	.816871	L	U
53	.765625	.005148	.747507	-.123119	.807343	L	U
54	.796875	.015074	.742675	-.100721	.797218	L	U
55	.828125	.026472	.737088	-.076713	.786220	L	U
56	.859375	.039336	.730731	-.051494	.774500	L	U
57	.890625	.052791	.724459	-.028437	.763627	L	U
58	.920000	.062681	.719051	-.009777	.754713	L	U
59	.950000	.073340	.713655	.007891	.746175	L	U
60	.980000	.085997	.707194	.027957	.736357	L	U
61	.945000	.101577	.699159	.051517	.724660	L	U
62	.960000	.121871	.688551	.080671	.709920	L	U
63	.975000	.150702	.673194	.119856	.689612	L	U
64	.990000	.192528	.650271	.173673	.650703	L	U
65	1.000000	.249123	.617902	.243370	.621269	L	U
AIRFOIL TRAILING EDGE							
66	1.010000	.215841	.637137	.215841	.637137		
67	1.025000	.165663	.665085	.165663	.665085		
68	1.050000	.123379	.687757	.123379	.687757		
69	1.090000	.089664	.705311	.089664	.705311		
70	1.150000	.064386	.718191	.064386	.718191		
71	1.225000	.047177	.726829	.047177	.726829		
72	1.300000	.035460	.732653	.035460	.732653		
73	1.400000	.026855	.736960	.026855	.736960		
74	1.500000	.020985	.739979	.020985	.739979		
75	1.625000	.015970	.742237	.015970	.742237		
76	1.750000	.012765	.743802	.012765	.743802		
77	1.875000	.010204	.745050	.010204	.745050		

Figure 4.5.- Continued

Y-GRID VALUES		Y(J) J= 1 TO 48					
-2.00000	-1.80000	-1.60000	-1.40000	-1.20000	-1.00000	-.80000	-.65000
-.55000	-.45000	-.35000	-.34000	-.30000	-.27000	-.24000	-.21000
-.18000	-.15000	-.12500	-.10000	-.07500	-.05000	-.03000	-.01000
.01000	.03000	.05000	.07500	.10000	.12500	.15000	.18000
.21000	.24000	.27000	.30000	.34000	.39000	.45000	.55000
.65000	.80000	1.00000	1.20000	1.40000	1.60000	1.80000	2.00000

FLOW CHARACTER MAP

P = PARABOLIC
H = HYPERBOLIC
S = SHOCK
- = ELLIPTIC



MACH NUMBER MAP

SYMBOL	RANGE	SYMBOL	RANGE	SYMBOL	RANGE	SYMBOL	RANGE
0	M.LE.0.05	A	0.95.LT.M.LE.1.05	K	1.95.LT.M.LE.2.05	U	2.95.LT.M.LE.3.05
1	0.05.LT.M.LE.0.15	B	1.05.LT.M.LE.1.15	L	2.05.LT.M.LE.2.15	V	3.05.LT.M.LE.3.15
2	0.15.LT.M.LE.0.25	C	1.15.LT.M.LE.1.25	M	2.15.LT.M.LE.2.25	W	3.15.LT.M.LE.3.25
3	0.25.LT.M.LE.0.35	D	1.25.LT.M.LE.1.35	N	2.25.LT.M.LE.2.35	X	3.25.LT.M.LE.3.35
4	0.35.LT.M.LE.0.45	E	1.35.LT.M.LE.1.45	O	2.35.LT.M.LE.2.45	Y	3.35.LT.M.LE.3.45
5	0.45.LT.M.LE.0.55	F	1.45.LT.M.LE.1.55	P	2.45.LT.M.LE.2.55	Z	3.45.LT.M.LE.3.55
6	0.55.LT.M.LE.0.65	G	1.55.LT.M.LE.1.65	Q	2.55.LT.M.LE.2.65	*	M.GT.3.55
7	0.65.LT.M.LE.0.75	H	1.65.LT.M.LE.1.75	R	2.65.LT.M.LE.2.75		
8	0.75.LT.M.LE.0.85	I	1.75.LT.M.LE.1.85	S	2.75.LT.M.LE.2.85		
9	0.85.LT.M.LE.0.95	J	1.85.LT.M.LE.1.95	T	2.85.LT.M.LE.2.95		

Figure 4.5.- Continued

 PRINTER PLOT OF CP ON BODY AND DIVIDING STREAMLINE

U = CP(UPPER)
 L = CP(LOWER)
 B = CP(UPPER) = CP(LOWER)
 --- = CRITICAL CP

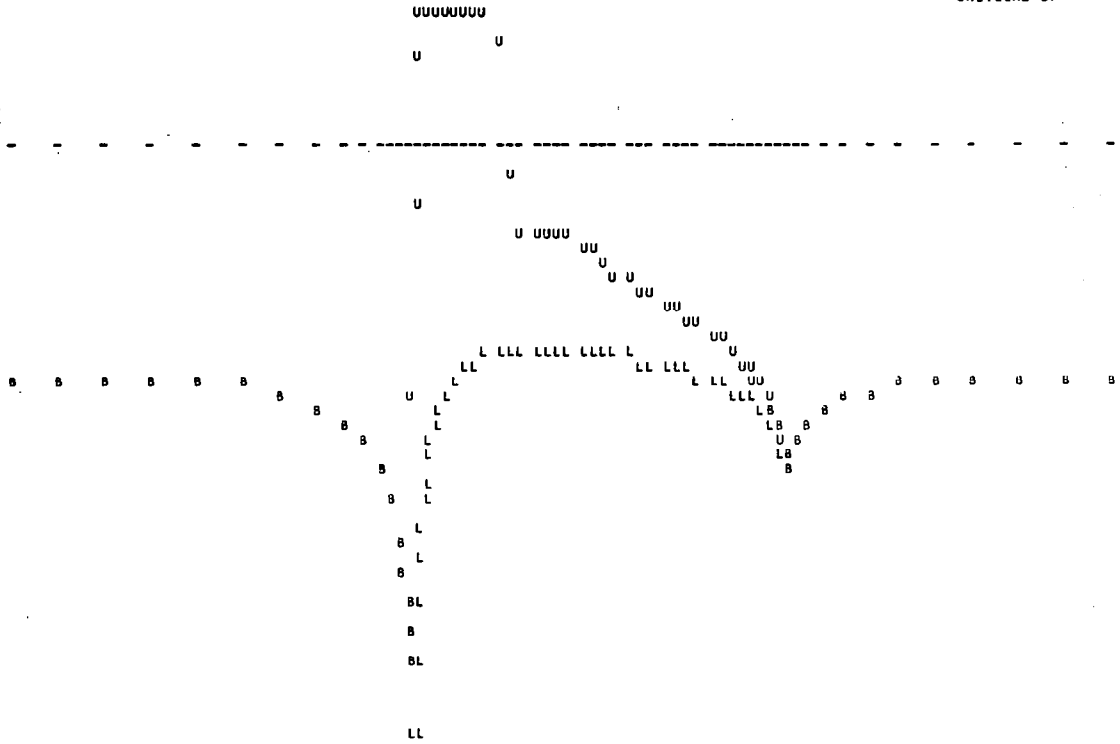


Figure 4.5.- Continued

SOLID WALL BOUNDARY CONDITION.

H (TUNNEL HALF HEIGHT) = 2.000000

SCALED H = .678075

CP* = -.603170

I	X	LOWER Y=-H		UPPER Y=+H	
		CP	THETA	CP	THETA
2	-.950000	.035447	0.000000	-.648436	0.000000
3	-.825000	.037896	0.000000	-.654969	0.000000
4	-.700000	.040234	0.000000	-.662600	0.000000
5	-.575000	.042339	0.000000	-.671347	0.000000
6	-.450000	.044013	0.000000	-.680584	0.000000
7	-.325000	.045193	0.000000	-.689420	0.000000
8	-.250000	.045920	0.000000	-.697374	0.000000
9	-.175000	.046272	0.000000	-.703624	0.000000
10	-.125000	.046393	0.000000	-.708106	0.000000
11	-.075000	.046424	0.000000	-.711338	0.000000
12	-.052500	.046424	0.000000	-.713315	0.000000
13	-.035000	.046423	0.000000	-.714488	0.000000
14	-.022500	.046427	0.000000	-.715272	0.000000
15	-.015000	.046502	0.000000	-.715802	0.000000
16	-.007500	.046558	0.000000	-.716193	0.000000
17	-.002500	.046618	0.000000	-.716499	0.000000
18	.002500	.046618	0.000000	-.716793	0.000000
19	.007500	.046614	0.000000	-.717088	0.000000
20	.012500	.046610	0.000000	-.717374	0.000000
21	.017500	.046605	0.000000	-.717657	0.000000
22	.022500	.046599	0.000000	-.717936	0.000000
23	.027500	.046592	0.000000	-.718211	0.000000
24	.032500	.046584	0.000000	-.718481	0.000000
25	.037500	.046511	0.000000	-.718810	0.000000
26	.045000	.046392	0.000000	-.719248	0.000000
27	.055000	.046319	0.000000	-.719739	0.000000
28	.065000	.046268	0.000000	-.720228	0.000000
29	.075000	.046213	0.000000	-.720697	0.000000
30	.085000	.046132	0.000000	-.721181	0.000000
31	.097500	.046007	0.000000	-.721754	0.000000
32	.115000	.045836	0.000000	-.722479	0.000000
33	.140625	.045612	0.000000	-.723327	0.000000
34	.171875	.045341	0.000000	-.724126	0.000000
35	.203125	.045037	0.000000	-.724714	0.000000
36	.234375	.044707	0.000000	-.725044	0.000000
37	.265625	.044354	0.000000	-.725113	0.000000
38	.296875	.043982	0.000000	-.724916	0.000000
39	.328125	.043593	0.000000	-.724455	0.000000
40	.359375	.043189	0.000000	-.723733	0.000000
41	.390625	.042770	0.000000	-.722760	0.000000
42	.421875	.042336	0.000000	-.721547	0.000000
43	.453125	.041895	0.000000	-.720104	0.000000
44	.484375	.041444	0.000000	-.718445	0.000000
45	.515625	.040984	0.000000	-.716585	0.000000
46	.546875	.040517	0.000000	-.714542	0.000000
47	.578125	.040046	0.000000	-.712330	0.000000
48	.609375	.039570	0.000000	-.709967	0.000000
49	.640625	.039091	0.000000	-.707469	0.000000
50	.671875	.038610	0.000000	-.704855	0.000000
51	.703125	.038125	0.000000	-.702141	0.000000
52	.734375	.037638	0.000000	-.699344	0.000000
53	.765625	.037147	0.000000	-.696481	0.000000
54	.796875	.036652	0.000000	-.693566	0.000000
55	.828125	.036155	0.000000	-.690612	0.000000
56	.859375	.035658	0.000000	-.687767	0.000000
57	.885000	.035211	0.000000	-.685425	0.000000
58	.900000	.035062	0.000000	-.683725	0.000000
59	.915000	.034822	0.000000	-.682286	0.000000
60	.930000	.034581	0.000000	-.680849	0.000000
61	.945000	.034329	0.000000	-.679424	0.000000
62	.960000	.034078	0.000000	-.678002	0.000000
63	.975000	.033840	0.000000	-.676570	0.000000
64	.990000	.033649	0.000000	-.675253	0.000000
65	1.000000	.033496	0.000000	-.674181	0.000000
66	1.010000	.033284	0.000000	-.673129	0.000000
67	1.025000	.032962	0.000000	-.671619	0.000000
68	1.050000	.032482	0.000000	-.669191	0.000000
69	1.090000	.031747	0.000000	-.665429	0.000000
70	1.130000	.030694	0.000000	-.660213	0.000000
71	1.225000	.029391	0.000000	-.654143	0.000000
72	1.300000	.027857	0.000000	-.647599	0.000000
73	1.400000	.026028	0.000000	-.640577	0.000000
74	1.500000	.023890	0.000000	-.633290	0.000000
75	1.625000	.021453	0.000000	-.625941	0.000000
76	1.750000	.018888	0.000000	-.618997	0.000000

Figure 4.5.- Continued

SONIC LINE COORDINATES

Y	X-SONIC	
.30000	.17004	.17632
.27000	.13900	.21245
.24000	.12090	.22340
.21000	.10553	.23204
.18000	.09171	.23862
.15000	.07840	.24356
.12500	.06694	.24689
.10000	.05546	.24968
.07500	.04410	.25205
.05000	.03266	.25408
.03000	.02244	.25549
.01000	.01172	.25674

BODY LOCATION

SONIC LINE OUTPUT

SONIC LINE OUTPUT

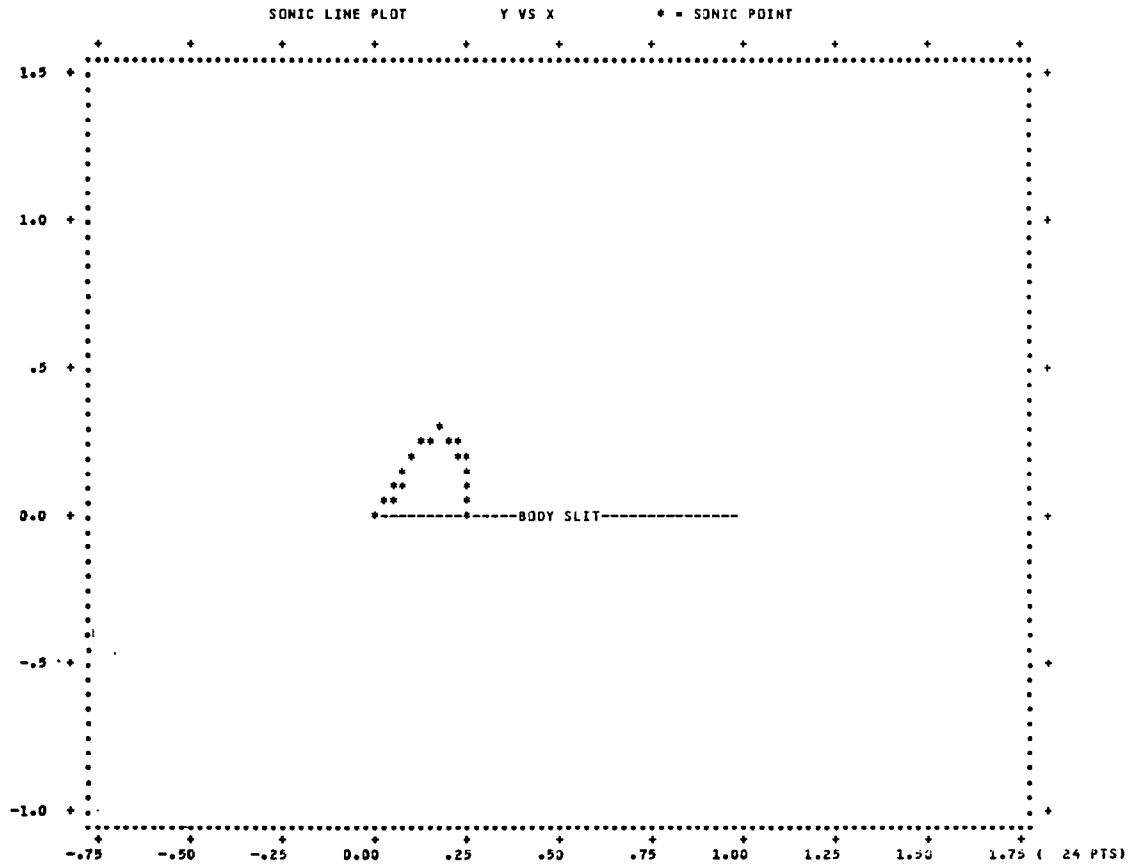


Figure 4.5.- Continued

SHOCK WAVE DRAG AND TOTAL PRESSURE PROFILE OUTPUT

INVISCID WAKE PROFILES FOR INDIVIDUAL SHOCK WAVES WITHIN MOMENTUM CONTOUR

SHOCK 1
 WAVE DRAG FOR THIS SHOCK- .001058
 Y CD(Y) PO/POINF

0.0000000	.00790009	.99485370
.01000000	.00769835	.99498512
.03000000	.00730523	.99524120
.05000000	.00677634	.99558573
.07500000	.00597943	.99610486
.10000000	.00509296	.99668233
.12500000	.00417618	.99727934
.15000000	.00328669	.99785897
.18000000	.00232634	.99848457
.21000000	.00102329	.99933340
.24000000	.00066724	.99958536
.27000000	.00040805	.99973419
.30000000	.00000895	.99999417

DRAG COEFFICIENT OUTPUT

CALCULATION OF DRAG COEFFICIENT BY MOMENTUM INTEGRAL METHOD

BOUNDARIES OF CONTOUR USED		CONTRIBUTION TO CD
UPSTREAM	X = -1.175000	CDUP = -.000055
DOWNSTREAM	X = 1.225000	CDDOWN = .000219
TOP	Y = 1.800000	CDTOP = .003140
BOTTOM	Y = -1.800000	CDBOT = .003029
TOTAL CONTRIBUTIONS AROUND CONTOUR		= .003333

THERE ARE 1 SHOCKS INSIDE CONTOUR. TOTAL CDWAVE = .001058

NOTE - ALL SHOCKS CONTAINED WITHIN CONTOUR
 CDWAVE EQUALS TOTAL WAVE DRAG

DRAG CALCULATED FROM MOMENTUM INTEGRAL CD = .001391

TIME TO RUN CASE WAS 23.64 SECONDS.

(b) Output.
 Figure 4.5.- Concluded

SAMPLE CASE 3 - SUBSONIC-NACA0006 AIRFOIL-SLOTTED TUNNEL WALL SIMULATION
 \$IMP
 BCTYPE=4, F=0.066, EPS=0.9, H=2.0, ALPHA=2.0, BETA=0.06, EMACH=0.75,
 BCFOIL=1,
 \$END

(a) Input.

SAMPLE CASE 3 - SUBSONIC - NACA0006 AIRFOIL - SLOTTED TUNNEL WALL SIMULATION

INPUT PARAMETERS

EMACH = .75000	POR = 0.00000	IMIN = 1	BCTYPE = 4	AMESH = F			
DELTA = .06000	CLSET = 0.00000	IMAXI = 77	BCFOIL = 1	PHYS = T			
ALPHA = 2.00000	EPS = .90000	JMIN = 1	PSTART = 1	PSAVE = F			
AK = 0.00000	RIGF = 0.00000	JMAXI = 48	PRTFLO = 1	KUTTA = T			
GAM = 1.40000	WCIRC = 1.00000	MAXIT = 500	IPRTER = 10	FCR = T			
F = .66600	CVERGE = .00001	NU = 100	SIMDEF = 3				
H = 2.00000	DVERGE = 10.0	NL = 75	ICUT = 2				
WE = 1.80,1.90,1.95							
XIM							
-1.075000	-.950000	-.825000	-.700000	-.575000	-.450000	-.325000	-.200000
-.175000	-.125000	-.075000	-.025000	-.035000	-.022500	-.015000	-.007500
-.002500	.002500	.007500	.012500	.017500	.022500	.027500	.032500
.037500	.045000	.055000	.065000	.075000	.085000	.097500	.115000
.140625	.171875	.203125	.234375	.265625	.296875	.328125	.359375
.390625	.421875	.453125	.484375	.515625	.546875	.578125	.609375
.640625	.671875	.703125	.734375	.765625	.796875	.828125	.859375
.889000	.900000	.915000	.930000	.945000	.960000	.975000	.990000
1.000000	1.010000	1.025000	1.050000	1.090000	1.150000	1.225000	1.300000
1.400000	1.500000	1.625000	1.750000	1.875000			
YIM							
-2.000000	-1.800000	-1.600000	-1.400000	-1.200000	-1.000000	-.800000	-.600000
-.550000	-.450000	-.390000	-.340000	-.300000	-.270000	-.240000	-.210000
-.180000	-.150000	-.125000	-.100000	-.075000	-.050000	-.030000	-.010000
.010000	.030000	.050000	.075000	.100000	.125000	.150000	.180000
.210000	.240000	.270000	.300000	.340000	.390000	.450000	.500000
.650000	.800000	1.000000	1.200000	1.400000	1.600000	1.800000	2.000000

Figure 4-6.- Sample test case 3 - subsonic, slotted tunnel wall simulation (abbreviated output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2^\circ$, $M_\infty = 0.75$, $H = 2.$, $F = 0.066$.

INTERMEDIATE OUTPUT FOR COARSE MESH

WE = 1.8000		EPS = .9000		MAXIT FOR THIS MESH = 125									
ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERRU	ICPL	CPERRL
10	.15802	.00490	18	6	.3662E-01	6	1	.3173E+03	.2974E-01	14	.5651E+00	17	.5356E+00
20	.23257	.60309	13	6	.1839E-01	6	6	.2311E+03	.1551E-01	6	.1129E+00	8	.9997E-01
30	.27626	.00252	5	3	.1299E-01	6	6	.2310E+03	.9457E-02	8	.6227E-01	6	.3455E-01
40	.30252	.00261	4	3	.7720E-02	6	6	.1355E+03	.5715E-02	7	.8238E-01	7	.1226E-01
50	.31826	.00275	4	3	.4461E-02	6	6	.7757E+02	.3441E-02	7	.5691E-01	16	.6528E-02
60	.32763	.00287	4	3	.2748E-02	6	6	.4782E+02	.2085E-02	7	.3101E-01	16	.2990E-02
70	.33361	.00295	4	3	.1661E-02	6	6	.2892E+02	.1263E-02	8	.2503E-01	16	.2614E-02
80	.33712	.00301	4	3	.1016E-02	6	6	.1768E+02	.7692E-03	8	.1238E-01	16	.1289E-02
90	.33927	.00306	4	3	.6170E-03	6	6	.1072E+02	.4686E-03	8	.8183E-02	16	.6482E-03
100	.34057	.00308	4	3	.3735E-03	6	6	.6490E+01	.2848E-03	8	.5041E-02	16	.4490E-03
110	.34136	.00310	4	3	.2292E-03	6	6	.3985E+01	.1741E-03	8	.3077E-02	16	.2877E-03
120	.34185	.00311	4	3	.1385E-03	6	6	.2409E+01	.1057E-03	8	.1922E-02	16	.1513E-03

***** ITERATION LIMIT REACHED *****

INTERMEDIATE OUTPUT FOR MEDIUM MESH

WE = 1.9000		EPS = .9000		MAXIT FOR THIS MESH = 250									
ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERRU	ICPL	CPERRL
10	.34350	.00632	13	10	.1818E-02	10	10	.1131E+03	.1769E-03	33	.1253E+01	10	.9536E+00
20	.34382	.00646	20	13	.8586E-03	10	12	.4597E+02	.1370E-03	3	.3599E-01	12	.7775E-02
30	.34436	.00653	12	6	.4063E-03	10	12	.2603E+02	.1698E-03	13	.4338E-01	30	.2813E-02
40	.34487	.00661	12	13	.3335E-03	10	12	.2686E+02	.1326E-03	13	.2392E-01	24	.1627E-02
50	.34534	.00672	12	13	.2844E-03	10	12	.1859E+02	.1101E-03	13	.2592E-01	3	.2228E-02
60	.34570	.00683	12	13	.2442E-03	10	12	.1174E+02	.9147E-04	13	.3718E-01	12	.1587E-02
70	.34604	.00700	12	13	.1631E-03	10	12	.8725E+01	.8511E-04	13	.5204E-01	12	.7622E-03
80	.34632	.00703	9	6	.7737E-04	10	12	.5746E+01	.6756E-04	10	.4393E-02	12	.3445E-03
90	.34654	.00703	29	12	.6318E-04	10	12	.4240E+01	.5085E-04	17	.1179E-02	10	.1572E-03
100	.34670	.00703	25	12	.4706E-04	10	12	.3643E+01	.3917E-04	17	.6060E-03	13	.1206E-03
110	.34683	.00703	8	5	.3855E-04	10	12	.3127E+01	.3062E-04	17	.6271E-03	16	.9317E-04
120	.34693	.00703	8	5	.3134E-04	10	12	.2579E+01	.2404E-04	17	.5031E-03	10	.4480E-04
130	.34700	.00703	8	6	.2481E-04	10	12	.2056E+01	.1883E-04	17	.5174E-03	12	.3616E-04
140	.34707	.00703	8	6	.1945E-04	10	12	.1614E+01	.1489E-04	17	.4245E-03	12	.3110E-04
150	.34711	.00703	8	6	.1537E-04	10	12	.1268E+01	.1178E-04	17	.3179E-03	12	.2190E-04
160	.34715	.00703	8	6	.1210E-04	10	12	.9984E+00	.9261E-05	17	.2236E-03	12	.1759E-04
170	.34718	.00703	8	6	.9471E-05	10	12	.7830E+00	.7283E-05	17	.1928E-03	12	.1465E-04

***** SOLUTION CONVERGED *****

INTERMEDIATE OUTPUT FOR FINE MESH

WE = 1.9500		EPS = .9000		MAXIT FOR THIS MESH = 500									
ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERRU	ICPL	CPERRL
10	.34784	.00843	22	21	.1502E-02	22	21	.3997E+03	.1223E-03	32	.1301E+01	19	.1377E+01
20	.34818	.00863	18	18	.8631E-03	17	24	.2342E+03	.8694E-04	33	.8339E-01	24	.1423E-01
30	.34846	.00871	16	13	.5411E-03	17	24	.1839E+03	.6761E-04	33	.2755E-01	20	.7369E-02
40	.34876	.00876	13	10	.3373E-03	17	24	.1338E+03	.6625E-04	33	.1233E-01	22	.4629E-02
50	.34909	.00881	16	8	.2559E-03	17	24	.9943E+02	.8924E-04	34	.7367E-02	21	.2729E-02
60	.34939	.00884	22	6	.1922E-03	17	24	.7891E+02	.7684E-04	34	.5420E-02	24	.1586E-02
70	.34966	.00885	15	6	.1620E-03	17	24	.6697E+02	.6681E-04	34	.3597E-02	24	.8228E-03
80	.34991	.00887	15	8	.1336E-03	17	24	.5874E+02	.6875E-04	34	.5249E-02	24	.5288E-03
90	.35016	.00891	16	23	.1150E-03	17	24	.5220E+02	.6231E-04	34	.5443E-02	24	.6467E-03
100	.35038	.00892	16	25	.9682E-04	17	24	.4351E+02	.5444E-04	34	.5184E-02	24	.7227E-03
110	.35057	.00894	16	25	.8170E-04	17	24	.3689E+02	.4820E-04	34	.4149E-02	24	.5016E-03
120	.35073	.00895	16	25	.6700E-04	17	24	.3018E+02	.4409E-04	34	.3341E-02	24	.4423E-03
130	.35090	.00895	14	23	.5502E-04	17	24	.2489E+02	.3939E-04	34	.2677E-02	24	.3741E-03
140	.35104	.00896	14	20	.4620E-04	17	24	.2086E+02	.3470E-04	34	.2081E-02	24	.2390E-03
150	.35116	.00896	14	14	.3943E-04	17	24	.1774E+02	.3063E-04	34	.1739E-02	24	.1622E-03

Figure 4.6.- Continued

160	.45127	.00897	10	8	.3511E-04	17	24	.1949E+02	.2739E-04	34	.1480E-02	24	.1649E-03
170	.35137	.00897	14	8	.3158E-04	17	24	.1365E+02	.2421E-04	34	.1232E-02	24	.8295E-04
180	.35145	.00897	15	9	.2801E-04	17	24	.1218E+02	.2141E-04	34	.1389E-02	24	.5460E-04
190	.35153	.00897	14	10	.2901E-04	17	24	.1098E+02	.1898E-04	34	.9305E-03	24	.4003E-04
200	.35159	.00898	14	11	.2228E-04	17	24	.9843E+01	.1686E-04	34	.8768E-03	24	.4295E-04
210	.35165	.00898	14	12	.1984E-04	17	24	.8786E+01	.1503E-04	34	.7859E-03	24	.4234E-04
220	.35171	.00898	14	13	.1757E-04	17	24	.7784E+01	.1336E-04	34	.7002E-03	24	.4176E-04
230	.35175	.00898	15	12	.1555E-04	17	24	.6895E+01	.1186E-04	34	.6328E-03	24	.3995E-04
240	.35180	.00898	14	11	.1386E-04	17	24	.6122E+01	.1053E-04	34	.5613E-03	24	.3403E-04
250	.35183	.00898	14	11	.1230E-04	17	24	.5436E+01	.9348E-05	34	.4991E-03	24	.2911E-04
260	.35187	.00898	14	11	.1094E-04	17	24	.4843E+01	.8294E-05	34	.4476E-03	24	.2570E-04
270	.35190	.00899	14	12	.9695E-05	17	24	.4293E+01	.7362E-05	34	.3901E-03	24	.2339E-04

***** SOLUTION CONVERGED *****

 * FINAL OUTPUT *

PRINTOUT IN PHYSICAL VARIABLES.

DEFINITION OF SIMILARITY PARAMETERS BY KRUPP

BOUNDARY CONDITION FOR SLOTTED WALL

DIFFERENCE EQUATIONS ARE FULLY CONSERVATIVE.

KUTTA CONDITION IS ENFORCED.

MACH = .7500000
 DELTA = .0660000
 ALPHA = 2.0000000
 K = 3.8061213
 DOUBLET STRENGTH = .0462305
 AIRFOIL VOLUME = .0410899

PARAMETERS USED TO TRANSFORM VARIABLES
 TO TRANSONIC SCALING

CPFACT = .1901683
 CDFACT = .0114101
 CMFACT = .1901603
 CLFACT = .1901603
 YFACT = 2.9495264
 VFACT = 3.4377467

Figure 4.6.- Continued

FINAL OUTPUT FOR FINE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.

CL = .351895
CM = .008985
CP* = -.603170

B = BEFORE OR BEHIND AIRFOIL
U = UPPER AIRFOIL SURFACE
L = LOWER AIRFOIL SURFACE
* = CRITICAL PRESSURE

I	X	LOWER Y=0-		UPPER Y=J+			
		CP	M1	CP	M1		
1	-1.075003	.016669	.741896	.016669	.741896		
2	-.950000	.019421	.740549	.019421	.740549	B	*
3	-.825000	.022801	.738892	.022801	.738892	B	*
4	-.700000	.027756	.736499	.027756	.736499	B	*
5	-.575000	.035234	.732784	.035234	.732784	B	*
6	-.450000	.046499	.727168	.046499	.727168	B	*
7	-.350000	.063082	.718850	.063082	.718850	B	*
8	-.250000	.088844	.705733	.088844	.705733	B	*
9	-.175000	.125833	.686461	.125833	.686461	B	*
10	-.125000	.176177	.659327	.176177	.659327	B	*
11	-.075000	.249250	.617827	.249250	.617827	B	*
12	-.052500	.332512	.566849	.332512	.566849	B	*
13	-.035000	.418874	.508602	.418874	.508602	B	*
14	-.022500	.510796	.438180	.510796	.438180	B	*
15	-.015000	.596657	.360172	.596657	.360172	B	*
16	-.007500	.677417	.266732	.677417	.266732	B	*
17	-.002500	.748961	.138754	.748961	.138754	B	*
AIRFOIL LEADING EDGE							
18	.002500	1.306960	0.000000	.101284	.699310	L	*
19	.007500	.929753	0.000000	-.405071	.925371	L	*
20	.012500	.712684	.213463	-.789476	1.065627	L	*
21	.017500	.565201	.392416	-1.027202	1.143488	L	*
22	.022500	.454480	.482546	-1.025281	1.142879	L	*
23	.027500	.372922	.540376	-1.035401	1.145643	L	*
24	.032500	.310278	.580900	-1.051762	1.151251	L	*
25	.037500	.255468	.614166	-1.051694	1.151230	L	*
26	.045000	.199869	.646164	-1.043091	1.148504	L	*
27	.055000	.148925	.674151	-1.033673	1.145539	L	*
28	.065000	.109046	.695273	-1.027877	1.143702	L	*
29	.075000	.078450	.711054	-1.023921	1.142447	L	*
30	.085000	.052623	.724107	-1.018759	1.140807	L	*
31	.097500	.027129	.736765	-1.009135	1.137744	L	*
32	.115000	-.000250	.756121	-.989874	1.131587	L	*
33	.140625	-.027365	.763118	-.941996	1.116137	L	*
34	.171875	-.049406	.773522	-.899662	1.094443	L	*
35	.203125	-.064523	.780577	-.855088	.944771	L	*
36	.234375	-.074153	.785039	-.808290	.932262	L	*
37	.265625	-.079822	.787667	-.757226	.933431	L	*
38	.296875	-.082633	.788946	-.702657	.926384	L	*
39	.328125	-.083200	.789247	-.644459	.917257	L	*
40	.359375	-.082061	.788684	-.581229	.907587	L	*
41	.390625	-.079594	.787548	-.513998	.897692	L	*
42	.421875	-.076083	.785930	-.442526	.888361	L	*
43	.453125	-.071744	.783925	-.367984	.879057	L	*
44	.484375	-.066738	.781606	-.287992	.869991	L	*
45	.515625	-.061185	.779025	-.206889	.861149	L	*
46	.546875	-.055165	.776217	-.126462	.852503	L	*
47	.578125	-.048728	.773204	-.046611	.844016	L	*
48	.609375	-.041892	.769991	-.072180	.835641	L	*
49	.640625	-.034645	.766570	-.103147	.827323	L	*
50	.671875	-.026941	.762916	-.139242	.818994	L	*
51	.703125	-.018697	.758987	-.180317	.810570	L	*
52	.734375	-.009781	.754715	-.226142	.801945	L	*
53	.765625	-.000000	.750100	-.276427	.792979	L	*
54	.796875	.010931	.744695	-.331787	.783462	L	*
55	.828125	.023440	.738579	-.392681	.773182	L	*
56	.859375	.037484	.731650	-.459541	.762203	L	*
57	.885000	.051138	.724850	-.531817	.752003	L	*
58	.904000	.062750	.719017	-.609039	.743619	L	*
59	.915000	.074192	.713222	-.691865	.735565	L	*
60	.930000	.087708	.706316	-.780893	.726272	L	*
61	.945000	.104256	.697767	-.876697	.715149	L	*
62	.960000	.125654	.686556	-.979899	.701064	L	*
63	.975000	.153784	.670451	-1.100092	.681552	L	*
64	.990000	.199113	.646588	-1.246521	.653613	L	*
65	1.000000	.257277	.612697	-.253477	.615341	L	*
AIRFOIL TRAILING EDGE							
66	1.010000	.225079	.631856	.225079	.631856	B	*
67	1.025000	.175309	.659805	.175309	.659805	B	*
68	1.050000	.133274	.682519	.133274	.682519	B	*
69	1.090000	.099656	.730154	.099656	.730154	B	*
70	1.150000	.074308	.723163	.074308	.723163	B	*
71	1.225000	.052668	.721978	.052668	.721978	B	*
72	1.300000	.044790	.728019	.044790	.728019	B	*
73	1.400000	.035714	.732527	.035714	.732527	B	*
74	1.500000	.028897	.735894	.028897	.735894	B	*
75	1.625000	.023731	.738436	.023731	.738436	B	*
76	1.750000	.020086	.740224	.020086	.740224	B	*
77	1.875000	.017179	.741646	.017179	.741646	B	*

Figure 4.6.- Continued

Y-GRID VALUES Y(T) J= 1 TO 48							
-2.000000	-1.800000	-1.600000	-1.400000	-1.200000	-1.000000	-.800000	-.600000
-.900000	-.700000	-.500000	-.300000	-.100000	.100000	.300000	.500000
-.100000	.100000	.300000	.500000	.700000	.900000	1.100000	1.300000
.010000	.030000	.050000	.075000	.100000	.125000	.150000	.180000
.210000	.240000	.270000	.300000	.340000	.390000	.450000	.500000
.650000	.800000	1.000000	1.200000	1.400000	1.600000	1.800000	2.000000

SONIC LINE OUTPUT

SONIC LINE COORDINATES

Y	X-SONIC	
.180000	.11078	.13374
.150000	.08704	.11473
.125000	.07326	.10311
.100000	.05973	.10900
.075000	.04625	.17383
.050000	.03394	.17785
.030000	.02284	.18058
.010000	.01204	.18304

BODY LOCATION

SHOCK WAVE DRAG AND TOTAL PRESSURE PROFILE OUTPUT

INVISCID WAKE PROFILES FOR INDIVIDUAL SHOCK WAVES WITHIN MOMENTUM CONTOUR

SHOCK 1
WAVE DRAG FOR THIS SHOCK = .000342
Y CD(Y) PO/POINF

0.00000000	.00387993	.99747253
.01000000	.00368048	.99760245
.03000000	.00336232	.99784879
.05000000	.00284000	.99814996
.07500000	.00220718	.99856219
.10000000	.00162955	.99893847
.12500000	.00102354	.99933324
.15000000	.00058815	.99961687
.18000000	.00001299	.99999134

DRAG COEFFICIENT OUTPUT

CALCULATION OF DRAG COEFFICIENT BY MOMENTUM INTEGRAL METHOD

BOUNDARIES OF CONTOUR USED CONTRIBUTION TO CD

UPSTREAM	X = -1.75000	CDUP = -.001618
DOWNSTREAM	X = 1.225000	CDDOWN = .001962
TOP	Y = 1.800000	CDTOP = .000000
BOTTOM	Y = -1.800000	CDBOT = .000000
TOTAL CONTRIBUTIONS AROUND CONTOUR		= .003315

THERE ARE 1 SHOCKS INSIDE CONTOUR. TOTAL CDWAVE = .000342

NOTE - ALL SHOCKS CONTAINED WITHIN CONTOUR
CDWAVE EQUALS TOTAL WAVE DRAG

DRAG CALCULATED FROM MOMENTUM INTEGRAL CD = .000357

TIME TO RUN CASE WAS 13.41 SECONDS.

(b) Output.
Figure 4.6.- Concluded

SAMPLE CASE 4 - SUPERSONIC-NACA0006 AIRFOIL-FREE AIR SIMULATION
 \$INP
 EMACH=1.1, BCTYPE=1, AMESH=.TRUE., BCFOIL=1, ALPHA=2.0,
 DELTA=0.06,
 \$END

(a) Input.

SAMPLE CASE 4 - SUPERSONIC - NACA0006 AIRFOIL - FREE AIR SIMULATION

INPUT PARAMETERS

EMACH = 1.10000	POR = 0.00000	IMIN = 1	BCTYPE = 1	AMESH = T
DELTA = .06000	CLSET = 0.00000	IMAXI = 81	BCFOIL = 1	PHYS = T
ALPHA = 2.00000	EPS = .20000	JMIN = 1	PSTART = 1	PSAVE = F
AK = 0.00000	RIGF = 0.00000	JMAXI = 64	PRTFLD = 1	KUTTA = T
GAM = 1.40000	WCIRC = 1.00000	MAXIT = 500	IPRTER = 10	FCR = T
F = 0.00000	CVERGE = .00001	NU = 100	SIMDEF = 3	
H = 0.00000	DVERGE = 10.0	NL = 75	ICUT = 2	

WE = 1.86, 1.90, 1.95

XIN							
-4.826958	-1.472160	-1.170661	-.972966	-.829857	-.653409	-.479854	-.295092
-1.178882	-.114702	-.075551	-.050198	-.032930	-.020518	-.011003	-.003465
.003024	.008803	.014240	.019568	.025061	.030961	.037314	.044213
.054966	.067013	.082363	.101517	.124201	.149372	.178800	.202963
.229179	.253353	.281029	.306213	.330950	.355299	.379323	.403081
.426629	.450015	.473280	.496462	.519582	.542689	.565741	.588707
.611881	.634971	.658084	.681229	.704417	.727605	.750888	.774418
.797988	.821742	.845736	.870039	.894736	.919931	.945757	.972377
1.000000	1.028899	1.059441	1.092135	1.127723	1.167351	1.212946	1.268145
1.341184	1.456833	1.671514	1.876313	2.042361	2.209055	2.406774	2.702497
4.800373							
YIN							
-5.200000	-2.594100	-1.722837	-1.285224	-1.021060	-.843611	-.725701	-.618741
-5.542402	-.480484	-.429040	-.385439	-.347855	-.314984	-.285009	-.259767
-.236188	-.214639	-.194799	-.176391	-.159191	-.143033	-.127712	-.113224
-.099167	-.085732	-.072733	-.060093	-.047742	-.035615	-.023853	-.012450
.011860	.023653	.035615	.047742	.060093	.072733	.085732	.099167
.113124	.127702	.143013	.159191	.176391	.194799	.214539	.236188
.259787	.285369	.314984	.347855	.385439	.429040	.480484	.542402
.618741	.715761	.843611	1.021060	1.265224	1.722837	2.594100	5.200000

Figure 4-7.- Sample test case 4 - supersonic, free air simulation
 (abbreviated output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2^\circ$
 $M_\infty = 1.1$

INTERMEDIATE OUTPUT FOR COARSE MESH

WE = 1.8000 EPS = .2000 MAXIT FOR THIS MESH = 125

ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERRU	ICPL	CPERRL
10	.20042	-.03457	21	8	.2493E+00	5	8	.1462E+03	.1586E+00	16	.9529E+00	5	.9323E+03
20	.24402	-.04272	21	8	.5993E+02	19	8	.2842E+03	.3296E+00	17	.1306E+00	17	.2731E+00
30	.24571	-.04315	21	8	.6895E+00	19	8	.3269E+03	.3786E+00	7	.1666E-01	17	.3869E-01
40	.24593	-.04329	21	8	.7282E+00	19	8	.3450E+03	.3993E+00	7	.9445E-02	17	.1759E-01
50	.24609	-.04337	21	8	.7485E+00	19	8	.3546E+03	.4162E+00	7	.5367E-02	17	.9690E-02
60	.24628	-.04340	21	8	.7597E+00	19	8	.3599E+03	.4163E+00	7	.3297E-02	17	.5582E-02
70	.24651	-.04343	21	8	.7659E+00	19	8	.3629E+03	.4197E+00	7	.1904E-02	17	.3323E-02
80	.24675	-.04344	21	8	.7694E+00	19	8	.3646E+03	.4217E+00	7	.1377E-02	17	.2661E-02
90	.24699	-.04345	21	8	.7714E+00	19	8	.3655E+03	.4229E+00	7	.8639E-03	17	.1344E-02
100	.24722	-.04345	21	8	.7726E+00	19	8	.3662E+03	.4236E+00	7	.4274E-03	17	.9347E-03
110	.24746	-.04346	21	8	.7733E+00	19	8	.3666E+03	.4240E+00	7	.2917E-03	17	.7025E-03
120	.24769	-.04347	21	8	.7737E+00	19	8	.3669E+03	.4243E+00	7	.2142E-03	17	.5756E-03

***** ITERATION LIMIT REACHED *****

INTERMEDIATE OUTPUT FOR MEDIUM MESH

WE = 1.9000 EPS = .2000 MAXIT FOR THIS MESH = 250

ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERRU	ICPL	CPERRL
10	.20335	-.04097	7	3	.7331E-02	15	4	.9920E+02	.1117E-02	9	.9316E+00	9	.1051E+01
20	.20638	-.04052	3	2	.5565E-02	9	16	.8568E+02	.1369E-02	12	.2749E-01	11	.2993E-01
30	.20844	-.04075	3	31	.2810E-02	9	16	.6194E+02	.2154E-03	12	.1124E-01	9	.2590E-02
40	.20930	-.04087	8	5	.2218E-02	9	16	.8346E+02	.2110E-03	13	.7507E-03	11	.2425E-02
50	.20959	-.04086	11	17	.1931E-02	9	16	.7515E+02	.2549E-04	12	.9336E-02	12	.1298E-02
60	.20964	-.04084	11	17	.1658E-02	9	16	.6432E+02	.3121E-03	12	.6797E-02	11	.2239E-02
70	.20961	-.04083	11	17	.1424E-02	9	16	.5472E+02	.1176E-04	12	.3713E-02	11	.2119E-02
80	.20957	-.04082	11	17	.1294E-02	9	16	.4612E+02	.1733E-04	12	.3023E-02	11	.1972E-02
90	.20952	-.04081	11	17	.1013E-02	9	16	.3871E+02	.1911E-04	12	.4262E-02	11	.1734E-02
100	.20945	-.04080	11	17	.8491E-03	9	16	.3242E+02	.2239E-04	12	.3376E-02	11	.1480E-02
110	.20938	-.04078	11	17	.7106E-03	9	16	.2713E+02	.3010E-04	12	.2993E-02	11	.1294E-02
120	.20929	-.04075	11	17	.5941E-03	9	16	.2264E+02	.3796E-04	12	.2522E-02	11	.1104E-02
130	.20914	-.04070	20	31	.5555E-03	9	16	.1890E+02	.5612E-04	12	.2180E-02	11	.8925E-03
140	.20890	-.04061	20	31	.6622E-03	9	16	.1586E+02	.1097E-03	12	.1855E-02	11	.7521E-03
150	.20853	-.04048	20	31	.4417E-03	9	16	.1325E+02	.1192E-03	12	.1230E-02	11	.5344E-03
160	.20832	-.04043	11	31	.3168E-03	9	16	.1122E+02	.5837E-04	12	.7533E-03	11	.3417E-03
170	.20820	-.04043	11	17	.2534E-03	9	16	.9600E+01	.3379E-04	12	.6615E-03	11	.4766E-03
180	.20815	-.04044	11	17	.2148E-03	9	16	.8136E+01	.1126E-04	12	.2391E-03	11	.4237E-03
190	.20813	-.04045	11	17	.1786E-03	9	16	.6772E+01	.2394E-05	12	.2430E-03	11	.3546E-03
200	.20813	-.04045	11	17	.1474E-03	9	16	.5597E+01	.1071E-05	12	.6600E-03	11	.2821E-03
210	.20813	-.04045	11	17	.1222E-03	9	16	.4639E+01	.1731E-05	12	.5599E-03	11	.2206E-03
220	.20812	-.04045	11	17	.1010E-03	9	16	.3897E+01	.1991E-05	12	.4339E-03	11	.1892E-03
230	.20812	-.04045	11	17	.8447E-04	9	16	.3208E+01	.1191E-05	12	.3543E-03	11	.1543E-03
240	.20812	-.04045	11	17	.7619E-04	9	16	.2665E+01	.9100E-06	12	.2975E-03	11	.1297E-03
250	.20811	-.04045	11	17	.5830E-04	9	16	.2213E+01	.7525E-06	12	.2497E-03	11	.1071E-03

***** ITERATION LIMIT REACHED *****

Figure 4.7.- Continued

INTERMEDIATE OUTPUT FOR FINE MESH

ME = 1.9500 EPS = .2000 MAXIT FOR THIS MESH = 500

ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERRU	ICPL	CPERRL
10	.20800	-.04152	8	33	.3659E-02	17	32	.4072E+03	.4022E-03	17	.1117E+J1	17	.1205E+J1
20	.20842	-.04162	20	2	.2342E-12	17	32	.3603E+03	.5969E-04	22	.1483E-01	20	.6974E-J2
30	.20867	-.04169	16	2	.2189E-02	17	32	.3412E+03	.1103E-33	22	.3726E-02	21	.2398E-02
40	.20900	-.04175	15	63	.1956E-02	17	32	.3311E+03	.1033E-03	22	.5761E-J3	21	.1220E-02
50	.20925	-.04180	7	63	.1974E-02	17	32	.3232E+03	.9063E-04	17	.4130E-03	21	.8942E-03
60	.20964	-.04192	7	63	.2111E-02	17	32	.3177E+03	.1520E-03	63	.6989E-03	61	.7533E-03
70	.21010	-.04203	7	63	.1849E-02	17	32	.3139E+03	.1642E-03	22	.7163E-03	55	.7985E-J3
80	.21050	-.04210	7	63	.1687E-02	17	32	.3094E+03	.1295E-03	65	.4117E-03	29	.7730E-03
90	.21081	-.04211	7	2	.1570E-02	17	32	.3003E+03	.9688E-J4	22	.5194E-J3	23	.1168E-02
100	.21102	-.04209	7	2	.1534E-02	17	32	.2898E+03	.6191E-04	22	.6130E-J3	22	.1382E-J2
110	.21114	-.04206	7	2	.1490E-02	17	32	.2807E+03	.3105E-04	22	.4989E-03	23	.1237E-02
120	.21118	-.04204	18	35	.1403E-02	17	32	.2729E+03	.8525E-05	22	.4905E-03	21	.3446E-03
130	.21118	-.04201	18	34	.1370E-02	17	32	.2670E+03	.3906E-05	22	.5349E-03	21	.6682E-J3
140	.21116	-.04199	18	33	.1341E-02	17	32	.2617E+03	.1122E-04	22	.3486E-03	65	.3126E-03
150	.21112	-.04197	6	59	.1318E-02	17	32	.2562E+03	.1371E-04	22	.4972E-J3	65	.3111E-03
160	.21108	-.04196	6	5	.1290E-02	17	32	.2500E+03	.1213E-04	22	.8334E-J3	22	.5493E-03
170	.21106	-.04195	17	33	.1242E-02	17	32	.2428E+03	.7663E-05	22	.3961E-03	22	.6203E-03
180	.21105	-.04194	17	33	.1204E-02	17	32	.2353E+03	.7947E-06	22	.1342E-J4	22	.5567E-J4
190	.21105	-.04194	7	3	.1174E-02	17	32	.2274E+03	.2841E-35	22	.1372E-02	22	.7020E-J3
200	.21105	-.04194	17	33	.1121E-02	17	32	.2191E+03	.4266E-05	22	.1138E-02	22	.7484E-03
210	.21107	-.04194	17	33	.1076E-02	17	32	.2103E+03	.1989E-04	22	.1103E-02	22	.8348E-03
220	.21112	-.04195	17	33	.1029E-02	17	32	.2011E+03	.1948E-04	22	.1138E-02	22	.8817E-03
230	.21119	-.04195	17	33	.9823E-03	17	32	.1919E+03	.2638E-04	22	.1194E-02	22	.9117E-03
240	.21128	-.04196	17	33	.9361E-03	17	32	.1829E+03	.3127E-04	22	.1102E-02	22	.9127E-03
250	.21139	-.04198	18	33	.8913E-03	17	32	.1741E+03	.4182E-04	22	.8391E-03	22	.9131E-03
260	.21152	-.04199	18	33	.8488E-03	17	32	.1657E+03	.4887E-04	22	.7453E-03	22	.8937E-03
270	.21167	-.04200	13	33	.8087E-03	17	32	.1578E+03	.5334E-04	22	.6116E-03	22	.8842E-03
280	.21182	-.04202	18	34	.7735E-03	17	32	.1504E+03	.5200E-04	22	.5109E-03	22	.8303E-03
290	.21197	-.04203	18	34	.7345E-03	17	32	.1434E+03	.5151E-04	22	.4279E-03	21	.7850E-03
300	.21212	-.04205	18	34	.7006E-03	17	32	.1368E+03	.5187E-04	22	.4149E-03	21	.7202E-03
310	.21227	-.04208	8	63	.6727E-03	17	32	.1309E+03	.5478E-04	22	.3533E-03	21	.6594E-03
320	.21243	-.04210	8	63	.6470E-03	17	32	.1253E+03	.5755E-04	22	.2932E-03	21	.6106E-03
330	.21260	-.04212	8	63	.6239E-03	17	32	.1200E+03	.5968E-04	22	.2191E-03	21	.5596E-03
340	.21278	-.04215	8	63	.6033E-03	17	32	.1151E+03	.6170E-04	63	.1863E-03	21	.5373E-03
350	.21296	-.04218	8	63	.5845E-03	17	32	.1105E+03	.6371E-04	17	.1773E-03	21	.5443E-03
360	.21314	-.04221	8	63	.5677E-03	17	32	.1064E+03	.6591E-04	17	.1813E-03	21	.5717E-03
370	.21333	-.04224	8	63	.5528E-03	17	32	.1026E+03	.6812E-04	17	.1709E-03	21	.5601E-03
380	.21353	-.04227	8	63	.5395E-03	17	32	.9910E+02	.7194E-04	17	.1607E-03	21	.4337E-03
390	.21374	-.04231	19	62	.5274E-03	17	32	.9550E+02	.7299E-04	17	.1809E-03	21	.4209E-03
400	.21386	-.04232	16	61	.4732E-03	17	32	.9176E+02	.2842E-04	17	.1727E-03	21	.4146E-03
410	.21387	-.04230	8	2	.4572E-03	17	32	.8812E+02	.6921E-05	63	.2336E-03	21	.3726E-03
420	.21380	-.04226	8	2	.4533E-03	17	32	.8445E+02	.3613E-04	63	.2036E-03	21	.3314E-03
430	.21365	-.04222	9	2	.4496E-03	17	32	.8066E+02	.5889E-04	22	.3245E-03	21	.2942E-03
440	.21345	-.04217	8	2	.4464E-03	17	32	.7689E+02	.7515E-04	22	.4311E-03	21	.2432E-03
450	.21322	-.04212	8	2	.4436E-03	17	32	.7312E+02	.8509E-04	22	.6244E-03	22	.1820E-03
460	.21296	-.04207	8	2	.4413E-03	17	32	.6927E+02	.9333E-04	22	.6933E-03	63	.1536E-03
470	.21270	-.04203	8	2	.4385E-03	17	32	.6524E+02	.9303E-04	22	.5133E-03	63	.1572E-03
480	.21243	-.04198	8	2	.4363E-03	17	32	.6127E+02	.9444E-04	22	.6003E-03	17	.1369E-03
490	.21216	-.04193	8	2	.4329E-03	17	32	.6058E+02	.9582E-04	22	.6359E-03	17	.1516E-03
500	.21188	-.04188	9	2	.4199E-03	17	32	.5787E+02	.9712E-04	22	.6139E-03	.7	.1635E-03

***** ITERATION LIMIT REACHED *****

* FINAL OUTPUT *

PRINTOUT IN PHYSICAL VARIABLES.

DEFINITION OF SIMILARITY PARAMETERS BY KRUPP

BOUNDARY CONDITION FOR FREE AIR

DIFFERENCE EQUATIONS ARE FULLY CONSERVATIVE.

KUTTA CONDITION IS ENFORCED.

MACH = 1.100000
DELTA = .600000
ALPHA = 2.000000
K = -1.2456397

PARAMETERS USED TO TRANSFORM VARIABLES
TO TRANSONIC SCALING

CPFACT = .1426887
CDFACT = .6085613
CMFACT = .1426887
CLFACT = .1426887
YFACT = 2.4354913
VFACT = 3.4377467

Figure 4.7.- Continued

FINAL OUTPUT FOR FINE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.

CL = .211883
CM = -.041882
CP* = .148116

B = BEFORE OR BEHIND AIRFOIL
U = UPPER AIRFOIL SURFACE
L = LOWER AIRFOIL SURFACE
* = CRITICAL PRESSURE

I	X	LOWER Y=0-		UPPER Y=0+			
		CP	M1	CP	M1		
1	-4.826958	-0.000600	1.100000	-0.000000	1.100000		* B
2	-1.472160	-0.000000	1.100000	-0.000000	1.100000		* B
3	-1.170661	.000000	1.100000	.000000	1.100000		* B
4	-.972966	.000000	1.100000	.000000	1.100000		* B
5	-.809857	.009657	1.093759	.009657	1.093759		* B
6	-.653469	.165719	.987442	.165719	.937442		* B
7	-.479854	.335488	.856937	.335488	.856937		* B
8	-.299092	.391359	.809399	.391359	.809399		* B
9	-.178882	.458584	.748207	.458584	.748207		* B
10	-.114702	.527590	.679689	.527590	.679689		* B
11	-.075551	.597217	.602709	.597217	.602709		* B
12	-.050198	.669409	.516788	.669409	.510788		* B
13	-.032930	.745001	.392083	.745001	.392083		* B
14	-.020518	.825603	.260750	.825013	.200750		* B
15	-.011063	.911035	0.000000	.911035	0.000000		* B
16	-.003465	1.005732	0.000000	1.005732	0.000000		* B
17	.003024	1.205373	0.000000	.891538	0.000000	L	* U
18	.008803	.801151	.272247	.451359	.755304		* U
19	.014240	.614255	.582325	.259810	.917409		* U
20	.019568	.502088	.705787	.154067	.995772		* U
21	.025061	.426305	.778190	.073993	1.051294		* U
22	.030961	.370393	.827592	.025077	1.083719		* U
23	.037614	.325600	.865079	.006299	1.095991		* U
24	.045413	.286976	.896171	-.017892	1.111471		* U
25	.054966	.251249	.924000	-.042969	1.127392		* U
26	.067013	.218493	.930321	-.066904	1.142006		* U
27	.082363	.181768	.929852	-.089946	1.156345		* U
28	.101517	.146271	1.001366	-.113000	1.170561		* U
29	.124201	.116795	1.026116	-.134861	1.183726		* U
30	.149372	.080210	1.047033	-.154744	1.195574		* U
31	.175806	.055675	1.063515	-.172279	1.205927		* U
32	.202583	.034856	1.077363	-.187404	1.214832		* U
33	.229179	.017186	1.088868	-.200598	1.222461		* U
34	.255353	.002066	1.098668	-.211931	1.229015		* U
35	.281029	-.011002	1.107068	-.221731	1.234684		* U
36	.306213	-.022414	1.114351	-.230495	1.239625		* U
37	.330950	-.032473	1.120732	-.238012	1.243968		* U
38	.355299	-.041436	1.126375	-.244774	1.247815		* U
39	.379323	-.049426	1.131405	-.250826	1.251249		* U
40	.403081	-.056650	1.135922	-.256280	1.254335		* U
41	.426629	-.063204	1.140005	-.261224	1.257126		* U
42	.450015	-.069181	1.143715	-.265729	1.259664		* U
43	.473282	-.074658	1.147106	-.269856	1.261985		* U
44	.496460	-.079763	1.150219	-.273656	1.264117		* U
45	.519582	-.084375	1.153095	-.277172	1.266087		* U
46	.542669	-.088726	1.155767	-.280444	1.267918		* U
47	.565740	-.092806	1.158267	-.283508	1.269630		* U
48	.588807	-.096660	1.160623	-.286398	1.271243		* U
49	.611881	-.100329	1.162862	-.289145	1.272774		* U
50	.634971	-.103892	1.165007	-.291780	1.274241		* U
51	.658084	-.107262	1.167061	-.294332	1.275659		* U
52	.681229	-.110593	1.169103	-.296808	1.277046		* U
53	.704417	-.113879	1.171149	-.299259	1.278415		* U
54	.727663	-.117150	1.173071	-.301760	1.279781		* U
55	.750988	-.120446	1.175058	-.304248	1.281158		* U
56	.774418	-.123784	1.177074	-.306785	1.282561		* U
57	.797988	-.127220	1.179141	-.309400	1.284006		* U
58	.821742	-.130788	1.181284	-.312120	1.285507		* U
59	.845736	-.134531	1.183528	-.314979	1.287082		* U
60	.870039	-.138498	1.185902	-.318012	1.288752		* U
61	.894736	-.142744	1.188438	-.321263	1.290539		* U
62	.919931	-.147334	1.191172	-.324783	1.292471		* U
63	.945737	-.152342	1.194149	-.328632	1.294581		* U
64	.972377	-.157863	1.197422	-.332891	1.296930		* U
65	1.000000	-.163908	1.167088	-.204855	1.224927		* U
66	1.028899	-.078862	1.149731	-.078862	1.149731		* B
67	1.059441	-.051269	1.132559	-.051269	1.132559		* B
68	1.092135	-.044426	1.128269	-.044426	1.128269		* B
69	1.127723	-.041877	1.126665	-.041877	1.126665		* B
70	1.167351	-.040240	1.125634	-.040240	1.125634		* B
71	1.212946	-.038992	1.124848	-.038992	1.124848		* B
72	1.268145	-.038174	1.124332	-.038174	1.124332		* B
73	1.341184	-.038225	1.124365	-.038225	1.124365		* B
74	1.456833	-.039577	1.125217	-.039577	1.125217		* B
75	1.671514	-.038742	1.124691	-.038742	1.124691		* B
76	1.876312	-.032682	1.126852	-.032682	1.120852		* B
77	2.042561	-.023734	1.115191	-.023734	1.115191		* B
78	2.209055	-.015180	1.109739	-.015180	1.109739		* B
79	2.406774	-.010460	1.106721	-.010460	1.106721		* B
80	2.702997	-.022405	1.114345	-.022405	1.114345		* B
81	4.890373	-.048029	1.130529	-.048029	1.130529		* B

Figure 4.7.- Continued

Y-GRID VALUES		Y(J) J= 1 TO 64					
-5.200000	-2.594100	-1.722837	-1.285224	-1.021060	-.843611	-.715701	-.618741
-.542402	-.480484	-.429040	-.385439	-.347855	-.314984	-.285869	-.259787
-.236188	-.214639	-.194799	-.176391	-.159191	-.143013	-.127702	-.113124
-.099167	-.085732	-.072733	-.060093	-.047742	-.035615	-.023653	-.011800
.011800	.023653	.035615	.047742	.060093	.072733	.085732	.099167
.113124	.127702	.143013	.159191	.176391	.194799	.214639	.236188
.259787	.285869	.314984	.347855	.385439	.429040	.480484	.542402
.618741	.715701	.843611	1.021060	1.285224	1.722837	2.594100	5.200000

SONIC LINE OUTPUT

SONIC LINE COORDINATES

Y	X-SONIC	
2.59410	-.27198	.49553
1.72284	-.43613	.43866
1.28522	-.52395	.35835
1.02106	-.56660	.30646
.84361	-.59380	.27041
.71570	-.61232	.24028
.61874	-.62540	.21731
.54240	-.63493	.19866
.48048	-.64203	.18360
.42904	-.64744	.17036
.38544	-.65163	.15928
.34786	-.65494	.14896
.31498	-.65798	.14051
.28587	-.66069	.13261
.25979	-.66313	.12487
.23619	-.66528	.11832
.21464	-.66709	.11214
.19480	-.66851	.10597
.17639	-.66960	.09993
.15919	-.67034	.09471
.14301	-.67074	.08941
.12770	-.67083	.08378
.11312	-.67054	.07866
.09917	-.66990	.07364
.08573	-.66840	.06861
.07273	-.66697	.06281
.06009	-.67006	.05705
.04774	-.67034	.05109
.03561	-.67058	.04429
.02365	-.67079	.03677
.01180	-.67097	.02798
BODY LOCATION		
-.01180	-.67125	.11132
-.02365	-.67135	.12039
-.03561	-.67142	.12962
-.04774	-.67147	.13794
-.06009	-.67149	.14460
-.07273	-.67148	.15168
-.08573	-.67144	.15940
-.09917	-.67136	.16640
-.11312	-.67124	.17292
-.12770	-.67108	.18000
-.14301	-.67087	.18736
-.15919	-.67059	.19436
-.17639	-.67025	.20119
-.19480	-.66982	.20911
-.21464	-.66929	.21710
-.23619	-.66862	.22503
-.25979	-.66780	.23383
-.28587	-.66677	.24336
-.31498	-.66547	.25366
-.34786	-.66380	.26439
-.38544	-.66165	.27640
-.42904	-.65878	.29019
-.48048	-.65487	.30519
-.54240	-.64940	.32332
-.61874	-.64169	.34473
-.71570	-.63059	.37075
-.84361	-.61418	.40162
-1.02106	-.58911	.44366
-1.28522	-.54893	.49611
-1.72284	-.47480	.56551
-2.59410	-.30048	.63440

Figure 4.7.- Continued

SHOCK WAVE DRAG AND TOTAL PRESSURE PROFILE OUTPUT

INVISCID WAKE PROFILES FOR INDIVIDUAL SHOCK WAVES WITHIN MOMENTUM CONTOUR

SHOCK 1 WAVE DRAG FOR THIS SHOCK= Y	CD(Y)	.007130 PD/POINF
-2.59410019	.00101194	.99927456
-1.72283680	.00085003	.99939063
-1.28522355	.00138639	.99900612
-1.02106025	.00160990	.99884618
-.84361099	.00179065	.99871632
-.71570104	.00195214	.99860055
-.61874102	.00209990	.99849462
-.54240166	.00223505	.99839717
-.48048365	.00181152	.99870136
-.42904031	.00190076	.99863739
-.38543890	.00197252	.99858594
-.34785516	.00203096	.99854405
-.31498449	.00207900	.99850361
-.28586858	.00211879	.99846178
-.25978714	.00215191	.99842534
-.23618771	.00217998	.99838796
-.21463894	.00220274	.99835070
-.19479857	.00222211	.99831311
-.17639103	.00223829	.99827541
-.15919133	.00225174	.99823857
-.14301344	.00226283	.99819783
-.12776160	.00227186	.99815735
-.11312379	.00227907	.99811618
-.09916674	.00228468	.998076216
-.08573207	.00228882	.998035919
-.07273320	.00229164	.9980005717
-.06009296	.00229324	.997963603
-.04774152	.00229369	.997925570
-.03561482	.00229305	.99788616
-.02365311	.00229137	.997845736
-.01179972	.00228868	.99780529
.01179972	.00228027	.99776532
.02365311	.00227454	.997725943
.03561482	.00226775	.997687430
.04774152	.00225986	.99764995
.06009296	.00225080	.997613645
.07273320	.00224050	.997578383
.08573207	.00222884	.997544219
.09916674	.00221572	.997511100
.11312379	.00220096	.997478218
.12770160	.00218439	.997445406
.14301344	.00216578	.997412740
.15919133	.00214485	.997380240
.17639103	.00212127	.997347930
.19479857	.00209464	.997315839
.21463894	.00206445	.997283974
.23618771	.00203007	.997252468
.25978714	.00199072	.997221289
.28586858	.00194540	.997190538
.31498449	.00189285	.997160307
.34785516	.00183139	.997130616
.38543890	.00234772	.99831697
.42904031	.00224426	.99839113
.48048365	.00213317	.99847078
.54240166	.00201386	.99855631
.61874102	.00188553	.99864830
.71570104	.00174673	.99874781
.84361099	.00159398	.99885731
1.02106025	.00141739	.99898396
1.28522355	.00118417	.99915109
1.72283680	.00086469	.99942313
2.59410019	.00067853	.99951501

SHOCK WAVE EXTENDS OUTSIDE CONTOUR.
PRINTOUT OF SHOCK LOSSES ARE NOT AVAILABLE FOR REST OF SHOCK

DRAG COEFFICIENT OUTPUT

CALCULATION OF DRAG COEFFICIENT BY MOMENTUM INTEGRAL METHOD

BOUNDARIES OF CONTOUR USED	CONTRIBUTION TO CD
UPSTREAM X = -1.170661	CDUP = 0.000000
DOWNSTREAM X = .727663	CDOWN = .031358
TOP Y = 2.594100	CDTOP = .034202
BOTTOM Y = -2.594100	CDBOT = .035083
BODY AFT OF X = .727663	CDBODY = .007620
TOTAL CONTRIBUTIONS AROUND CONTOUR =	.027963

THERE ARE 1 SHOCKS INSIDE CONTOUR. TOTAL CDWAVE = .007130

NOTE - ONE OR MORE SHOCKS EXTEND OUTSIDE OF CONTOUR
CDWAVE DOES NOT EQUAL TOTAL WAVE DRAG

DRAG CALCULATED FROM MOMENTUM INTEGRAL CD = .035093

TIME TO RUN CASE WAS 32.44 SECONDS.

(b) Output.
Figure 4.7.- Concluded

SAMPLE CASE 5 - SUPERSONIC-NACA0006 AIRFOIL-FREE JET SIMULATION
 \$INP
 BCTYPE=3, H=2.0, EMACH=1.1, AMESH=.TRUE., ALPHA=2.0, DELTA=0.06,
 BCFOIL=1,
 \$END

(a) Input.

SAMPLE CASE 5 - SUPERSONIC - NACA0006 AIRFOIL - FREE JET SIMULATION

INPUT PARAMETERS

EMACH = 1.10000 POR = 0.00000 IMIN = 1 BCTYPE = 3 AMESH = T
 DELTA = .06000 C1SET = 0.50000 IMAXI = 81 BCFOIL = 1 PHYS = T
 ALPHA = 2.00000 EPS = .20000 JMIN = 1 PSTART = 1 PSAVE = F
 AK = 0.00000 RIGF = 0.00000 JMAXI = 64 PRFLO = 1 KUTTA' = T
 GAM = 1.40000 WCIRC = 1.00000 MAXIT = 500 IPRTF = 10 FCR = T
 F = 0.00000 CVERGE = .00001 NU = 100 SIMDEF = 3
 H = 2.00000 DVERGE = 10.0 NL = 75 ICUT = 2

WE = 1.80,1.90,1.95

XIN

-4.826958	-1.472160	-1.170661	-.972966	-.8.9857	-.553469	-.479804	-.295092
-.178882	-.114702	-.075551	-.050198	-.032930	-.020518	-.011003	-.003405
.003024	.008603	.014240	.019568	.025961	.030961	.037514	.044413
.054966	.067013	.082363	.101517	.124201	.149372	.175306	.202534
.229179	.255353	.281029	.306213	.330990	.355299	.379323	.403381
.428629	.453015	.473280	.496460	.519582	.542669	.565740	.588307
.611881	.634971	.656084	.681229	.704417	.727663	.750988	.774418
.797988	.821742	.845736	.873339	.894736	.919931	.945737	.972377
1.000000	1.028899	1.059441	1.092135	1.127723	1.167351	1.212940	1.268.45
1.341184	1.456833	1.671514	1.876313	2.042561	2.209055	2.400774	2.702997
4.890373							

YIN

-1.000000	-.906455	-.821465	-.744089	-.673514	-.609036	-.550046	-.496200
-.446463	-.403997	-.359253	-.323914	-.285702	-.253369	-.223096	-.196488
-.171573	-.148797	-.128029	-.109135	-.092719	-.076584	-.060274	-.050420
-.039566	-.030108	-.022004	-.015212	-.009701	-.005441	-.002413	-.000600
.000603	.002413	.005441	.009701	.015212	.022004	.030108	.039566
.050426	.060274	.076584	.092719	.109135	.128025	.148797	.171573
.196488	.223096	.253369	.285702	.320914	.359253	.400797	.446488
.496010	.550646	.609036	.673514	.744089	.821465	.906455	1.000000

Figure 4-8.- Sample test case 5 - supersonic, free jet simulation
 (abbreviated output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2^\circ$,
 $M_\infty = 1.1$, $H = 2$.

INTERMEDIATE OUTPUT FOR COARSE MESH

WE = 1.8000 EPS = .2000 MAXIT FOR THIS MESH = 125

ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERRU	ICPL	CPERRL
19	.29775	-.05325	20	8	.4890E+00	19	9	.2022E+04	.6227E+00	16	.1481E+01	17	.2613E+01
20	.23063	-.04624	20	8	.3057E+00	19	9	.1396E+04	.4300E+00	17	.1172E+00	17	.2672E+00
30	.21871	-.04396	20	8	.2199E+00	19	9	.1055E+04	.3237E+00	17	.2847E+01	17	.3594E+01
40	.22180	-.04418	20	8	.2049E+00	19	9	.1115E+04	.3408E+00	6	.5387E+02	17	.1495E+01
50	.22230	-.04427	20	8	.2081E+00	19	9	.1136E+04	.3472E+00	5	.3617E+02	17	.6052E+02
60	.22250	-.04426	20	8	.2072E+00	19	9	.1140E+04	.3482E+00	5	.1766E+02	7	.1398E+02
70	.22246	-.04424	20	8	.2063E+00	19	9	.1140E+04	.3481E+00	5	.6315E+03	7	.6876E+03
80	.22245	-.04424	20	8	.2058E+00	19	9	.1140E+04	.3481E+00	5	.4169E+03	7	.3419E+03
90	.22244	-.04423	20	8	.2056E+00	19	9	.1140E+04	.3482E+00	5	.2120E+03	7	.1713E+03
100	.22244	-.04423	20	8	.2055E+00	19	9	.1140E+04	.3482E+00	5	.1378E+03	7	.6822E+04
110	.22244	-.04423	20	8	.2054E+00	19	9	.1140E+04	.3482E+00	5	.5469E+04	7	.4348E+04
120	.22244	-.04423	20	8	.2054E+00	19	9	.1140E+04	.3482E+00	5	.2771E+04	7	.2196E+04

***** ITERATION LIMIT REACHED *****

INTERMEDIATE OUTPUT FOR MEDIUM MESH

WE = 1.9000 EPS = .2000 MAXIT FOR THIS MESH = 250

ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERRU	ICPL	CPERRL
10	.17667	-.03404	37	16	.3445E-01	35	16	.1868E+04	.3594E-01	9	.1058E+01	9	.1212E+01
20	.17601	-.03392	10	17	.3936E-02	9	16	.1796E+03	.1148E-03	33	.1047E+01	33	.3606E+01
30	.17701	-.03391	10	17	.3056E-02	9	16	.1384E+03	.3180E-03	29	.0453E+02	11	.8569E+02
40	.17760	-.03374	10	17	.2310E-02	9	16	.1048E+03	.1822E-03	23	.5003E+02	11	.7405E+02
50	.17806	-.03357	10	17	.1736E-02	9	16	.7884E+02	.1528E-03	23	.4124E+02	11	.5999E+02
60	.17844	-.03344	10	17	.1294E-02	9	16	.5884E+02	.1228E-03	22	.2900E+02	11	.4731E+02
70	.17871	-.03334	10	17	.9533E-03	9	16	.4349E+02	.8333E-04	22	.2135E+02	11	.3721E+02
80	.17890	-.03327	10	17	.7030E-03	9	16	.3204E+02	.5892E-04	21	.1541E+02	11	.2772E+02
90	.17904	-.03322	10	17	.5155E-03	9	16	.2352E+02	.4277E-04	22	.1117E+02	11	.2075E+02
100	.17914	-.03318	10	17	.3770E-03	9	16	.1721E+02	.3095E-04	22	.8113E+01	11	.1538E+02
110	.17921	-.03316	10	17	.2752E-03	9	16	.1257E+02	.2247E-04	21	.5990E+01	11	.1132E+02
120	.17926	-.03314	10	17	.2006E-03	9	16	.9168E+01	.1632E-04	22	.4265E+01	11	.8324E+01
130	.17930	-.03312	10	17	.1462E-03	9	16	.6680E+01	.1184E-04	22	.3115E+01	11	.6087E+01
140	.17933	-.03311	10	17	.1064E-03	9	16	.4864E+01	.8588E-05	22	.2264E+01	11	.4440E+01
150	.17935	-.03311	10	17	.7742E-04	9	16	.3540E+01	.6234E-05	22	.1646E+01	11	.3243E+01
160	.17936	-.03310	10	17	.5632E-04	9	16	.2575E+01	.4529E-05	22	.1195E+01	11	.2363E+01
170	.17937	-.03310	10	17	.4096E-04	9	16	.1873E+01	.3294E-05	22	.8594E+00	11	.1721E+01
180	.17938	-.03309	10	17	.2978E-04	9	16	.1362E+01	.2396E-05	22	.6319E+00	11	.1253E+01
190	.17939	-.03309	10	17	.2165E-04	9	16	.9971E+00	.1743E-05	22	.4592E+00	11	.9119E+00
200	.17939	-.03309	10	17	.1573E-04	9	16	.7196E+00	.1268E-05	22	.3337E+00	11	.6634E+00
210	.17940	-.03309	10	17	.1144E-04	9	16	.5230E+00	.9215E-06	22	.2425E+00	11	.4859E+00
220	.17940	-.03309	10	17	.8310E-05	9	16	.3801E+00	.6697E-06	22	.1762E+00	11	.3536E+00

***** SOLUTION CONVERGED *****

Figure 4-8.- Continued

INTERMEDIATE OUTPUT FOR FINE MESH

WE = 1.9500 EPS = .2000 MAXIT FOR THIS MESH = 500

ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERRU	ICPL	CPERRL
10	.17816	-.03330	8	33	.3351E-02	16	32	.3816E+03	.2225E-04	17	.1178E+04	17	.1371E+04
20	.17814	-.03325	15	18	.1189E-02	17	32	.2861E+03	.9872E-06	22	.6199E-02	21	.7104E-02
30	.17821	-.03319	17	36	.9006E-03	17	32	.2281E+03	.4088E-04	24	.2580E-02	21	.4984E-02
40	.17843	-.03320	80	32	.7876E-03	17	32	.1934E+03	.8575E-04	65	.1624E-02	20	.2721E-02
50	.17867	-.03322	18	35	.6715E-03	17	32	.1713E+03	.8253E-04	69	.1430E-02	22	.1591E-02
60	.17886	-.03322	17	35	.5803E-03	17	32	.1480E+03	.5796E-04	38	.1152E-02	22	.1978E-02
70	.17901	-.03318	17	36	.4923E-03	17	32	.1254E+03	.4856E-04	27	.1125E-02	22	.1926E-02
80	.17913	-.03318	17	36	.4131E-03	17	32	.1053E+03	.4173E-04	27	.1078E-02	21	.1794E-02
90	.17924	-.03317	17	36	.3473E-03	17	32	.8846E+02	.3559E-04	27	.7117E-03	21	.1448E-02
100	.17933	-.03316	17	36	.2928E-03	17	32	.7466E+02	.3017E-04	27	.5928E-03	21	.1178E-02
110	.17941	-.03315	17	36	.2470E-03	17	32	.6301E+02	.2511E-04	27	.5037E-03	21	.9950E-03
120	.17947	-.03314	17	36	.2082E-03	17	32	.5314E+02	.2106E-04	27	.4273E-03	21	.8425E-03
130	.17953	-.03313	17	36	.1755E-03	17	32	.4480E+02	.1762E-04	27	.3602E-03	21	.7144E-03
140	.17957	-.03313	17	36	.1479E-03	17	32	.3776E+02	.1487E-04	27	.3050E-03	21	.6031E-03
150	.17961	-.03312	17	36	.1246E-03	17	32	.3182E+02	.1249E-04	26	.2574E-03	21	.5096E-03
160	.17964	-.03312	17	36	.1049E-03	17	32	.2681E+02	.1048E-04	26	.2168E-03	21	.4301E-03
170	.17967	-.03311	17	36	.8838E-04	17	32	.2258E+02	.8925E-05	26	.1827E-03	21	.3626E-03
180	.17969	-.03311	17	36	.7446E-04	17	32	.1902E+02	.7546E-05	26	.1536E-03	21	.3117E-03
190	.17971	-.03311	17	36	.6273E-04	17	32	.1603E+02	.6363E-05	26	.1293E-03	21	.2562E-03
200	.17973	-.03310	17	36	.5294E-04	17	32	.1350E+02	.5359E-05	26	.1091E-03	21	.2181E-03
210	.17974	-.03310	17	36	.4451E-04	17	32	.1138E+02	.4477E-05	26	.9214E-04	21	.1833E-03
220	.17975	-.03310	17	36	.3750E-04	17	32	.9588E+01	.3756E-05	26	.7788E-04	21	.1541E-03
230	.17976	-.03310	17	36	.3159E-04	17	32	.8077E+01	.3154E-05	26	.6574E-04	21	.1298E-03
240	.17977	-.03310	17	36	.2651E-04	17	32	.6804E+01	.2657E-05	26	.5541E-04	21	.1093E-03
250	.17978	-.03310	17	36	.2242E-04	17	32	.5733E+01	.2239E-05	26	.4646E-04	21	.9211E-04
260	.17978	-.03310	17	36	.1889E-04	17	32	.4829E+01	.1887E-05	26	.3933E-04	21	.7764E-04
270	.17979	-.03310	17	36	.1591E-04	17	32	.4069E+01	.1589E-05	26	.3313E-04	21	.6543E-04
280	.17979	-.03309	17	36	.1340E-04	17	32	.3477E+01	.1338E-05	26	.2791E-04	21	.5514E-04
290	.17979	-.03309	17	36	.1129E-04	17	32	.2887E+01	.1127E-05	26	.2322E-04	21	.4646E-04
300	.17980	-.03309	17	36	.9511E-05	17	32	.2432E+01	.9468E-06	26	.1981E-04	21	.3914E-04

***** SOLUTION CONVERGED *****

* FINAL OUTPUT *

PRINTOUT IN PHYSICAL VARIABLES.
DEFINITION OF SIMILARITY PARAMETERS BY KRUPP
BOUNDARY CONDITION FOR FREE JET
DIFFERENCE EQUATIONS ARE FULLY CONSERVATIVE.
KUTTA CONDITION IS ENFORCED.

MACH = 1.1000000
DELTA = .0600000
ALPHA = 2.0000000
K = -1.2456397

PARAMETERS USED TO TRANSFORM VARIABLES
TO TRANSONIC SCALING

CPFACT = .1426887
CDFACT = .0085613
CMFACT = .1426887
CLFACT = .1426887
YFACT = 2.4354913
VFAC = 3.9377467

Figure 4-8.- Continued

FINAL OUTPUT FOR FINE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.

CL = .179798
CM = -.033093
CP* = -1.48116

B = BEFORE OR BEHIND AIRFOIL
U = UPPER AIRFOIL SURFACE
L = LOWER AIRFOIL SURFACE
* = CRITICAL PRESSURE

I	X	LOWER Y=0-		UPPER Y=0+			
		CP	M1	CP	M1		
1	-4.826958	-0.000000	1.100000	-0.000000	1.100000		* B
2	-4.472160	-0.000000	1.100000	-0.000000	1.100000		* B
3	-1.170661	-0.000000	1.100000	-0.000000	1.100000		* B
4	-0.972966	-0.000000	1.100000	-0.000000	1.100000		* B
5	-0.809837	.000000	1.100000	.000000	1.100000		* B
6	-0.653469	.004128	1.097336	.004128	1.097336		* B
7	-0.479854	.160203	.991394	.160203	.991394		*
8	-0.295092	.356900	.844091	.357900	.844691		*
9	-0.178882	.428632	.776067	.428632	.776067		*
10	-0.114702	.505191	.702663	.505191	.702663		*
11	-0.075551	.581942	.620416	.581942	.620416		*
12	-0.050198	.664116	.518082	.664116	.518082		*
13	-0.032939	.757480	.388831	.757480	.388831		*
14	-0.020518	.873372	0.000000	.873372	0.000000		*
15	-0.011063	1.039884	0.000000	1.039884	0.000000		*
16	-0.003465	1.326950	0.000000	1.326950	0.000000	B	*
AIRFOIL LEADING EDGE							
17	.003024	1.330277	0.000000	.926890	0.000000	L	U
18	.008803	.699742	.466796	.368148	.879264		*
19	.014240	.522078	.685413	.163714	.988881		*
20	.019568	.437645	.767791	.100125	1.033461		*
21	.025061	.381418	.818665	.060113	1.060552		*
22	.030961	.337251	.855477	.030970	1.080119		*
23	.037372	.296972	.886631	.004678	1.096981		*
24	.045413	.263379	.916486	-.119466	1.112487		*
25	.054966	.228482	.941305	-.043288	1.127552		*
26	.067013	.193095	.967588	-.067548	1.142703		*
27	.082363	.156126	.994305	-.092332	1.157976		*
28	.101517	.116170	1.022395	-.117106	1.173045		*
29	.124201	.078579	1.048136	-.140930	1.187935		*
30	.149372	.047527	1.068932	-.162869	1.203083		*
31	.175806	.020784	1.086523	-.182423	1.211875		*
32	.202583	-.002261	1.101456	-.199575	1.221864		*
33	.229179	-.021176	1.114205	-.214566	1.230534		*
34	.255339	-.039546	1.125193	-.227748	1.237165		*
35	.281029	-.054875	1.134814	-.239429	1.242429		*
36	.306213	-.068600	1.143356	-.249931	1.245742		*
37	.330950	-.081226	1.151034	-.259440	1.248120		*
38	.355299	-.092383	1.158008	-.268151	1.261126		*
39	.379323	-.102844	1.164394	-.276263	1.265545		*
40	.403081	-.112545	1.170285	-.283701	1.269738		*
41	.426629	-.121598	1.175756	-.291724	1.273653		*
42	.450015	-.130095	1.180868	-.297333	1.277326		*
43	.473280	-.138113	1.185672	-.303577	1.280787		*
44	.496469	-.145719	1.190211	-.309496	1.284039		*
45	.519582	-.152972	1.194523	-.315129	1.287165		*
46	.542669	-.159926	1.198641	-.320515	1.290125		*
47	.565740	-.166619	1.202595	-.325677	1.292964		*
48	.588807	-.173105	1.206412	-.330667	1.295694		*
49	.611881	-.179424	1.210119	-.335517	1.298345		*
50	.634971	-.185614	1.213740	-.340265	1.300935		*
51	.658084	-.191712	1.217297	-.344948	1.303484		*
52	.681229	-.197753	1.220810	-.349600	1.306012		*
53	.704417	-.203771	1.224299	-.354256	1.308537		*
54	.727663	-.209796	1.227783	-.358949	1.311077		*
55	.750988	-.215858	1.231278	-.363710	1.313649		*
56	.774418	-.221990	1.234803	-.368571	1.316269		*
57	.797988	-.228221	1.238376	-.373522	1.318954		*
58	.821742	-.234586	1.242014	-.378715	1.321721		*
59	.845736	-.241120	1.245738	-.384067	1.324589		*
60	.870039	-.247866	1.249571	-.389656	1.327575		*
61	.894736	-.254872	1.253539	-.395228	1.330708		*
62	.919931	-.262194	1.257673	-.401736	1.334011		*
63	.945757	-.269905	1.262012	-.408346	1.337519		*
64	.972377	-.278091	1.266602	-.415441	1.341274		*
65	1.000000	-.284942	1.272217	-.403550	1.280772		*
AIRFOIL TRAILING EDGE							
66	1.028899	-.189936	1.216262	-.189936	1.216262		* B
67	1.059441	-.189366	1.215343	-.188360	1.215343		* B
68	1.092135	-.191908	1.217411	-.191908	1.217411		* B
69	1.127723	-.196275	1.219951	-.196275	1.219951		* B
70	1.167351	-.201266	1.222848	-.201266	1.222848		* B
71	1.212946	-.207003	1.226169	-.207003	1.226169		* B
72	1.268145	-.213838	1.230114	-.213838	1.230114		* B
73	1.341184	-.222620	1.235165	-.222620	1.235165		* B
74	1.456833	-.233196	1.241221	-.233196	1.241221		* a
75	1.671514	-.237715	1.243799	-.237715	1.243799		* B
76	1.876313	-.227569	1.238002	-.227569	1.238002		* B
77	2.062561	-.200768	1.222599	-.200768	1.222599		* B
78	2.209035	-.147135	1.191054	-.147135	1.191054		* B
79	2.406774	-.016744	1.106982	-.016744	1.106982		* B
80	2.702997	.095516	1.036618	.095516	1.036618		* B
81	4.890373	.107149	1.028631	.107149	1.028631		* B

Figure 4-B.- Continued

***** CAUTION *****
 MAXIMUM MACH NUMBER EXCEEDS 1.3
 SHOCK JUMPS IN ERROR IF UPSTREAM NORMAL MACH NUMBER GREATER THAN 1.3

Y-GRID VALUES		Y(J) J= 1 TO 64						
-2.000000	-1.812909	-1.642930	-1.488178	-1.347027	-1.218072	-1.100091	-.992019	
-.892925	-.801993	-.718505	-.641829	-.571409	-.506738	-.447391	-.392976	
-.343146	-.297595	-.256050	-.218269	-.184038	-.153167	-.125487	-.1001205	
-.079132	-.060216	-.044007	-.030425	-.019401	-.010882	-.004827	-.001205	
.001205	.004827	.010882	.019401	.030425	.044007	.060216	.079132	
.100852	.125487	.153167	.184038	.218269	.256050	.297595	.343146	
.392976	.447391	.506738	.571409	.641829	.718505	.801993	.892925	
.992019	1.100091	1.218072	1.347027	1.488178	1.642930	1.812909	2.000000	

SONIC LINE OUTPUT

SONIC LINE COORDINATES

Y	X-SONIC	
1.48818	-.13536	-.08245
1.34703	-.21078	.00350
1.21807	-.25496	.07413
1.10009	-.29580	.12619
.99202	-.32908	.15905
.89293	-.35538	.17672
.80199	-.37752	.17473
.71851	-.39664	.17336
.64183	-.41325	.17391
.57140	-.42762	.17122
.50674	-.43996	.16407
.44739	-.45044	.15576
.39298	-.45923	.14623
.34315	-.46649	.13688
.29759	-.47241	.12701
.25605	-.47714	.11763
.21827	-.48100	.10839
.18404	-.48425	.09894
.15317	-.48675	.09039
.12549	-.48866	.08137
.10085	-.49009	.07344
.07913	-.49114	.06502
.06022	-.49189	.05720
.04401	-.49242	.04985
.03042	-.49278	.04218
.01940	-.49301	.03503
.01088	-.49316	.02832
.00483	-.49324	.02229
.00121	-.49329	.01756
BODY LOCATION		
-.00121	-.49331	.08738
-.00483	-.49335	.09073
-.01088	-.49339	.09566
-.01940	-.49343	.10152
-.03042	-.49343	.10968
-.04401	-.49336	.11789
-.06022	-.49318	.12649
-.07913	-.49283	.13668
-.10085	-.49225	.14999
-.12549	-.49135	.15642
-.15317	-.49005	.16675
-.18404	-.48823	.17659
-.21827	-.48576	.18767
-.25605	-.48247	.19783
-.29759	-.47834	.20829
-.34315	-.47335	.21825
-.39298	-.46705	.22763
-.44739	-.45923	.23562
-.50674	-.44969	.24283
-.57140	-.43823	.24803
-.64183	-.42465	.25085
-.71851	-.40872	.25089
-.80199	-.39017	.24703
-.89293	-.36854	.23650
-.99202	-.34294	.21289
-1.10009	-.31114	.17628
-1.21807	-.26934	.10499
-1.34703	-.22295	.02188
-1.48818	-.15083	-.07634

Figure 4-8.- Continued

SHOCK WAVE DRAG AND TOTAL PRESSURE PROFILE OUTPUT

INVISCID WAKE PROFILES FOR INDIVIDUAL SHOCK WAVES WITHIN MOMENTUM CONTOUR

SHOCK 1 WAVE DRAG FOR Y	THIS SHOCK= CD(Y)	.003882 PG/POINF
-1.48817799	.00002216	.99998411
-1.34702736	.00023112	.99983431
-1.21807223	.00027030	.99980623
-1.10009107	.00065835	.99952804
-.99201912	.00082368	.99940952
-.89292538	.00098401	.99929459
-.80199344	.00114331	.99918038
-.71850542	.00130623	.99906359
-.64182864	.00147691	.99894123
-.57140431	.00165876	.99881387
-.50673795	.00185440	.99867062
-.44739136	.00206558	.99851923
-.39297572	.00229307	.99835615
-.34314575	.00253636	.99818173
-.29759466	.00279343	.99799745
-.25604987	.00209871	.99849548
-.21826935	.00222358	.99849596
-.18403842	.00239862	.99832923
-.15316703	.00241893	.99826592
-.12548743	.00248856	.99821600
-.10085215	.00254057	.99817872
-.07913226	.00257690	.99815268
-.06021592	.00260013	.99813602
-.04400713	.00261318	.99812667
-.03042464	.00261893	.99812254
-.01940111	.00262001	.99812176
-.01088238	.00261862	.99812276
-.00482689	.00261645	.99812432
-.00120527	.00261469	.99812558
.00120527	.00261332	.99812657
.00482689	.00261097	.99812825
.01088238	.00260628	.99813161
.01940111	.00259806	.99813751
.03042464	.00258470	.99814708
.04400713	.00256419	.99816178
.06021592	.00253427	.99818323
.07913226	.00249261	.99821310
.10085215	.00243711	.99825289
.12548743	.00236613	.99830377
.15316703	.00227872	.99836639
.18403842	.00217499	.99844080
.21826935	.00205554	.99852643
.25604987	.00272972	.99804312
.29759466	.00248230	.99822049
.34314575	.00224928	.99838753
.39297572	.00263275	.99854276
.44739136	.00183322	.99868580
.50673795	.00165001	.99881715
.57140431	.00148143	.99893800
.64182864	.00132538	.99905008
.71850542	.00117782	.99915564
.80199344	.00103386	.99925741
.89292538	.00089479	.99935454
.99201912	.00074998	.99946236
1.10009107	.00059780	.99957145
1.21807223	.00026902	.99980715
1.34702736	.00023089	.99983448
1.48817799	.00002755	.99998025

SHOCK WAVE EXTENDS OUTSIDE CONTOUR
PRINTOUT OF SHOCK LOSSES ARE NOT AVAILABLE FOR REST OF SHOCK

DRAG COEFFICIENT OUTPUT

CALCULATION OF DRAG COEFFICIENT BY MOMENTUM INTEGRAL METHOD

BOUNDARIES OF CONTOUR USED	CONTRIBUTION TO CD
UPSTREAM X = -1.170661	CDUP = 0.000000
DOWNSTREAM X = .727663	CDOWN = .117504
TOP Y = 1.488178	CDTP = .002420
BOTTOM Y = -1.488178	CDBT = .003498
BODY AFT OF X = .727663	CDBODY = .017822
TOTAL CONTRIBUTIONS AROUND CONTOUR =	.K34245

THERE ARE 1 SHOCKS INSIDE CONTOUR. TOTAL CDWAVE = .003882

NOTE - ONE OR MORE SHOCKS EXTEND OUTSIDE OF CONTOUR
CDWAVE DOES NOT EQUAL TOTAL WAVE DRAG

DRAG CALCULATED FROM MOMENTUM INTEGRAL CD = .038127

TIME TO RUN CASE WAS 20.11 SECONDS.

(b) Output.
Figure 4-8.- Concluded

SAMPLE CASE 6 - SUPERSONIC-NACA0006 AIRFOIL-POROUS TUNNEL WALL SIMULATION
 \$INP
 BCTYPE=5, POR=0.25, MAXIT=1000, AMESH=.TRUE., ALPHA=2.0, EMACH=1.1,
 H=2.0, DELTA=0.06, BCFOIL=1,
 \$END

(*) Input.

SAMPLE CASE 6 - SUPERSONIC - NACA0006 AIRFOIL - POROUS TUNNEL WALL SIMULATION

INPUT PARAMETERS

EMACH = 1.10000 POR = .25000 IMIN = 1 BCTYPE = 5 AMESH = T
 DELTA = .06000 CLSET = 0.00000 IMAXI = 31 BCFOIL = 1 PHY3 = T
 ALPHA = 2.00000 EPS = .20000 JMIN = 1 PSTART = 1 PSAVE = F
 AK = 0.00000 RIGF = 0.00000 JMAXI = 64 PRTFLO = 1 KUTTA = T
 GAM = 1.40000 WCIRC = 1.00000 MAXIT = 1000 IPRTER = 10 FCR = T
 F = 0.00000 CVERGE = .00001 NU = 100 SIMDEF = 3
 H = 2.00000 DVERGE = 10.0 NL = 75 ICUT = 2
 WE = 1.8001901195

XIN
 -4.826958 -1.472160 -1.170661 -.972966 -.809857 -.653469 -.479854 -.295192
 -1.178882 -.114702 -.075551 -.050198 -.032930 -.020518 -.011103 -.003455
 .003024 .008803 .014240 .019568 .025761 .030961 .037614 .045413
 .054966 .067013 .082363 .101517 .124201 .149372 .179806 .22583
 .229179 .255393 .281529 .306213 .330950 .355299 .379323 .403061
 .426629 .450015 .473280 .496400 .519582 .542669 .565747 .588807
 .611881 .634971 .658084 .681229 .704417 .727663 .750982 .774418
 .797988 .821742 .845736 .870039 .894736 .919931 .945757 .972377
 1.000000 1.028399 1.059441 1.092135 1.127723 1.167351 1.212946 1.265145
 1.341184 1.456833 1.671514 1.876313 2.042561 2.269055 2.436774 2.722977
 4.890373

YIN
 -1.000000 -.906455 -.821465 -.744089 -.673514 -.609036 -.551146 -.496144
 -.446463 -.400997 -.359253 -.322914 -.285702 -.253369 -.225696 -.196498
 -.171573 -.148797 -.128025 -.109135 -.092019 -.076584 -.062744 -.050426
 -.039566 -.030108 -.022384 -.015212 -.009701 -.005441 -.002413 -.000503
 .000603 .002413 .005441 .009701 .015212 .022384 .030108 .039566
 .050426 .062744 .076584 .092019 .109135 .128025 .148797 .171573
 .196488 .223696 .253369 .285702 .320914 .359253 .400997 .446463
 .496010 .550046 .609036 .673514 .744089 .821465 .906455 1.000000

INTERMEDIATE OUTPUT FOR COARSE MESH

WE = 1.8000 EPS = .2000 MAXIT FOR THIS MESH = 250

ITERATION	CL	CH	IERR	JERR	ERROR	IRL	JRL	9IGRL	ERICRC	ICPU	CPEHRU	ICPL	CPERRL
10	.24718	-.04579	21	16	.9428E-01	5	8	.1692E+03	.2621E-01	15	.1054E+01	17	.4494E+11
20	.22445	-.04678	21	8	.1393E+00	19	9	.2409E+03	.7539E-01	7	.5343E-01	6	.5860E-01
30	.21628	-.04569	5	9	.5930E-02	5	8	.4444E+02	.4603E-02	17	.2749E-01	17	.3673E-01
40	.21568	-.04584	5	9	.3268E-02	5	8	.2445E+02	.1434E-03	5	.7394E-02	7	.9736E-02
50	.21551	-.04588	5	9	.1838E-02	5	8	.1351E+02	.5288E-04	6	.4341E-02	6	.5392E-02
60	.21546	-.04591	5	8	.1007E-02	5	8	.7522E+01	.4593E-05	5	.2511E-02	5	.2888E-02
70	.21546	-.04594	5	8	.5620E-03	5	8	.4195E+01	.3777E-05	6	.1430E-02	6	.1593E-02
80	.21547	-.04595	5	8	.3133E-03	5	8	.2338E+01	.2887E-05	5	.3179E-03	6	.3820E-03
90	.21548	-.04596	5	8	.1747E-03	5	8	.1304E+01	.1664E-05	5	.4570E-03	6	.4893E-03
100	.21548	-.04597	5	8	.9745E-04	5	8	.7271E+00	.9199E-06	5	.2370E-03	6	.2726E-03
110	.21548	-.04597	5	8	.5436E-04	5	8	.4056E+00	.5072E-06	5	.1439E-03	6	.1529E-03
120	.21548	-.04597	5	8	.3032E-04	5	8	.2262E+00	.2866E-06	5	.8323E-04	6	.8473E-04
130	.21549	-.04597	5	8	.1692E-04	5	8	.1262E+00	.1557E-06	5	.4404E-04	6	.4726E-04
140	.21549	-.04597	5	8	.9436E-05	5	8	.7040E-01	.8562E-07	5	.2449E-04	6	.2636E-04

***** SOLUTION CONVERGED *****

Figure 4-9.- Sample test case 6 - supersonic, perforated/porous wall simulation (abbreviated output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2^\circ$, $M_\infty = 1.1$, $H = 2$.

INTERMEDIATE OUTPUT FOR MEDIUM MESH

WE = 1.9000		EPS = .2000		MAXIT FOR THIS MESH = 500									
ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERKU	ICPL	CPERRL
10	.21231	-.04389	41	16	.7013E-02	10	4	.5456E+02	.5279E-03	9	.1039E+1	9	.1189E+11
20	.21428	-.04363	9	1	.2578E-02	9	16	.5817E+02	.5714E-03	9	.3194E+02	10	.1830E-11
30	.21537	-.04365	8	4	.1432E-02	9	16	.6361E+02	.2881E-03	10	.2165E+02	10	.4220E-12
40	.21586	-.04368	9	14	.1207E-02	9	16	.5724E+02	.1219E-03	11	.1780E+02	9	.0817E-03
50	.21605	-.04372	9	17	.1051E-02	9	16	.4990E+02	.4649E-04	12	.1275E+02	11	.1060E-02
60	.21613	-.04374	9	17	.9196E-03	9	16	.4261E+02	.1941E-04	13	.8323E+02	11	.1144E-02
70	.21616	-.04376	10	17	.8019E-03	9	16	.3799E+02	.1036E-04	13	.6703E+03	11	.9013E-05
80	.21619	-.04378	13	17	.6966E-03	9	16	.3298E+02	.6725E-05	13	.5729E+03	11	.3777E-03
90	.21620	-.04379	13	17	.6034E-03	9	16	.2856E+02	.5051E-05	13	.5925E+03	11	.7823E-03
100	.21621	-.04381	10	17	.5220E-03	9	16	.2470E+02	.3547E-05	13	.4433E-03	11	.6867E-03
110	.21622	-.04382	10	17	.4577E-03	9	16	.2132E+02	.2503E-05	13	.3997E+03	11	.0019E+03
120	.21623	-.04383	10	17	.3895E-03	9	16	.1837E+02	.1915E-05	13	.3374E+03	11	.5272E-05
130	.21623	-.04384	10	17	.3345E-03	9	16	.1582E+02	.1538E-05	13	.2914E+03	11	.4583E-03
140	.21624	-.04384	10	17	.2878E-03	9	16	.1369E+02	.1267E-05	13	.2516E+03	11	.3971E-03
150	.21624	-.04385	10	17	.2474E-03	9	16	.1169E+02	.1057E-05	13	.2154E+03	11	.3433E-03
160	.21624	-.04386	10	17	.2125E-03	9	16	.1004E+02	.8867E-05	13	.1876E+03	11	.2963E-03
170	.21624	-.04386	10	17	.1825E-03	9	16	.8621E+01	.7775E-06	13	.1604E+03	11	.2552E-03
180	.21625	-.04386	10	17	.1587E-03	9	16	.7401E+01	.6811E-06	13	.1377E+03	11	.2190E-03
190	.21625	-.04387	10	17	.1344E-03	9	16	.6351E+01	.7808E-06	13	.1189E+03	11	.1889E-03
200	.21625	-.04387	10	17	.1193E-03	9	16	.547E+01	.6944E-06	13	.1122E+03	11	.1623E-03
210	.21625	-.04387	10	17	.9888E-04	9	16	.4671E+01	.6228E-06	13	.832E+02	11	.1393E-03
220	.21625	-.04388	13	17	.8477E-04	9	16	.4004E+01	.5186E-06	13	.7359E+02	11	.1193E-03
230	.21626	-.04388	13	17	.7266E-04	9	16	.3432E+01	.4444E-06	13	.6422E+02	11	.1127E-03
240	.21626	-.04388	16	17	.6227E-04	9	16	.2941E+01	.3806E-06	13	.5592E+02	11	.8816E-04
250	.21626	-.04388	10	17	.5336E-04	9	16	.2527E+01	.3256E-06	13	.4700E+02	11	.7523E-04
260	.21626	-.04388	13	17	.4571E-04	9	16	.2159E+01	.2786E-06	13	.4185E+02	11	.6476E-04
270	.21626	-.04388	10	17	.3916E-04	9	16	.1849E+01	.2383E-06	13	.371E+02	11	.5541E-04
280	.21626	-.04389	10	17	.3355E-04	9	16	.1594E+01	.2039E-06	13	.3307E+02	11	.4701E-04
290	.21626	-.04389	10	17	.2873E-04	9	16	.1357E+01	.1744E-06	13	.3002E+02	11	.4079E-04
300	.21626	-.04389	10	17	.2451E-04	9	16	.1162E+01	.1443E-06	13	.272E+02	11	.3493E-04
310	.21626	-.04389	13	17	.2108E-04	9	16	.9953E+00	.1277E-06	13	.198E+02	11	.2992E-04
320	.21626	-.04389	17	17	.1826E-04	9	16	.8522E+00	.1093E-06	13	.1603E+02	11	.2589E-04
330	.21626	-.04389	10	17	.1546E-04	9	16	.7299E+00	.9354E-07	13	.1384E+02	11	.2247E-04
340	.21626	-.04390	10	17	.1324E-04	9	16	.6251E+00	.8006E-07	13	.1163E+02	11	.1862E-04
350	.21626	-.04389	10	17	.1133E-04	9	16	.5352E+00	.6852E-07	13	.1113E+02	11	.1612E-04
360	.21626	-.04389	10	17	.9705E-05	9	16	.4583E+00	.5865E-07	13	.8930E+01	11	.1381E-04

***** SOLUTION CONVERGED *****

INTERMEDIATE OUTPUT FOR FINE MESH

WE = 1.9500		EPS = .2000		MAXIT FOR THIS MESH = 1000									
ITERATION	CL	CM	IERR	JERR	ERROR	IRL	JRL	BIGRL	ERCIRC	ICPU	CPERKU	ICPL	CPERRL
10	.21464	-.04466	9	33	.3748E-02	17	42	.5463E+03	.1834E-03	17	.1133E+01	17	.1394E+11
20	.21513	-.04427	17	17	.2078E-02	17	42	.4921E+03	.1254E-03	22	.4204E+02	20	.3120E-12
30	.21533	-.04431	19	12	.1893E-02	17	32	.4637E+03	.4542E-04	23	.3840E+02	20	.2133E-12
40	.21544	-.04429	18	37	.1715E-02	17	32	.4461E+03	.3481E-04	43	.6135E+03	21	.9770E-03
50	.21551	-.04426	18	35	.1651E-02	17	32	.4298E+03	.2224E-04	31	.7471E+03	23	.1133E-02
60	.21556	-.04423	17	35	.1574E-02	17	32	.4090E+03	.1361E-04	27	.4800E+03	22	.1008E-02
70	.21560	-.04421	17	36	.1497E-02	17	32	.3890E+03	.1546E-04	24	.3922E+03	21	.1233E-02
80	.21564	-.04419	18	37	.1440E-02	17	32	.3752E+03	.1282E-04	27	.4499E+03	21	.1043E-02
90	.21567	-.04417	18	35	.1392E-02	17	32	.3632E+03	.8310E-05	26	.4983E+03	22	.9144E-03
100	.21569	-.04415	18	36	.1347E-02	17	32	.3516E+03	.7620E-05	20	.3900E+03	21	.8334E-03
110	.21571	-.04414	18	35	.1307E-02	17	32	.3413E+03	.6661E-05	20	.3721E+03	22	.7300E-03
120	.21573	-.04413	18	35	.1270E-02	17	32	.3316E+03	.4733E-05	25	.3525E+03	22	.5975E-03
130	.21574	-.04411	18	35	.1234E-02	17	32	.3224E+03	.2818E-05	25	.3258E+03	22	.0459E-03

Figure 4-9.- Continued

140	.21375	-.04410	18	35	.1200E-02	17	32	.3137E+03	.5540E-05	26	.3483E-03	22	.6665E-03
150	.21377	-.04410	18	35	.1168E-02	17	32	.3054E+03	.7294E-05	26	.2964E-03	22	.5811E-03
160	.21578	-.04409	18	35	.1136E-02	17	32	.2969E+03	.4139E-05	26	.2946E-03	22	.5758E-03
170	.21580	-.04408	18	35	.1103E-02	17	32	.2885E+03	.5502E-05	26	.3401E-03	22	.5805E-03
180	.21582	-.04407	18	35	.1070E-02	17	32	.2798E+03	.4956E-05	26	.3264E-03	22	.5972E-03
190	.21583	-.04406	18	34	.1035E-02	17	32	.2708E+03	.3619E-05	26	.3693E-03	22	.6317E-03
200	.21584	-.04404	18	34	.9933E-03	17	32	.2612E+03	.4095E-05	26	.3979E-03	22	.6801E-03
210	.21585	-.04403	18	34	.9603E-03	17	32	.2513E+03	.3742E-05	26	.4057E-03	22	.7670E-03
220	.21586	-.04402	18	35	.9219E-03	17	32	.2413E+03	.4323E-05	26	.4021E-03	22	.7126E-03
230	.21587	-.04401	18	35	.8842E-03	17	32	.2314E+03	.2866E-05	25	.3866E-03	22	.6993E-03
240	.21588	-.04400	18	35	.8474E-03	17	32	.2218E+03	.1877E-05	25	.3687E-03	22	.6731E-03
250	.21589	-.04398	18	35	.8116E-03	17	32	.2125E+03	.3031E-05	26	.3536E-03	22	.6479E-03
260	.21589	-.04397	18	35	.7759E-03	17	32	.2035E+03	.1034E-05	26	.3373E-03	22	.6259E-03
270	.21589	-.04396	18	35	.7434E-03	17	32	.1947E+03	.2244E-05	26	.3228E-03	22	.5982E-03
280	.21589	-.04395	18	35	.7110E-03	17	32	.1863E+03	.7190E-06	26	.3082E-03	22	.5744E-03
290	.21589	-.04394	18	35	.6794E-03	17	32	.1780E+03	.8905E-06	26	.2945E-03	22	.5532E-03
300	.21588	-.04392	18	35	.6489E-03	17	32	.1701E+03	.2203E-05	25	.2841E-03	22	.5272E-03
310	.21588	-.04391	18	35	.6197E-03	17	32	.1625E+03	.9953E-06	25	.2703E-03	22	.5041E-03
320	.21588	-.04390	18	35	.5916E-03	17	32	.1551E+03	.2114E-05	25	.2587E-03	22	.4835E-03
330	.21587	-.04389	18	35	.5646E-03	17	32	.1481E+03	.2136E-05	25	.2476E-03	22	.4591E-03
340	.21586	-.04388	18	35	.5387E-03	17	32	.1413E+03	.1718E-05	25	.2375E-03	22	.4352E-03
350	.21586	-.04387	18	35	.5137E-03	17	32	.1347E+03	.2892E-05	25	.2271E-03	22	.4244E-03
360	.21585	-.04386	18	35	.4898E-03	17	32	.1285E+03	.2826E-05	25	.2149E-03	21	.4057E-03
370	.21584	-.04385	18	35	.4665E-03	17	32	.1224E+03	.2596E-05	25	.2174E-03	22	.3910E-03
380	.21583	-.04385	18	34	.4441E-03	17	32	.1165E+03	.3321E-05	25	.2141E-03	21	.3728E-03
390	.21582	-.04384	18	34	.4229E-03	17	32	.1110E+03	.3006E-05	24	.1955E-03	21	.3499E-03
400	.21581	-.04383	18	34	.4026E-03	17	32	.1057E+03	.2687E-05	24	.1844E-03	22	.3329E-03
410	.21581	-.04382	18	34	.3833E-03	17	32	.1006E+03	.3066E-05	24	.1766E-03	21	.3187E-03
420	.21580	-.04382	18	34	.3651E-03	17	32	.9585E+02	.2643E-05	24	.1690E-03	21	.3011E-03
430	.21579	-.04381	18	34	.3477E-03	17	32	.9120E+02	.2395E-05	24	.1638E-03	22	.2873E-03
440	.21578	-.04380	18	34	.3311E-03	17	32	.8694E+02	.2234E-05	24	.1592E-03	21	.2759E-03
450	.21578	-.04380	18	34	.3153E-03	17	32	.8290E+02	.2154E-05	24	.1434E-03	21	.2604E-03
460	.21577	-.04379	18	35	.3000E-03	17	32	.7899E+02	.2199E-05	24	.1378E-03	22	.2538E-03
470	.21577	-.04379	18	35	.2853E-03	17	32	.7491E+02	.2124E-05	25	.1349E-03	22	.2491E-03
480	.21576	-.04378	18	35	.2712E-03	17	32	.7121E+02	.1989E-05	25	.1283E-03	21	.2363E-03
490	.21575	-.04378	18	34	.2578E-03	17	32	.6770E+02	.1934E-05	24	.1230E-03	21	.2228E-03
500	.21575	-.04377	18	34	.2452E-03	17	32	.6441E+02	.1663E-05	24	.1160E-03	21	.2113E-03
510	.21574	-.04377	18	34	.2333E-03	17	32	.6133E+02	.1634E-05	25	.1197E-03	21	.1994E-03
520	.21574	-.04376	18	34	.2220E-03	17	32	.5832E+02	.1608E-05	25	.1162E-03	21	.1897E-03
530	.21574	-.04376	18	35	.2113E-03	17	32	.5552E+02	.1307E-05	25	.1100E-03	22	.1829E-03
540	.21573	-.04375	18	34	.2013E-03	17	32	.5288E+02	.1291E-05	25	.9273E-04	21	.1726E-03
550	.21573	-.04375	18	34	.1917E-03	17	32	.5037E+02	.1165E-05	25	.9223E-04	21	.1634E-03
560	.21573	-.04375	18	34	.1826E-03	17	32	.4797E+02	.1022E-05	25	.8715E-04	22	.1568E-03
570	.21572	-.04374	18	34	.1740E-03	17	32	.4571E+02	.1003E-05	25	.8489E-04	21	.1494E-03
580	.21572	-.04374	18	34	.1659E-03	17	32	.4359E+02	.8897E-06	25	.8117E-04	22	.1411E-03
590	.21572	-.04374	18	34	.1581E-03	17	32	.4155E+02	.7891E-06	25	.7669E-04	22	.1360E-03
600	.21571	-.04373	18	34	.1508E-03	17	32	.3964E+02	.7367E-06	25	.7309E-04	21	.1293E-03
610	.21571	-.04373	18	34	.1439E-03	17	32	.3782E+02	.6647E-06	25	.7041E-04	22	.1227E-03
620	.21571	-.04373	18	34	.1373E-03	17	32	.3608E+02	.6142E-06	25	.6797E-04	22	.1184E-03
630	.21571	-.04373	18	34	.1311E-03	17	32	.3445E+02	.4745E-06	25	.6521E-04	22	.1121E-03
640	.21571	-.04373	18	34	.1250E-03	17	32	.3285E+02	.5335E-06	25	.6463E-04	22	.1082E-03
650	.21571	-.04372	17	35	.1192E-03	17	32	.3134E+02	.5012E-06	25	.6490E-04	22	.1050E-03
660	.21571	-.04372	17	35	.1137E-03	17	32	.2989E+02	.4702E-06	25	.5901E-04	22	.1012E-03
670	.21570	-.04372	17	35	.1085E-03	17	32	.2851E+02	.4174E-06	25	.5621E-04	22	.9681E-04
680	.21570	-.04372	17	35	.1036E-03	17	32	.2724E+02	.3787E-06	26	.5390E-04	22	.9492E-04
690	.21570	-.04372	17	35	.9902E-04	17	32	.2603E+02	.3358E-06	26	.5210E-04	22	.8697E-04
700	.21570	-.04371	17	35	.9466E-04	17	32	.2489E+02	.3035E-06	26	.5126E-04	22	.8269E-04
710	.21570	-.04371	17	35	.9059E-04	17	32	.2382E+02	.2698E-06	26	.4955E-04	22	.7897E-04
720	.21570	-.04371	17	35	.8680E-04	17	32	.2282E+02	.2374E-06	26	.4694E-04	22	.7513E-04
730	.21570	-.04371	17	35	.8334E-04	17	32	.2192E+02	.2066E-06	26	.4449E-04	22	.7095E-04
740	.21570	-.04371	17	35	.8017E-04	17	32	.2108E+02	.1622E-06	26	.4443E-04	22	.5649E-04
750	.21570	-.04371	17	35	.7744E-04	17	32	.2026E+02	.1445E-06	26	.4403E-04	22	.5575E-04
760	.21570	-.04371	17	35	.7498E-04	17	32	.1946E+02	.1290E-06	26	.4335E-04	22	.6453E-04
770	.21570	-.04370	17	35	.7099E-04	17	32	.1867E+02	.1124E-06	26	.4168E-04	22	.6284E-04
780	.21570	-.04370	17	35	.6810E-04	17	32	.1791E+02	.9616E-07	26	.4070E-04	22	.6075E-04
790	.21570	-.04370	17	35	.6532E-04	17	32	.1718E+02	.8166E-07	26	.3813E-04	22	.5851E-04
800	.21570	-.04370	17	35	.6264E-04	17	32	.1648E+02	.6987E-07	26	.3683E-04	22	.5625E-04
810	.21570	-.04370	17	35	.6007E-04	17	32	.1580E+02	.6071E-07	26	.3393E-04	22	.5499E-04
820	.21570	-.04370	17	35	.5758E-04	17	32	.1515E+02	.5348E-07	26	.3393E-04	22	.5429E-04
830	.21570	-.04370	17	35	.5520E-04	17	32	.1452E+02	.4751E-07	26	.3257E-04	22	.5022E-04
840	.21570	-.04370	17	35	.5290E-04	17	32	.1391E+02	.4243E-07	26	.3126E-04	22	.4807E-04
850	.21570	-.04370	17	35	.5069E-04	17	32	.1333E+02	.3811E-07	26	.3005E-04	22	.4619E-04
860	.21570	-.04370	17	35	.4857E-04	17	32	.1278E+02	.4450E-07	26	.2875E-04	22	.4436E-04
870	.21570	-.04370	17	35	.4653E-04	17	32	.1224E+02	.4952E-07	26	.2754E-04	22	.4261E-04
880	.21570	-.04370	17	35	.4457E-04	17	32	.1172E+02	.3643E-07	26	.2639E-04	22	.4091E-04
890	.21570	-.04369	17	35	.4259E-04	17	32	.1123E+02	.3294E-07	26	.2531E-04	22	.3927E-04
900	.21570	-.04369	17	35	.4088E-04	17	32	.1075E+02	.2985E-07	26	.2425E-04	22	.3759E-04
910	.21570	-.04369	17	35	.3915E-04	17	32	.1030E+02	.2694E-07	26	.2326E-04	22	.3616E-04
920	.21570	-.04369	17	35	.3748E-04	17	32	.9859E+01	.2438E-07	26	.2229E-04	22	.3469E-04
930	.21569	-.04369	17	35	.3588E-04	17	32	.9439E+01	.2209E-07	26	.2136E-04	22	.3327E-04
940	.21569	-.04369	17	35	.3435E-04	17	32	.9036E+01	.2000E-07	26	.2147E-04	22	.3194E-04
950	.21569	-.04369	17	35	.3288E-04	17	32	.8649E+01	.1810E-07	26	.1961E-04	22	.3059E-04
960	.21569	-.04369	17	35	.3147E-04	17	32	.8278E+01	.1638E-07	26	.1878E-04	22	.2938E-04
970	.21569	-.04369	17	35	.3012E-04	17	32	.7922E+01	.1482E-07	26	.1799E-04	22	.2811E-04
980	.21569	-.04369	17	35	.2832E-04	17	32	.7581E+01	.1339E-07	26	.1723E-04	22	.2694E-04
990	.21569	-.04369	17	35	.2758E-04	17	32	.7254E+01	.1209E-07	26	.1549E-04	22	.2581E-04
1000	.21569	-.04369	17	35	.2638E-04	17	32	.6941E+01	.1091E-07	26	.1579E-04	22	.2473E-04

***** ITERATION LIMIT REACHED *****

```
*****  
* FINAL OUTPUT *  
*****
```

PRINTOUT IN PHYSICAL VARIABLES.
DEFINITION OF SIMILARITY PARAMETERS BY KRUPP
BOUNDARY CONDITION FOR POROUS WALL
DIFFERENCE EQUATIONS ARE FULLY CONSERVATIVE.
KUTTA CONDITION IS ENFORCED.

```
MACH = 1.1000000  
DELTA = .0600000  
ALPHA = 2.0000000  
K = -1.2456397
```

PARAMETERS USED TO TRANSFORM VARIABLES
TO TRANSONIC SCALING

```
CPFACT = .1426887  
CDFACT = .0085613  
CHFACT = .1426887  
CLFACT = .1426887  
YFACT = 2.4354913  
VFACT = 3.4377467
```

Figure 4-9.- Continued

FINAL OUTPUT FOR FINE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.

I	X	LOWER		UPPER			
		CP	M1	CP	M1		
		CL = .213695					
		CM = -.043689					
		CP* = .148116					
1	-4.826958	-0.000000	1.100000	-0.000000	1.100000		* B
2	-1.472160	-0.000000	1.100000	-0.000000	1.100000		* B
3	-1.170661	0.000000	1.100000	0.000000	1.100000		* B
4	-.972966	0.000000	1.100000	0.000000	1.100000		* B
5	-.809857	0.000000	1.100000	0.000000	1.100000		* B
6	-.653469	.137155	1.037740	.137155	1.037740		* B
7	-.479854	.314178	.874388	.314178	.874388		* B
8	-.295092	.580917	.613289	.580917	.613289		* B
9	-.178882	.854829	.451757	.854829	.451757		* B
10	-.114702	.525148	.682273	.525148	.682273		* B
11	-.075551	.597971	.601822	.597971	.601822		* B
12	-.050198	.677483	.499457	.677483	.499457		* B
13	-.032930	.768879	.346229	.768879	.346229		* B
14	-.020518	.883189	0.000000	.883189	0.000000		* B
15	-.011063	1.048274	0.000000	1.048274	0.000000		* B
16	-.003465	1.332789	0.000000	1.332789	0.000000		* B
		AIRFOIL LEADING EDGE					
17	.003024	1.350567	0.000000	.913049	0.000000	L	* U
18	.008803	.727384	.422738	.294086	.897029	L	* U
19	.014240	.549219	.656743	.152382	.996971	L	* U
20	.019568	.454576	.742528	.691298	1.039498	L	* U
21	.025061	.408406	.794328	.623712	1.066422	L	* U
22	.030961	.364450	.832633	.625855	1.083207	L	* U
23	.037624	.326578	.864334	.603170	1.098697	L	* U
24	.045413	.291394	.892669	.573795	1.113321	L	* U
25	.054966	.257163	.919452	.542874	1.127232	L	* U
26	.067013	.222728	.945629	.509286	1.141299	L	* U
27	.082363	.187851	.971423	.468197	1.155438	L	* U
28	.101517	.152341	.997000	.418994	1.169345	L	* U
29	.124201	.116842	1.021930	.362137	1.182437	L	* U
30	.149372	.085611	1.043369	.298211	1.194250	L	* U
31	.175806	.056152	1.060279	.227003	1.204568	L	* U
32	.202528	.029583	1.074188	.148197	1.213493	L	* U
33	.229179	.001680	1.085807	.062929	1.221127	L	* U
34	.255353	-.037660	1.095635	-.020953	1.227690	L	* U
35	.281029	-.090289	1.104645	-.074493	1.233307	L	* U
36	.306213	-.171666	1.111326	-.128130	1.238023	L	* U
37	.330950	-.276767	1.117694	-.182756	1.242632	L	* U
38	.355293	-.406595	1.123311	-.242541	1.246547	L	* U
39	.379323	-.564462	1.128303	-.308618	1.249997	L	* U
40	.403061	-.751660	1.132766	-.380094	1.253009	L	* U
41	.426629	-.968026	1.136781	-.456954	1.255933	L	* U
42	.450015	-.120638	1.140411	-.538575	1.258451	L	* U
43	.473280	-.069173	1.143711	-.624711	1.260779	L	* U
44	.496465	-.044646	1.146727	-.715514	1.262915	L	* U
45	.519582	-.078534	1.149499	-.810311	1.264868	L	* U
46	.542669	-.082692	1.152060	-.908301	1.266719	L	* U
47	.565740	-.086565	1.154441	-.101358	1.268429	L	* U
48	.588807	-.090196	1.156668	-.184236	1.270036	L	* U
49	.611881	-.093622	1.158766	-.268963	1.271598	L	* U
50	.634971	-.096881	1.160758	-.355959	1.273017	L	* U
51	.658084	-.100066	1.162665	-.444330	1.274407	L	* U
52	.681229	-.103032	1.164509	-.533952	1.275765	L	* U
53	.704417	-.105993	1.166309	-.624820	1.277097	L	* U
54	.727663	-.108921	1.168088	-.716820	1.278417	L	* U
55	.750988	-.111852	1.169865	-.810957	1.279747	L	* U
56	.774413	-.114821	1.171663	-.907109	1.281081	L	* U
57	.797988	-.117865	1.173504	-.1006594	1.282456	L	* U
58	.821742	-.121024	1.175413	-.1099171	1.283879	L	* U
59	.845736	-.124339	1.177408	-.1187874	1.285371	L	* U
60	.870039	-.127857	1.179524	-.1274740	1.286951	L	* U
61	.894736	-.131632	1.181791	-.1358111	1.288641	L	* U
62	.919931	-.135725	1.184244	-.1438138	1.290472	L	* U
63	.945737	-.140212	1.186927	-.1514933	1.292471	L	* U
64	.972737	-.145186	1.189893	-.1588821	1.294684	L	* U
65	1.000000	-.097086	1.160883	-.188877	1.215645	L	* U
		AIRFOIL TRAILING EDGE					
66	1.028899	-.043830	1.127893	-.043830	1.127893		* B
67	1.059441	-.033450	1.121357	-.033450	1.121357		* B
68	1.092135	-.029416	1.118797	-.029416	1.118797		* B
69	1.127723	-.026685	1.116684	-.026685	1.116684		* B
70	1.167391	-.022915	1.114670	-.022915	1.114670		* B
71	1.212946	-.019723	1.112838	-.019723	1.112838		* B
72	1.268145	-.016427	1.110536	-.016427	1.110536		* B
73	1.341184	-.013170	1.108455	-.013170	1.108455		* B
74	1.456833	-.010805	1.106942	-.010805	1.106942		* B
75	1.671514	-.008974	1.105758	-.008974	1.105758		* B
76	1.876313	-.007730	1.104970	-.007730	1.104970		* B
77	2.042561	-.009657	1.106206	-.009657	1.106206		* B
78	2.209055	-.017748	1.111379	-.017748	1.111379		* B
79	2.406774	-.034402	1.121952	-.034402	1.121952		* B
80	2.702997	-.043931	1.127956	-.043931	1.127956		* B
81	4.890373	-.041666	1.126494	-.041666	1.126494		* B

Figure 4-9.- Continued

Y-GRID VALUES		Y(J) J= 1 TO 64					
-2.000000	-1.812909	-1.642933	-1.488178	-1.347027	-1.218072	-1.100091	-.992019
-.892925	-.801993	-.718505	-.641829	-.571404	-.506738	-.447391	-.392976
-.343146	-.297595	-.256059	-.218269	-.184038	-.153167	-.125487	-.100852
-.079132	-.060216	-.044007	-.030425	-.019401	-.010882	-.004827	-.001265
.001205	.004827	.010882	.019401	.030425	.044007	.060216	.079132
.100852	.125487	.153167	.184038	.218269	.256059	.297595	.343146
.392976	.447391	.506738	.571404	.641829	.718505	.801993	.892925
.992019	1.100091	1.218072	1.347027	1.488178	1.642930	1.812909	2.000000

SONIC LINE OUTPUT

SONIC LINE COORDINATES

Y	X-SONIC	
2.000000	-.33996	.57889
1.81291	-.38427	.52907
1.64293	-.42050	.48093
1.48818	-.45917	.43681
1.34703	-.48533	.39794
1.21807	-.50724	.36341
1.10009	-.52586	.33294
.99202	-.54224	.30665
.89293	-.55682	.28327
.80199	-.56980	.26189
.71851	-.58131	.24204
.64183	-.59143	.22341
.57140	-.60026	.20628
.50674	-.60789	.19055
.44739	-.61441	.17532
.39298	-.61993	.16152
.34315	-.62455	.14836
.29759	-.62839	.13645
.25605	-.63155	.12429
.21827	-.63411	.11383
.18404	-.63617	.10299
.15317	-.63781	.09339
.12549	-.63910	.08355
.10085	-.64010	.07490
.07913	-.64086	.06589
.06022	-.64144	.05780
.04401	-.64187	.04983
.03042	-.64218	.04207
.01940	-.64240	.03470
.01088	-.64255	.02781
.00483	-.64265	.02163
.00121	-.64270	.01685
BODY LOCATION		
-.00121	-.64274	.16545
-.00483	-.64279	.10899
-.01088	-.64286	.11434
-.01940	-.64295	.12085
-.03042	-.64304	.12907
-.04401	-.64312	.13853
-.06022	-.64315	.14789
-.07913	-.64310	.15936
-.10085	-.64295	.17058
-.12549	-.64265	.18276
-.15317	-.64214	.19541
-.18404	-.64137	.20956
-.21827	-.64026	.22239
-.25605	-.63873	.23687
-.29759	-.63671	.25172
-.34315	-.63408	.26775
-.39298	-.63073	.28296
-.44739	-.62656	.30142
-.50674	-.62142	.32005
-.57140	-.61521	.33958
-.64183	-.60780	.36022
-.71851	-.59906	.38225
-.80199	-.58885	.40608
-.89293	-.57719	.43225
-.99202	-.56385	.46132
-1.10009	-.54874	.49293
-1.21807	-.53167	.52827
-1.34703	-.51218	.56770
-1.48818	-.48915	.61174
-1.64293	-.45651	.65995
-1.81291	-.41901	.71061
-2.00000	-.37572	.76136

Figure 4-9.- Continued

SHOCK WAVE DRAG AND TOTAL PRESSURE PROFILE OUTPUT

INVISCID WAKE PROFILES FOR INDIVIDUAL SHOCK WAVES WITHIN MOMENTUM CONTOUR

SHOCK 1 WAVE DRAG FOR THIS SHOCK= Y	THIS SHOCK= CD(Y)	.006437 PO/POINF
-1.81290940	.00074477	.99946609
-1.64293038	.00077644	.99944339
-1.48817799	.00104243	.99925270
-1.34702736	.00120194	.99913836
-1.21807223	.00134241	.99903765
-1.10009107	.00147094	.99894551
-.99201912	.00159340	.99885773
-.89292538	.00171416	.99877115
-.80199344	.00183630	.99868360
-.71850542	.00196176	.99859365
-.64182864	.00209155	.99850061
-.57140431	.00222580	.99840437
-.50673795	.00236387	.99830539
-.44739136	.00250436	.99820468
-.39297572	.00264520	.99810371
-.34314575	.00278373	.99800440
-.29759466	.00291688	.99790895
-.25604987	.00304137	.99781970
-.21826935	.00315408	.99773891
-.18403842	.00325233	.99766847
-.15316703	.00333429	.99760971
-.12548743	.00339920	.99755319
-.10085215	.00344746	.99752858
-.07913226	.00348062	.99750482
-.06021592	.00350102	.99749019
-.04400713	.00351149	.99748268
-.03042464	.00351495	.99748020
-.01940111	.00351407	.99748083
-.01088238	.00351110	.99748297
-.00482689	.00350777	.99748535
-.00120527	.00350520	.99748713
.00120527	.00350345	.99748845
.00482689	.00350038	.99749065
.01088238	.00349447	.99749488
.01940111	.00348451	.99750202
.03042464	.00346887	.99751324
.04400713	.00344554	.99752997
.06021592	.00341232	.99755377
.07913226	.00336708	.99758621
.10085215	.00330802	.99762855
.12548743	.00323402	.99768160
.15316703	.00314487	.99774551
.18403842	.00304139	.99781969
.21826935	.00292531	.99790290
.25604987	.00279911	.99799338
.29759466	.00266567	.99808903
.34314575	.00252802	.99818772
.39297572	.00238897	.99828739
.44739136	.00225099	.99838631
.50673795	.00211595	.99848312
.57140431	.00198513	.99857690
.64182864	.00185909	.99866726
.71850542	.00173771	.99875427
.80199344	.00162012	.99883857
.89292538	.00150465	.99892135
.99201912	.00138875	.99900443
1.10009107	.00126890	.99909035
1.21807223	.00114057	.99918235
1.34702736	.00099855	.99928416
1.48817799	.00074008	.99946945
1.64293038	.00072907	.99947735
1.81290940	.00067902	.99951322

SHOCK WAVE EXTENDS OUTSIDE CONTOUR
PRINTOUT OF SHOCK LOSSES ARE NOT AVAILABLE FOR REST OF SHOCK

DRAG COEFFICIENT OUTPUT

CALCULATION OF DRAG COEFFICIENT BY MOMENTUM INTEGRAL METHOD

BOUNDARIES OF CONTOUR USED	CONTRIBUTION TO CD
UPSTREAM X = -1.17661	COUP = 0.000000
DOWNSTREAM X = .727663	CDDOWN = .009068
TOP Y = 1.812909	COTOP = .005569
BOTTOM Y = -1.812909	CDBOT = .007056
BODY AFT OF X = .727663	CDBODY = .007356
TOTAL CONTRIBUTIONS AROUND CONTOUR =	.029048

THERE ARE 1 SHOCKS INSIDE CONTOUR. TOTAL CDWAVE = .006437

NOTE - ONE OR MORE SHOCKS EXTEND OUTSIDE OF CONTOUR
CDWAVE DOES NOT EQUAL TOTAL WAVE DRAG

DRAG CALCULATED FROM MOMENTUM INTEGRAL CD = .035485

TIME TO RUN CASE WAS 62.07 SECONDS.

(b) Output.
Figure 4-9.- Concluded

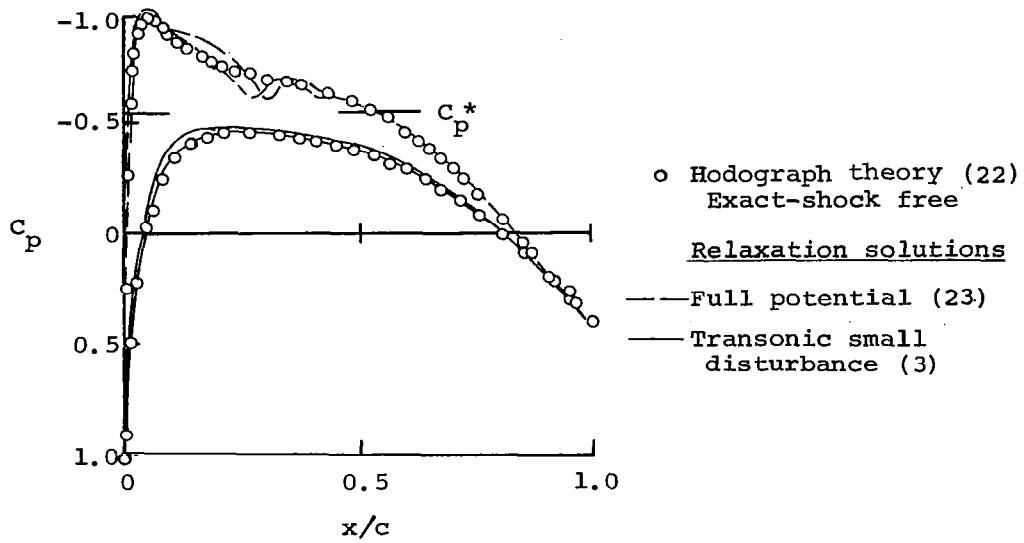


Figure 4-10.- Comparison of hodograph and relaxation solutions for the surface pressure distribution on a NLR quasi-elliptical airfoil with $\delta = 0.1212$, $\alpha = 1.32^\circ$ and $M_\infty = 0.7557$.

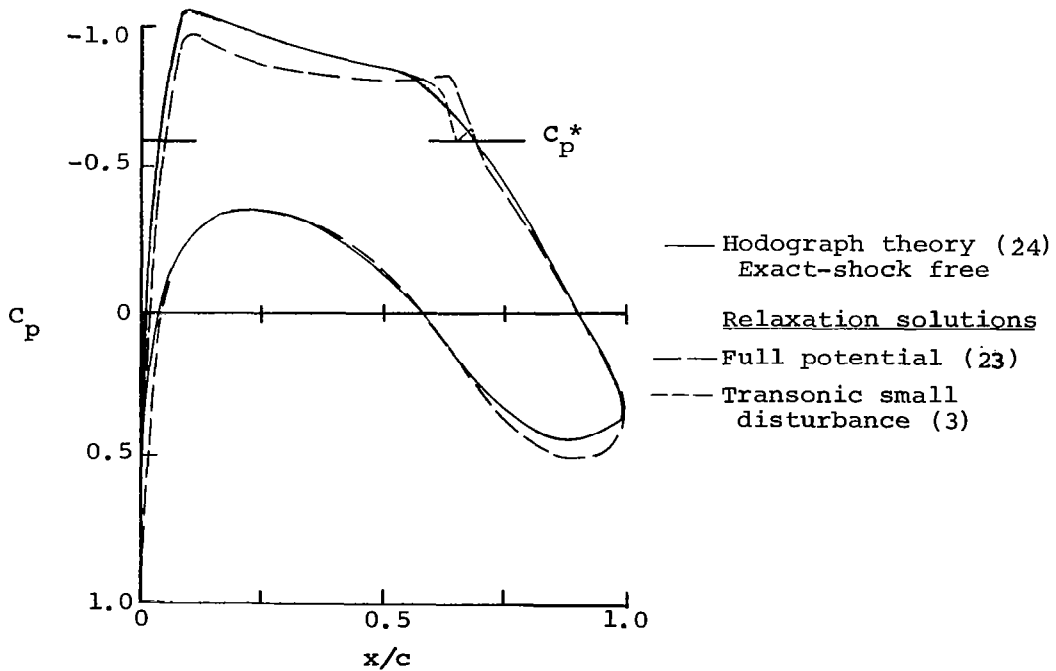


Figure 4-11.- Comparison of hodograph and relaxation solutions for the surface pressure distribution on a Korn airfoil with $\delta = 0.115$, $\alpha = 0.12^\circ$, and $M_\infty = 0.75$.

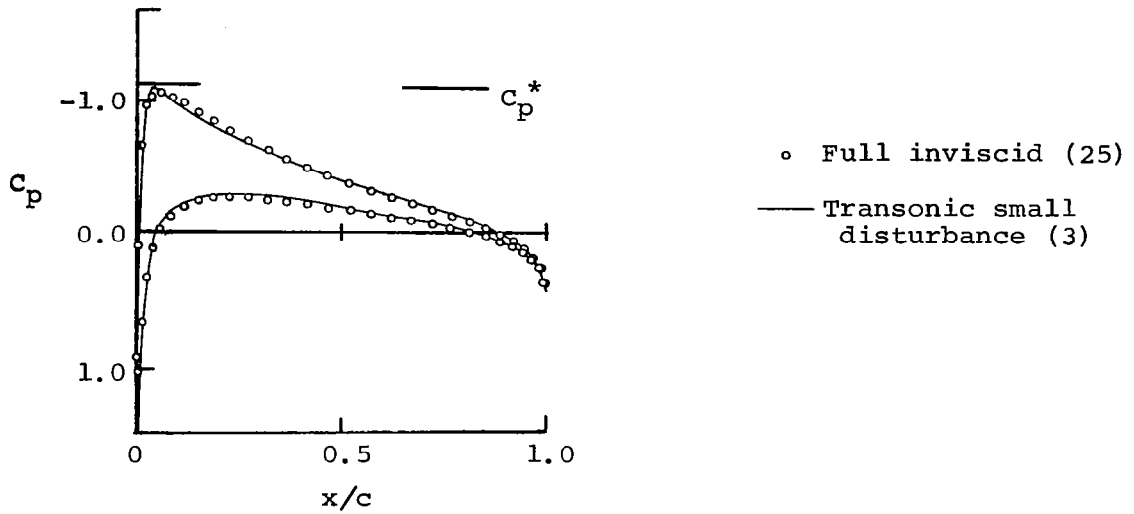


Figure 4-12.- Comparison of relaxation solutions for the surface pressure distribution of a NACA 0012 airfoil at $\alpha = 2^\circ$, $M_\infty = 0.630$.

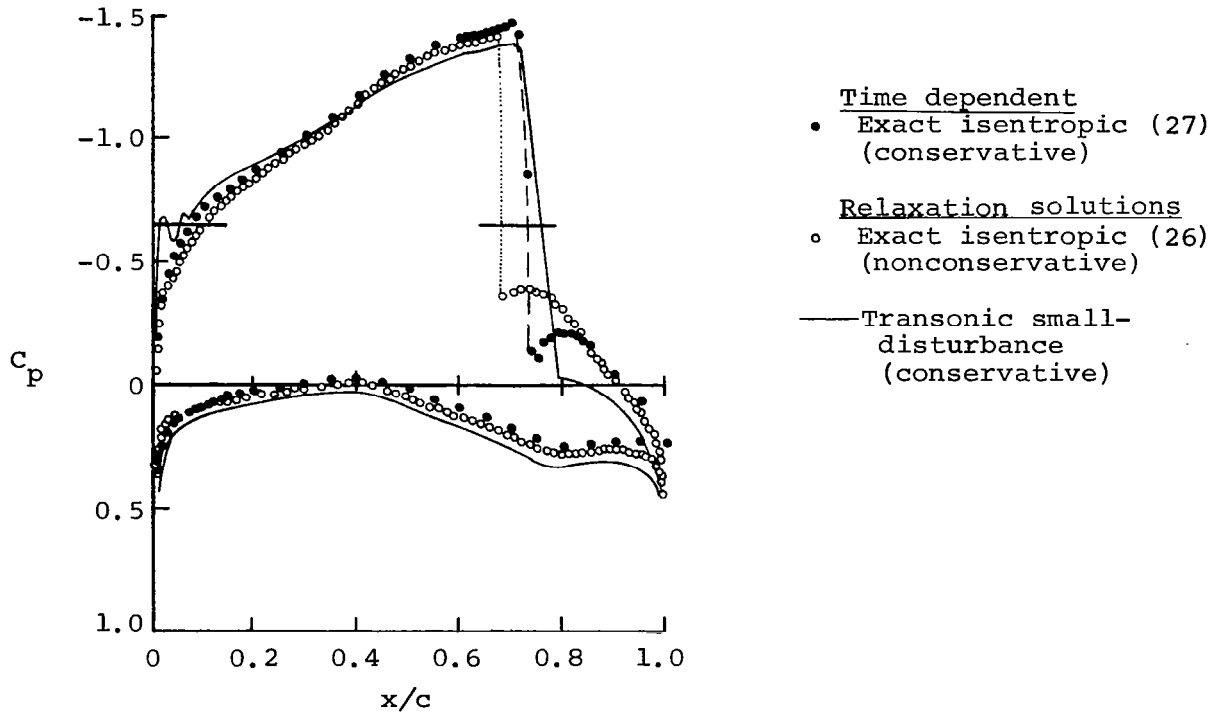


Figure 4-13.- Comparison of conservative time-dependent and nonconservative relaxation solutions of exact isentropic equation and the small-disturbance conservative solution for the surface pressure distribution on an NACA 64A410 airfoil at $M_\infty = 0.735$ and $\alpha = 1^\circ$.

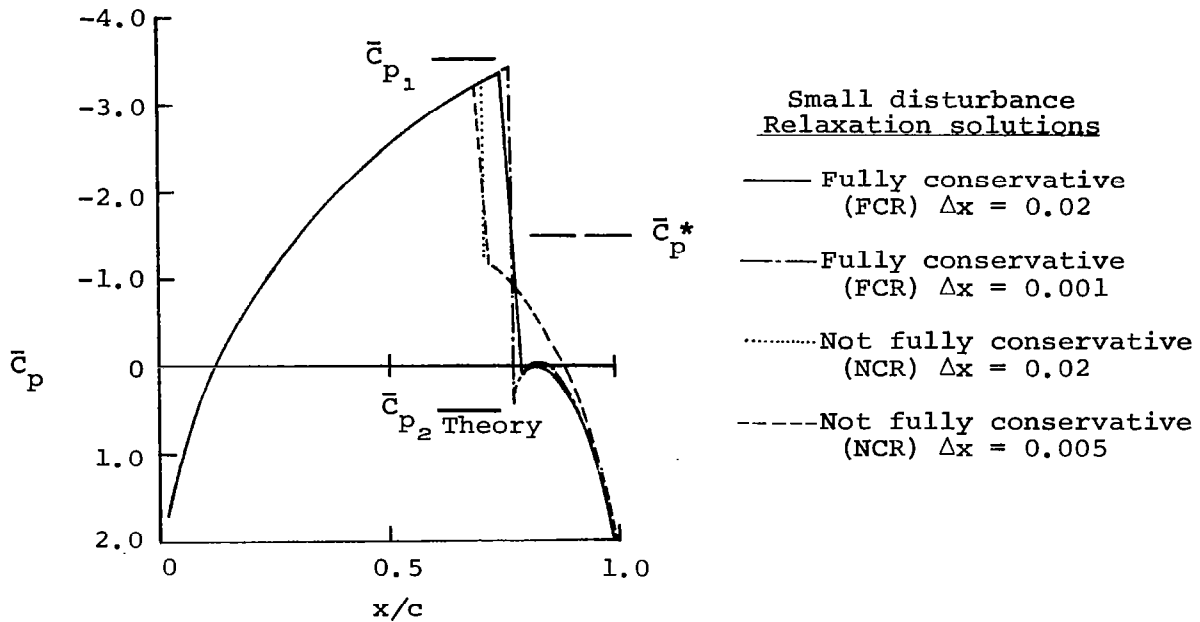


Figure 4-14.- Comparison of FCR and NCR solutions for the similarity surface pressure distribution on a parabolic arc airfoil with $\delta = 0.06$ and $M_\infty = 0.872$. (19)

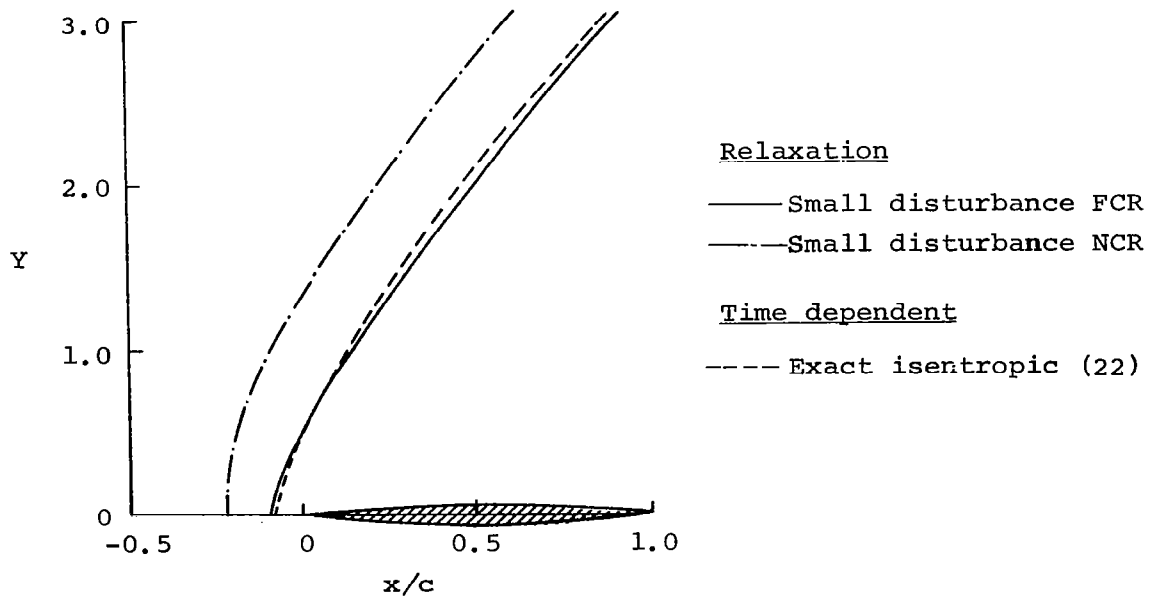


Figure 4-15.- Comparison of various solutions for the detached bow shock wave location for a parabolic arc airfoil with $\delta = 0.06$ and $M_\infty = 1.15$. (19)

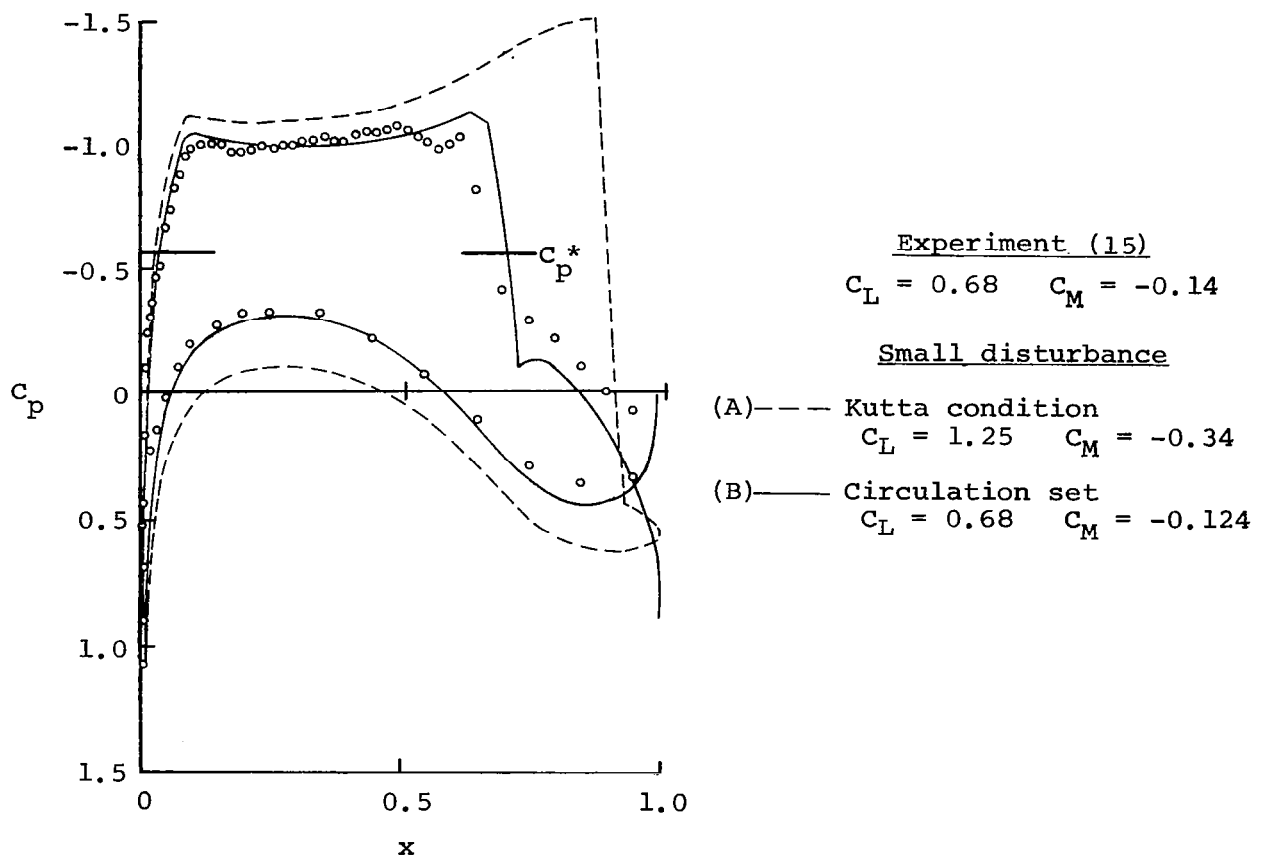


Figure 4-16.- Comparison of small disturbance FCR solutions with data for the surface pressure distribution on a Garabedian-Korn airfoil at $\alpha = 1.38^\circ$, $M_\infty = 0.768$. (19)

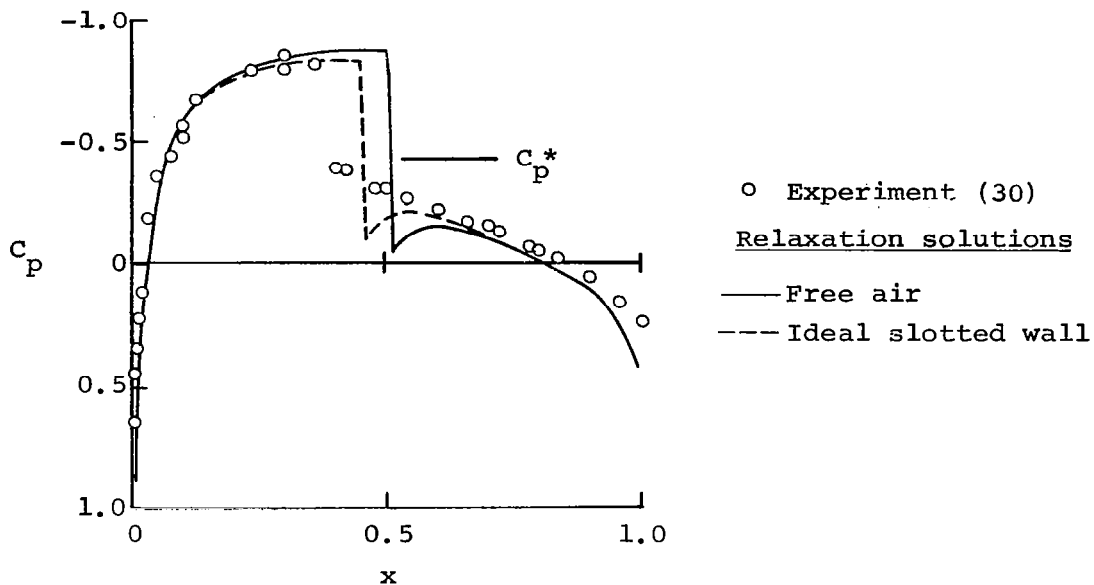


Figure 4-17.- Comparison of small disturbance FCR solutions with data for the surface pressure on a NACA 0012 at $\alpha = 0^\circ$ and $M = 0.8$. (19)

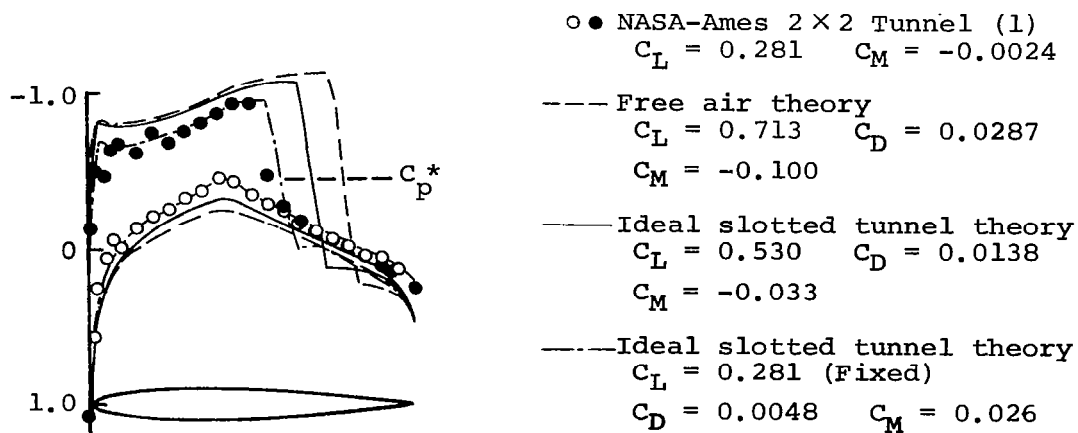


Figure 4-18.- Comparison of free-air and wind-tunnel small disturbance FCR solutions for the surface pressure distribution on a NACA 64A010 airfoil at $\alpha = 2.0^\circ$ and $M_\infty = 0.802$.

5. PROGRAMMER'S MANUAL

This section of the operational manual provides detailed information concerning the programming structure of the code, and is intended primarily to assist those users interested in modifying or extending the program. Included in this section are a dictionary of all subroutine variables that appear in common blocks in the program and a detailed description of the individual subroutines. These should provide all the information necessary to alter the code in a rapid and error-free fashion.

5.1 Dictionary of Subroutine Variables in Common

This section provides a dictionary of all subroutine variables which appear in common blocks in the program. The dictionary is arranged serially according to the named common block number rather than alphabetically in order to provide a more rapid access to the information.

<u>VARIABLE</u>	<u>COMMON BLOCK</u>	<u>DESCRIPTION AND COMMENTS</u>
	<u>COMMON</u>	
P(100,101)		array containing the small-disturbance velocity potential ϕ used in current calculation
X(100)		array containing the x-location of mesh points used in current calculation
Y(100)		array containing the y-location of mesh points used in current calculation
	<u>COM 1</u>	
IDOWN		downstream limit of I index for solution of difference equations; equal to IMAX - 1 for subsonic free streams, and IMAX otherwise
ILE		value of I index associated with the mesh point at the leading edge (x = 0.) or mesh point on the body closest to leading edge
IMAX		number of x-mesh points in current calculation
IMIN		value of I index designating the x-mesh point where the calculation is to start; user-option input, default value equal to 1

<u>VARIABLE</u>	<u>COMMON BLOCK</u>	<u>DESCRIPTION AND COMMENTS</u>
	<u>COM 1</u> (cont'd)	
ITE		value of I index associated with the mesh point at the trailing edge (x = 1.) or mesh point on body closest to trailing edge
IUP		upstream limit of I index for solution of difference equations; equal to IMIN + 1 for subsonic free streams and IMIN + 2 otherwise
JBOT		lower limit of J index for solution of finite-difference equations; equal to JMIN + 1 for free-air subsonic calculations, free-jet calculations, and porous tunnel wall calculations where the scaled porosity is greater than 1.5; equal to JMIN otherwise
JLOW		value of J index associated with the row of mesh points immediately below the body
JMAX		number of y-mesh points in current calculation
JMIN		value of J index designating the y-mesh point where the calculation is to start; user-option input, default value equal to 1
JTOP		upper limit of J index for solution of finite-difference equations; equal to JMAX - 1 for free-air subsonic calculations, free-jet calculations, and porous tunnel wall calculations where the scaled porosity is greater than 1.5; equal to JMAX otherwise
JUP		value of J index associated with the row of mesh points immediately above the body
	<u>COM 2</u>	
AK		transonic similarity parameter; equal to $(1 - M_\infty^2)/M_\infty^2 m \delta^{2/3}$ where m is equal to zero for Cole scaling, 2/3 for Spreiter scaling, and 1/2 for Krupp scaling
ALPHA		angle of attack; user-option input, default value equal to 0.12°
DUB		strength of doublet characterizing far-field behavior; set equal to airfoil volume for nonlifting free-air flows; equal to volume $+ (\gamma+1)/4 \iint \phi_x^2 dx dy$ otherwise
GAM1		equal to $\gamma + 1$
RTK		equal to $\sqrt{ AK }$

<u>VARIABLE</u>	<u>COMMON BLOCK</u>	<u>DESCRIPTION AND COMMENTS</u>
	<u>COM 3</u>	
ABORT (logical)		program control for terminating calculation if solution diverges
ICUT		control for mesh cut and refinement; equal to 0, input mesh used to convergence; equal to 1, input mesh may be cut once; equal to 2, input mesh may be cut twice; user-option input, default value equal to 2
IREF		internal program control for mesh cutting
KSTEP		internal program control for index increment for mesh cutting
	<u>COM 4</u>	
XIN(100)		array of x-ordinates of user-supplied x-grid; set equal to XKRUPP if default x-grid chosen
YIN(100)		array of y-ordinates of user-supplied y-grid; set equal to YFREE or YTUN if default y-grid chosen
AMESH (logical)		control for use of analytical default mesh calculated by AYMESH; user-option input, default value is FALSE
	<u>COM 5</u>	
XDIFF(100)		array of difference coefficients for determining ϕ_x ; equal to $1./(X(I) - X(I-1))$
YDIFF(100)		array of difference coefficients for determining ϕ_y ; equal to $1./(Y(J) - Y(J-1))$
	<u>COM 6</u>	
CAMBER(100)		array of ordinates of airfoil camber-line distribution
FL(100)		array of y-ordinates of airfoil lower surface
FU(100)		array of y-ordinates of airfoil upper surface
FXL(100)		array of slopes (dF_l/dx) of airfoil lower surface
FXU(100)		array of slopes (dF_u/dx) of airfoil upper surface
IFOIL		number of mesh points on airfoil; equal to ITE - ILE + 1

<u>VARIABLE</u>	<u>COMMON BLOCK</u>	<u>DESCRIPTION AND COMMENTS</u>
	<u>COM 6</u> (con't)	
THICK(100)		array of ordinates of airfoil thickness distribution
VOL		volume of airfoil/unit span
XFOIL(100)		array of x-ordinates of mesh points on airfoil
	<u>COM 7</u>	
CJLOW		difference coefficient for extrapolation formulas to obtain airfoil lower surface properties; equal to $-Y(JLOW-1)/(Y(JLOW) - Y(JLOW-1))$
CJLOW1		difference coefficient for extrapolation formulas to obtain airfoil lower surface properties; equal to $-Y(JLOW)/(Y(JLOW) - Y(JLOW-1))$
CJUP		difference coefficient for extrapolation formulas to obtain airfoil upper surface properties; equal to $Y(JUP+1)/(Y(JUP+1) - Y(JUP))$
CJUP1		difference coefficient for extrapolation formulas to obtain airfoil upper surface properties; equal to $Y(JUP)/(Y(JUP+1) - Y(JUP))$
	<u>COM 8</u>	
CVERGE		convergence criterion for maximum iteration error of ϕ ; user-option input, default value equal to 0.00005
DVERGE		divergence criterion for maximum iteration error ϕ ; user-option input, default value equal to 10.
EPS		coefficient of pseudo-time term $\Delta t \phi_{xt} / \Delta x$ in differential equation; value between 0. and 1.; default value equal to 0.
IPRTER		control for print frequency of y-line in mesh where error is largest; for example, IPRTER = 10 implies line will be printed every 10th iteration; user-option input, default value equal to 10
MAXIT		control for maximum number of iteration cycles allowed on each mesh; user-option input, default value equal to 500

<u>VARIABLE</u>	<u>COMMON BLOCK</u>	<u>DESCRIPTION AND COMMENTS</u>
	<u>COM 8</u> (cont'd)	
WE(3)		relaxation factor at elliptic points on the coarse, medium, and fine mesh, respectively; user-option input, default values in order are 1.8, 1.9, and 1.95; if user-specified, all three values must be given
	<u>COM 9</u>	
BCFOIL (integer)		control for airfoil geometry options; equal to 1 for symmetric NACA four-digit (OOXX) series; equal to 2 for parabolic arc; equal to 3 for user-specified ordinates, program calculates ordinates and slopes at mesh points by cubic spline interpolation; equal to 4 for complete user-supplied description; user-option input, default value is 3 and program uses Korn airfoil ordinates
NL		number of ordinate points used to describe airfoil lower surface; default value (for Korn airfoil) is 75
NU		number of ordinate points used to describe airfoil upper surface; default value (for Korn airfoil) is 100
RIGF		Reigles rule nose correction factor for modifying airfoil slopes; user-option input, default value is 0.0
XL(100)		array of x-ordinates for airfoil lower surface; user-supplied if BCFOIL = 3, unless default (Korn) airfoil is used
XU(100)		array of x-ordinates for airfoil upper surface; user-supplied if BCFOIL = 3, unless default (Korn) airfoil is used
YL(100)		array of y-ordinates for airfoil lower surface; user-supplied if BCFOIL = 3, unless default (Korn) airfoil is used
YU(100)		array of y-ordinates for airfoil upper surface; user-supplied if BCFOIL = 3, unless default (Korn) airfoil is used
	<u>COM 10</u>	
GAM		ratio of specific heats; user-option input, default value equal to 1.4
JMXF		number of y-mesh points for basic Krupp free-air grid; equal to 57
JMXT		number of y-mesh points for basic wind-tunnel grid; equal to 49

<u>VARIABLE</u>	<u>COMMON BLOCK</u>	<u>DESCRIPTION AND COMMENTS</u>
	<u>COM 10</u> (cont'd)	
XGRDIN		control for input of x-grid; equal to TRUE if x-grid is user-specified; equal to FALSE if default x-grid (XKRUPP) is used; default value equal to FALSE
XKRUPP(100)		array of x-ordinates for basic Krupp free-air grid
YFREE(100)		array of y-ordinates for basic Krupp free-air grid
YGRDIN		control for input of y-grids; equal to TRUE if y-grid is user-specified; equal to FALSE if either default y-grid (YFREE or YTUN) is used; default value equal to FALSE
YTUN(100)		array of y-ordinates for basic wind-tunnel grid
	<u>COM 11</u>	
ALPHAO		value of angle of attack (ALPHA) from previous calculation
CLOLD		value of lift coefficient (CL) from previous calculation
DELTAO		value of thickness ratio (DELTA) from previous calculation
DUBO		value of far-field doublet strength (DUB) from previous calculation
EMACHO		value of Mach number (EMACH) from previous calculation
IMAXI		number of x-mesh points; less than or equal to 100; user-option input, default value is 77 for basic Krupp grid
IMAXO		number of x-mesh points (IMAX) used in previous calculation
IMINO		value of I index (IMIN) where previous calculation was initiated
JMAXI		number of y-mesh points; less than or equal to 100; user-option input, default value is 57 (JMXF) for basic Krupp free-air grid or 49 (JMXT) for basic wind-tunnel grid

<u>VARIABLE</u>	<u>COMMON BLOCK</u>	<u>DESCRIPTION AND COMMENTS</u>
	<u>COM 11</u> (cont'd)	
JMAXO		number of y-mesh points (JMAX) used in previous calculation
JMINO		value of J index (JMIN) where calculation was initiated
PSAVE (logical)		control for saving restart block of values on unit 3; equal to TRUE for final results of current calculation to be saved; user-option input, default value is FALSE
PSTART (integer)		control for initializing velocity potential ϕ array; equal to 1, ϕ set to zero; equal to 2, ϕ read from tape unit 7; equal to 3, ϕ used from previous case; user-option input, default value equal to 1
TITLE(8)		user-input alphanumeric array containing run title information
TITLEO(8)		alphanumeric array containing run title information from previous run
VOLO		value of airfoil volume (per unit span) from previous calculation
XOLD(100)		array of x-mesh ordinates from previous calculation
YOLD(100)		array of y-mesh ordinates from previous calculation
	<u>COM 12</u>	
F		wind-tunnel slot parameter; user-option input, default value equal to 0.
H		wind-tunnel half height to chord ratio; user-option input, default value equal to 0.
HALFPI		equal to $\pi/2$
PI		equal to π
RTKPOR		equal to $\sqrt{ AK } / \text{POR}$
TWOPI		equal to 2π

<u>VARIABLE</u>	<u>COMMON BLOCK</u>	<u>DESCRIPTION AND COMMENTS</u>
<u>COM 13</u>		
CDFACT		transonic scale factor for drag coefficient; equal to $\delta^{4/3} M_\infty^{-n}$ where n equals 0 for Cole scaling, 2/3 for Spreiter scaling, and 3/4 for Krupp scaling
CLFACT		transonic scale factor for lift coefficient; equal to $\delta^{2/3} M_\infty^{-n}$ (see CDFACT)
CMFACT		transonic scale factor for moment coefficient; equal to $\delta^{2/3} M_\infty^{-n}$ (see CDFACT)
CPFACT		transonic scale factor for pressure coefficient; equal to $\delta^{2/3} M_\infty^{-n}$ (see CDFACT)
CPSTAR		critical pressure coefficient
<u>COM 14</u>		
CLSET		specified lift coefficient; used if Kutta condition is FALSE; user-option input, default value equal to 0.
FCR (logical)		control for fully conservative differencing; equal to TRUE for fully conservative form; otherwise, difference equations not conservative at shock waves; user-option input, default value equal to TRUE
KUTTA (logical)		control for Kutta condition specification; equal to TRUE for Kutta condition enforcement; equal to FALSE for lift-coefficient specification; user-option input, default value equal to TRUE
WCIRC		relaxation factor for circulation; user-option input, default value equal to 1.
<u>COM 15</u>		
B		constant in subsonic far-field doublet solution for various wind-tunnel walls; equal to 0.5 for solid wall; equal to 0. for free-jet, slotted, or perforated/porous walls
BETA0		constant in subsonic far-field vortex solution for various wind-tunnel walls; calculated by VROOTS for ideal slotted tunnel wall; equal to $\pi/2$ for solid wall; equal to 0. for free jet; equal to $\tan^{-1}(RTK/POR)$ for ideal perforated/porous wall

<u>VARIABLE</u>	<u>COMMON BLOCK</u>	<u>DESCRIPTION AND COMMENTS</u>
	<u>COM 15 (cont'd)</u>	
BETA1	see BETA0	
BETA2	see BETA0	
PSI0		constant in subsonic far-field vortex solution for various wind-tunnel walls; calculated by VROOTS for ideal slotted tunnel wall; equal to 1. for solid wall, free-jet, and ideal perforated/porous wall
PSI1	see PSI0	
PSI2	see PSI0	
	<u>COM 16</u>	
ALPHA0		constant in subsonic far-field doublet solution for various wind-tunnel walls; calculated by DROOTS for ideal slotted tunnel wall; equal to π for solid wall; equal to $\pi/2$ for free-jet; equal to $\pi/2 - \tan^{-1}(RTK/POR)$ for ideal perforated/porous wall
ALPHA1	see ALPHA0	
ALPHA2	see ALPHA0	
JET (real)		constant in subsonic far-field vortex solution for various wind-tunnel walls; equal to 0. for solid and ideal perforated/porous wall; equal to 0.5 for free-jet and ideal slotted walls
OMEGA0		constant in subsonic far-field doublet solution for various wind-tunnel walls; calculated by DROOTS for ideal slotted tunnel wall; equal to 1. for solid wall, free-jet, and ideal perforated/porous walls
OMEGA1	see OMEGA0	
OMEGA2	see OMEGA0	
XSING		x-location of vortex and doublet used to determine far-field behavior on outer boundaries; equal to 0.5
	<u>COM 17</u>	
CYYBLC		special ϕ_{yy} difference coefficient at airfoil lower surface; equal to $-CYYBLU/(Y(JLOW) - Y(JLOW-1))$

<u>VARIABLE</u>	<u>COMMON BLOCK</u>	<u>DESCRIPTION AND COMMENTS</u>
	<u>COM 17</u> (cont'd)	
CYYBLD		special ϕ_{yy} difference coefficient at airfoil lower surface; equal to -CYYBLC
CYYBLU		special ϕ_{yy} difference coefficient at airfoil lower surface; equal to $-2./(Y(JLOW) + Y(JLOW-1))$
CYYBUC		special ϕ_{yy} difference coefficient at airfoil upper surface; equal to $CYYBUD/(Y(JUP+1) - Y(JUP))$
CYYBUD		special ϕ_{yy} difference coefficient at airfoil upper surface; equal to $-2./(Y(JUP+1) + Y(JUP))$
CYYBUU		special ϕ_{yy} difference coefficient at airfoil upper surface; equal to -CYYBUC
FXLBC(100)		array containing body-slope boundary condition on lower surface multiplied by mesh spacing constant CYYBLU
FXUBC(100)		array containing body-slope boundary condition on upper surface multiplied by mesh spacing constant CYYBUD
ITEMP1		dummy index
ITEMP2		dummy index
	<u>COM 18</u>	
DCIRC		iteration error in circulation; equal to $\Delta\phi_{te}^{n+1} - \Delta\phi_{te}^n$
ERROR		absolute value of the maximum iteration error in ϕ ; that is, $ \phi_{j,i}^n - \phi_{j,i}^{n-1} _{\max}$
I1		dummy I index used to alternate storage in arrays POLD (J,I) and EMU (J,I)
I2		dummy I index used to alternate storage in arrays POLD (J,I) and EMU (J,I)
IERROR		value of I index where maximum iteration error in ϕ occurs
JERROR		value of J index where maximum iteration error in ϕ occurs
OUTERR (logical)		internal program control for printout of iteration count, CL, CM, CD, IERROR, JERROR, ERROR, and ERCIRC
POLD(100,2)		array used to store current and previous column of $\phi_{j,i}$

<u>VARIABLE</u>	<u>COMMON BLOCK</u>	<u>DESCRIPTION AND COMMENTS</u>
	<u>COM 18</u> (cont'd)	
VC(100)		array used to store current column of $2 \cdot (1 - M_1^2) / (X_{i+1} - X_{i-1})$; equal to $C1(I) - (CXL(I) \cdot POLD(J, I2) + CXC(I) \cdot P(J, I) + CXR(I) \cdot P(J, I+1))$
	<u>COM 19</u>	
WI		reciprocal of elliptic relaxation factor for current mesh
DIAG(100)		array of diagonal elements of tridiagonal matrix equation for column solution of ϕ
RHS(100)		array for the right-hand side column vector of tridiagonal matrix equation for column solution of ϕ
SUB(100)		array of sub-diagonal elements of tridiagonal matrix equation for column solution of ϕ
SUP(100)		array of super-diagonal elements of tridiagonal matrix equation for column solution of ϕ
	<u>COM 20</u>	
XMID(100)		array of x-mesh points obtained by deleting every other point in input array XIN starting with $I = IMIN + 1$
YMID(100)		array of y-mesh points obtained by deleting every other point in input array YIN starting with $J = JMIN + 1$
	<u>COM 22</u>	
CXC(100)		centered ϕ_x difference coefficient divided by $X(I+1) - X(I-1)$ for $IMIN + 1 \leq I \leq IMAX - 1$; equal to $-CXL(I) - CXR(I)$
CXL(100)		left-biased ϕ_x difference coefficient divided by $X(I+1) - X(I-1)$ for $IMIN + 1 \leq I \leq IMAX - 1$; equal to $-0.5 \cdot (\gamma + 1) / ((X(I) - X(I-1)) \cdot (X(I+1) - X(I-1)))$
CXR(100)		right-biased ϕ_x difference coefficient divided by $X(I+1) - X(I-1)$ for $IMIN + 1 \leq I \leq IMAX - 1$; equal to $0.5 \cdot (\gamma + 1) / ((X(I+1) - X(I)) \cdot (X(I+1) - X(I-1)))$
CXXC(100)		centered ϕ_{xx} difference coefficient multiplied by $X(I+1) - X(I-1)$ for $IMIN + 1 \leq I \leq IMAX - 1$; equal to $-CXXL(I) - CXXR(I)$

<u>VARIABLE</u>	<u>COMMON BLOCK</u>	<u>DESCRIPTION AND COMMENTS</u>
	<u>COM 22</u> (cont'd)	
CXXL(100)		left-biased ϕ_{xx} difference coefficient multiplied by $X(I+1) - X(I-1)$ for $IMIN + 1 \leq I \leq IMAX - 1$; equal to $2./(X(I) - X(I-1))$
CXXR(100)		right-biased ϕ_{xx} difference coefficient multiplied by $X(I+1) - X(I-1)$ for $IMIN + 1 \leq I \leq IMAX - 1$; equal to $2./(X(I+1) - X(I))$
C1(100)		equal to $AK/(X(I+1) - X(I-1))$
	<u>COM 23</u>	
CYYC(100)		centered ϕ_{yy} difference coefficient; equal to $-CYYD(J) - CYYU(J)$ for $JMIN + 1 \leq J \leq JMAX - 1$
CYYD(100)		down-biased ϕ_{yy} difference coefficient; equal to $2./((Y(J) - Y(J-1))*(Y(J+1) - Y(J-2)))$ for $JMIN + 1 \leq J \leq JMAX - 1$
CYYU(100)		up-biased ϕ_{yy} difference coefficient; equal to $2./((Y(J+1) - Y(J))*(Y(J+2) - Y(J-2)))$ for $JMIN + 1 \leq J \leq JMAX - 1$
IVAL		index identifying the current column location; equal to I
	<u>COM 24</u>	
DBOT(100)		array of subsonic, far-field, unit-strength doublet values on bottom boundary
DDOWN(100)		array of subsonic, far-field, unit-strength doublet values on downstream boundary
DTOP(100)		array of subsonic, far-field, unit-strength doublet values on top boundary
DUP(100)		array of subsonic, far-field, unit-strength doublet values on upstream boundary
VBOT(100)		array of subsonic, far-field, unit-strength vortex values on bottom boundary
VDOWN(100)		array of subsonic, far-field, unit-strength vortex values on downstream boundary
VTOP(100)		array of subsonic, far-field, unit-strength vortex values on top boundary
VUP(100)		array of subsonic, far-field, unit-strength vortex values on upper boundary

<u>VARIABLE</u>	<u>COMMON BLOCK</u>	<u>DESCRIPTION AND COMMENTS</u>
	<u>COM 25</u>	
CPL(100)		array of pressure coefficient values on $y = 0^-$ lines from $IMIN \leq I \leq IMAX$
CPU(100)		array of pressure coefficient values on $y = 0^+$ line from $IMIN \leq I \leq IMAX$
	<u>COM 26</u>	
PJUMP(100)		array of values for the jump in potential across the wake from $ITE \leq I \leq IMAX$; equal to $CIRCTE + (X(I) - 1.)*(CIRCFF-CIRCTE)/X(IMAX1$ $- 1.)$
	<u>COM 27</u>	
CL		lift coefficient
DELRT2		equal to $(\delta)^{2/3}$
DELTA		airfoil thickness ratio δ ; user-option input, default value equal to 0.115
EMACH		free-stream Mach number M_∞ ; user-option input, default value equal to 0.75
EMROOT		equal to $(M_\infty)^{2/3}$
PHYS (logical)		control for transonic scaling used for input/ output; equal to TRUE for physically scaled values and FALSE for transonically scaled values; user-option input, default value equal to TRUE
PRTFLO (integer)		control for print option of final flow field; equal to 1 for no flow field printout, equal to 2 for entire flow field printout, equal to 3 for printout of 3J lines around maximum error; user-option input, default value equal to 1
SIMDEF (integer)		control for transonic similarity scaling definition; equal to 1 for Cole scaling, equal to 2 for Spreiter scaling, equal to 3 for Krupp scaling, equal to 4 for user- supplied scaling; user-option input, default value equal to 3
SONVEL		perturbation velocity at sonic point; equal to AK/GAM1
VFACT		transonic scale factor for flow angle deflection; equal to $\delta \cdot (180^\circ/\pi)$

<u>VARIABLE</u>	<u>COMMON BLOCK</u>	<u>DESCRIPTION AND COMMENTS</u>
	<u>COM 27</u> (cont'd)	
YFACT		transonic scale factor for lateral coordinate; equal to $\delta^{-1/3} M_\infty^{-m}$ where m equals 0 for Cole scaling, 2/3 for Spreiter scaling, and 1/2 for Krupp scaling
	<u>COM 28</u>	
BCTYPE (integer)		input control for type of flow to be computed; equal to 1 for free air, 2 for solid wall tunnels, 3 for free jet, 4 for ideal slotted-wall tunnel, and 5 for ideal perforated/porous wall tunnel; user-option input, default value equal to 1
CIRCF		circulation in far field
CIRCTE		jump in potential at the trailing edge, $\Delta\phi_{te}$
FHINV		equal to $1/(F*H)$
POR		wall porosity factor; user-option input, default value equal to 0.
	<u>COM 30</u>	
		common block COM 30 is a scratch common used to shorten storage requirements; all variables appearing in COM 30 are used internally only within each subroutine and not passed for use to other subroutines
	<u>COM 32</u>	
BIGRL		maximum residual of finite difference equation; equal to $(R_{j,i})_{max}$
IRL		value of I index where maximum residual occurs
JRL		value of J index where maximum residual occurs
	<u>COM 33</u>	
THETA(100,100)		array containing values of the angle used to update the potential for subsonic free-air flows by spreading the circulation rapidly through the flow field; calculated by ANGLE; equal to $1/2\pi \tan^{-1}(\sqrt{K} \bar{y}/(X_1 - XSING))$
	<u>SPLN</u>	
A(200)		coefficient array for cubic-fit, continuous derivative interpolation formula for fitting tabulated airfoil ordinates

<u>VARIABLE</u>	<u>COMMON BLOCK</u>	<u>DESCRIPTION AND COMMENTS</u>
	<u>SPLN</u> (cont'd)	
B(200)		coefficient array for cubic-fit, continuous derivation interpolation formula for fitting tabulated airfoil ordinates
DY1		slope of airfoil upper or lower surface at leading edge
DY2		slope of airfoil upper or lower surface at trailing edge
DYP		slope of airfoil at XP calculated by cubic spline interpolation SPLN1
K1		index for internal control of SPLN1 subroutine; equal to 1 implies first derivative specified, equal to 2 implies second derivative specified
K2		index for internal control of SPLN1 subroutine; equal to 1 implies first derivative specified, equal to 2 implies second derivative specified
XP		x-location of current mesh point on airfoil XIN(I), ILE ≤ I ≤ ITE
YP		y-ordinate of airfoil at XP calculated by cubic spline interpolation

5.2 Description of Individual Subroutines

This section provides a description of the individual subroutines in the program, including the main program. The descriptions are ordered alphabetically according to subroutine name.

5.2.1 Subroutine ANGLE.- Subroutine ANGLE computes the quantity THETA(J,I) at the input mesh points (I,J) according to

$$\text{THETA}(J,I) = \frac{1}{2\pi} \tan^{-1} \left(\frac{\sqrt{K} \tilde{Y}_j}{x_i - \text{XSING}} \right)$$

where XSING = 0.5. For $R = \sqrt{\tilde{Y}_j^2 + (x_i - \text{XSING})^2} \leq 1$, the quantity is reduced by $\text{THETA}(J,I) = R * \text{THETA}(J,I)$. These values are used in subroutine SOLVE to enhance the convergence characteristics of subsonic free-air flows by transmitting the effect of updated circulation throughout the flow field. This is accomplished by updating the potential at the end of each iteration sweep in x by

$$\phi_{j,i}^{n+1} = \phi_{j,i}^{n+1} + (\Gamma^{n+1} - \Gamma^n) \cdot \text{THETA}(J,I)$$

5.2.2 Subroutine ARF(X).- Subroutine ARF(X) evaluates the error function

$$\text{ARF}(X) = \frac{2}{\sqrt{\pi}} \int_0^X e^{-t^2} dt$$

by using the rational fraction approximation given in the Handbook of Mathematical Functions, AMS 55, edited by M. Abramowitz and L. A. Stegun, Fifth Printing, August 1966, p. 299, Equation 7.1.26.

5.2.3 Subroutine AYMESH.- Subroutine AYMESH computes an analytical (x, \tilde{y}) mesh based in part on the formulation in reference 11. That mesh formulation was designed for $M_\infty < 1$ free-air flows; however, it can be used for $M_\infty > 1$ flows as well as for wind tunnel environments. A coordinate transformation in x is made according to the following formula:

$$x(x) = (1 - a_0) \left\{ a_1 x e^{-(a_2 x)^2} + \left[1 - e^{-(a_7 x)^2} \right] \frac{2}{\sqrt{\pi}} \int_0^{a_4 x} e^{-t^2} dt \right\} + \frac{2a_0}{\pi} \tan^{-1} \left[a_5 (x + a_0) \right]$$

An initial $x^{(1)}$ mesh is chosen by setting

$$x_I^{(1)} = \tan \left[\frac{\pi}{2} \left(\frac{I - 201}{200} \right) \right]_{I=201}^{401}$$

$$x_I^{(1)} \Big|_{I=1}^{200} = -x_{402-I}^{(1)}$$

Corresponding $x_I^{(1)}$ ($x_I^{(1)}$) are then computed from the above stretching with the following coefficient values: $a_0 = 0.225$, $a_1 = 1.4$, $a_2 = 1.6$, $a_4 = 0.75$, $a_5 = 30.0$, $a_6 = 0.603$, and $a_7 = 2.0$. Then, for a fixed ΔX increment equal to $\Delta X = 2/(\text{IMAXI}-1)$, the corresponding x_i points are found by using a quadratic interpolation scheme with the $x_I^{(1)}$, $x^{(1)}$ arrays. The final x_i mesh is found by using $x_i = x_i + a_3$ where $a_3 = 0.6188$.

The \tilde{y} array definition depends on whether the calculation is for free air or a tunnel environment. For free air

$$\tilde{Y}_j = 5.3 \frac{\tan \left[\frac{\pi}{2} \cdot \frac{j - \frac{JMAXI}{2}}{\left(\frac{JMAXI}{2} + 1\right)} \right]}{\tan \left[\frac{\pi}{2} \cdot \frac{\frac{JMAXI}{2}}{\frac{JMAXI}{2} + 1} \right]}, \quad \frac{JMAX}{2} + 1 \leq j \leq JMAXI$$

$$\tilde{Y}_j = -\tilde{Y}_{JMAXI+1-j}, \quad 1 \leq j \leq \frac{JMAXI}{2}$$

while for a tunnel environment

$$\tilde{Y}_j = \tan^2 \left[\frac{\pi}{2} \cdot \frac{j - \frac{JMAXI}{2}}{JMAXI} \right], \quad \frac{JMAX}{2} + 1 \leq j \leq JMAXI$$

$$\tilde{Y}_j = -\tilde{Y}_{JMAXI+1-j}, \quad 1 \leq j \leq \frac{JMAXI}{2}$$

The default values for IMAXI and JMAXI used with this analytical mesh are IMAXI = 81, JMAXI = 64. These values produce a mesh sufficiently fine for almost all aerodynamic applications and permit the mesh halving option to be used twice; that is, fine to medium to coarse.

5.2.4 Subroutine BCEND (I,JEND,L).- Subroutine BCEND modifies the diagonal DIAG(J) and right-hand side RHS(J) elements of the tridiagonal matrix in the tridiagonal matrix equation for the residual of the velocity potential along a column to include the boundary conditions along the bottom (J = JBOT) and top (J = JTOP) rows. The routine first sets the index I (I = IVAL) to correspond to the current column being solved by SYOR and then branches to the appropriate modification of the type of flow environment being considered according to the following schedule:

For the top boundary,

BCTYPE	JTOP	DIAG MODIFICATION	RHS MODIFICATION
1(K) > 0)	JMAXI-1	----	----
1(K) < 0)	JMAX	$\frac{-\sqrt{-K} \cdot \text{CYYU}(\text{JTOP})}{x_i - x_{i-1}}$	$\frac{\sqrt{-K} \cdot \text{CYYU}(\text{JTOP}) \cdot (\phi_{\text{JMAX},i} - \phi_{\text{JMAX},i-1})}{x_i - x_{i-1}} + \text{CYYU}(\text{JTOP}) \cdot \phi_{\text{JTOP}+1,i}$
2	JMAX	----	----
3(K > 0)	JMAX-1	----	$\text{CYYU}(\text{JTOP}) \left(\frac{\Gamma_{\text{ff}}}{4} + \phi_{\text{JTOP}+1,i} \right)$
3(K < 0)	JMAX-1	----	$\text{CYYU}(\text{JTOP}) \cdot \phi_{\text{JTOP}+1,i}$
4(K > 0)	JMAX	$\frac{-\text{CYYU}(\text{JTOP})}{\text{FH}}$	$\text{CYYU}(\text{JTOP}) \left[\frac{1}{\text{FH}} \left(\frac{\Gamma_{\text{ff}}}{4} + \phi_{\text{JTOP},i} \right) + \phi_{\text{JTOP},1,i} \right]$
4(K < 0)	JMAX	$\frac{-\text{CYYU}(\text{JTOP})}{\text{FH}}$	$\text{CYYU}(\text{JTOP}) \left[\frac{1}{\text{FH}} \phi_{\text{JTOP},i} + \phi_{\text{JTOP}+1,i} \right]$
5(POR > 1.5)	JMAX-1	----	$\text{CYYU}(\text{JTOP}) \left(\phi_{\text{JMAX},i}^{(n-1)} - \phi_{\text{JTOP}+1,i}^{(n)} \right)$ $(\phi_{\text{JMAX},i} \text{ from eqs. (A), (B)})$
5(POR < 1.5)	JMAX	$\frac{-\text{POR} \cdot \text{CYYU}(\text{JTOP})}{x_i - x_{i-1}}$	$\frac{\text{POR} \cdot \text{CYYU}(\text{JTOP}) (\phi_{\text{JMAX},i} - \phi_{\text{JMAX},i-1})}{x_i - x_{i-1}} + \text{CYYU}(\text{JTOP}) \cdot \phi_{\text{JTOP}+1,i}$

and for the bottom boundary

BCTYPE	JBOT	DIAG MODIFICATION	RHS MODIFICATION
1 (K > 0)	JMIN+1	----	----
1 (K < 0)	JMIN	$\frac{-\sqrt{-K} \cdot \text{CYYD}(\text{JBOT})}{x_i - x_{i-1}}$	$\frac{-\sqrt{-K} \cdot \text{CYYD}(\text{JBOT}) \cdot \phi_{\text{JMIN},i} - \phi_{\text{JMIN}-1,i}}{x_i - x_{i-1}}$ $+ \text{CYYD}(\text{JBOT}) \cdot \phi_{\text{JBOT}-1,i}$
2	JMIN	----	----
3 (K > 0)	JMIN+1	----	$\text{CYYD}(\text{JBOT}) \left(\frac{3\Gamma_{ff}}{4} + \phi_{\text{JBOT}-1,i} \right)$
3 (K < 0)	JMIN+1	----	$\text{CYYD}(\text{JBOT}) \cdot \phi_{\text{JBOT}-1,i}$
4 (K > 0)	JMIN	$\frac{-\text{CYYD}(\text{JBOT})}{\text{FH}}$	$\text{CYYD}(\text{JBOT}) \left[\frac{1}{\text{FH}} \left(\frac{3\Gamma_{ff}}{4} + \phi_{\text{JBOT},i} \right) + \phi_{\text{JBOT}-1,i} \right]$
4 (K < 0)	JMIN	$\frac{-\text{CYYD}(\text{JBOT})}{\text{FH}}$	$\text{CYYD}(\text{JBOT}) \left[\frac{1}{\text{FH}} \phi_{\text{JBOT},i} + \phi_{\text{JBOT}-1,i} \right]$
5 (POR > 1.5)	JMIN+1	----	$\text{CYYD}(\text{JBOT}) \left(\phi_{\text{JMIN},i}^{(n-1)} - \phi_{\text{JBOT}-1,i}^{(n)} \right)$ $(\phi_{\text{JMIN},i} \text{ from eqs. (A), (B)})$
5 (POR < 1.5)	JMIN	$\frac{-\text{POR} \cdot \text{CYYD}(\text{JBOT})}{x_i - x_{i-1}}$	$\frac{\text{POR} \cdot \text{CYYD}(\text{JBOT}) (\phi_{\text{JMIN},i} - \phi_{\text{JMIN},i-1})}{x_i - x_{i-1}}$ $+ \text{CYYD}(\text{JBOT}) \cdot \phi_{\text{JBOT}-1,i}$

where for BCTYPE = 5, superscript n refers to the iteration count,

$$\phi_{j,i}^{(n)} = \phi_{j,i-1}^{(n)} + \frac{x_i - x_{i-1}}{POR} (\phi_y^{(n)})_{j,i-1/2} \quad (A)$$

and

$$(\phi_y^{(n)})_{JMAX, JMIN, i-1/2} = \frac{1}{2} \left\{ \frac{\phi_{JMAX, i}^{(n)} - \phi_{JMAX+1, i}^{(n)}}{Y_{JMAX} - Y_{JMAX+1}} + \frac{\phi_{JMAX, i-1}^{(n)} - \phi_{JMAX+1, i-1}^{(n)}}{Y_{JMAX} - Y_{JMAX+1}} \right\} \quad (B)$$

where the (-) sign in equation (A) corresponds to the top (J = JTOP) boundary. This modification applies only for i = IUP, at which point all of the $\phi_{JMAX, i}^{(n)}$ and $\phi_{JMIN, i}^{(n)}$ values are set for the current iteration x sweep. No modification is necessary for i > IUP.

For BCTYPE = 6, the difference equations have not yet been worked out. Consequently, the user must insert the appropriate information required; otherwise, an error message is printed and an abnormal stop occurs.

5.2.5 Subroutine BLOCK DATA.- Subroutine BLOCK DATA is a storage subroutine which contains all of the default values for the various parameters and arrays required as input to the program.

5.2.6 Subroutine BODY.- Subroutine BODY determines the body geometry information required for the boundary conditions and output. Four user-option choices of body description are programmed and available according to the input control BCFOIL as follows:

<u>BCFOIL</u>	<u>AIRFOIL</u>
1	NACA OOX series
2	Parabolic-arc
3	User-supplied ordinates (default = Korn)
4	User specification

The subroutine computes the body ordinates and slopes at the input x-mesh locations and normalizes them by the thickness ratio δ . For BCFOIL = 3, a cubic spline interpolation is used for both slopes and ordinates. Next, the body volume per unit span, camber and thickness distributions are computed, then the airfoil upper and lower airfoil slopes are recomputed using a Reigel's rule nose correction RIGF according to

$$FXU(I) = \frac{FXU(I)}{\sqrt{1 + RIGF(\delta \cdot FXU(I))^2}}$$

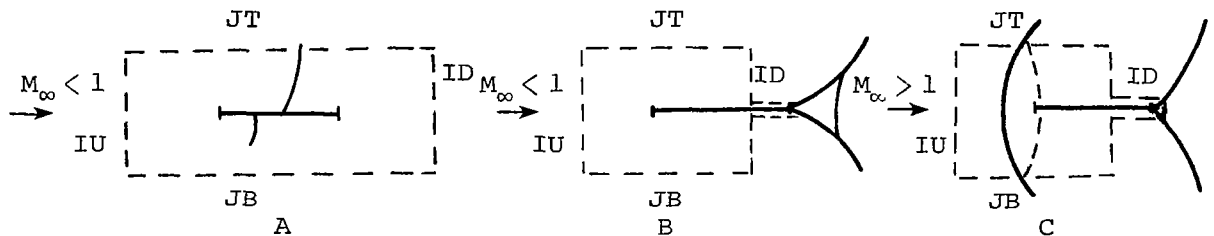
$$FXL(I) = \frac{FXL(I)}{\sqrt{1 + RIGF(\delta \cdot FXL(I))^2}}$$

where $0 \leq RIGF \leq 1$ and is user-specified. Finally, subroutine PRBODY is called to output the calculated information.

5.2.7 Subroutine CDCOLE.- Subroutine CDCOLE computes the airfoil inviscid drag by the momentum integral method; that is, by integrating around a contour enclosing the body and along all shocks inside the contour according to the following formulation:

$$\frac{C_D}{2\delta^{5/3}M_\infty^{-n}} = \oint_C \left[\left(K \frac{u^2}{2} - \frac{v^2}{2} - \frac{\gamma+1}{3} u^3 \right) d\tilde{y} - (uv) dx \right] - \frac{\gamma+1}{12} \int_{S \cap C} [u]^3 d\tilde{y}$$

where \tilde{y} is the transonically scaled y coordinate, n depends on the transonic scaling used (see Table 4-I), and $[u]$ is the jump in ϕ_x across the shock waves S contained within the contour C . Sample contours incorporated in the program are shown below:



where for the subsonic contours A and B, the upstream boundary is set at $IU = (IMIN + ILE)/2$ and the top and bottom boundaries at $JT = JMAX - 1$ and $JB = JMIN + 1$, respectively. For situations where the velocity at the upper surface of the trailing edge is less than sonic, as in contour A, the downstream boundary is set at $ID = (IMAX + ITE)/2$. Otherwise, ID is the I index at the x-mesh point closest to the three-quarter chord point on the airfoil, $X(I) \leq 0.75$, as shown in contours B and C. For supersonic free streams, as in contour C, the upper and lower boundaries are set to encompass the subsonic region ahead of the airfoil. We note that the reason for indenting the contours B and C in the manner shown is to avoid placing the drag contour behind an oblique supersonic shock;

that is, a shock whose downstream flow is supersonic, since the present method uses a first-order accurate hyperbolic operator which tends to dissipate the shock structure over six to ten mesh points. This is an inherent weakness in the present method if sharp supersonic oblique shocks are required.

Subroutine CDCOLE first fixes the locations of the contour boundaries according to the above criteria. In the case of a supersonic free stream, before the calculations are initiated, a check is made to insure that the bow shock is not attached or located too close to the body nose or too close to the upstream boundary. If it is, a message is printed to that effect, the drag computed by a surface pressure integration, and the subroutine exited. If not, the subroutine proceeds to calculate the contour drag integral on the upstream, top, bottom, downstream, and body boundaries, and also on the shocks within the contour using in all cases a trapezoidal rule integration. After the computations are completed, information is printed regarding the drag contribution from each boundary, the total contour drag, the number of shock waves within the contour, the total wave drag, and a message stating whether all the shocks are contained within the contour or whether one or more of the shocks extend beyond the boundaries.

5.2.8 Subroutine CKMESH.- Subroutine CKMESH checks the number of points in both the input x-mesh and the y-mesh. If necessary, the x-mesh is adjusted to contain an odd number of points before the tail and an odd number after the tail, and the y-mesh adjusted to contain an even number of points above and below the airfoil. In counting points in the x-mesh, the point at the trailing edge ($I = I_{TE}$) is included in both the sets before and after the tail. If the addition of the extra point(s) causes either I_{MAX} or J_{MAX} to exceed 100, a message to that effect is printed, no points are added, and the subroutine exited.

5.2.9 Subroutine CPLOT.- Subroutine CPLOT produces a printer plot of the pressure coefficient on the dividing streamline ($y = 0_+$) ahead of, on, and behind the airfoil. The character "U" is printed for the pressure coefficient on the airfoil upper surface, "L" for the lower surface, "B" for points before and behind the airfoil, and "---" for the critical (sonic) pressure coefficient.

5.2.10 Subroutine CUTOUT.- Subroutine CUTOUT reduces the number of input mesh points to establish a coarse grid for the first attempt at a solution. The x-mesh and the y-mesh are halved, and if possible halved

again. If it is desired to use the input mesh to convergence or if the input mesh cannot be refined, the XIN and YIN arrays are loaded into the working arrays X and Y and the routine exited. Otherwise, CUTOUT eliminates half the points in both the x-mesh and y-mesh and adjusts the indices IMAX and JMAX, keeping the first and last points of the meshes at the same location as they were originally.

If it is desired to perform only one grid halving (ICUT = 1), the subroutine loads the remaining points into the X- and Y-arrays and exits. If two mesh cuts are desired (ICUT = 2) and if after the first cut there are an odd number of points before and after the airfoil and an even number below and above the airfoil, the mesh is again halved and the points are loaded into the X- and Y-arrays. After each halving, subroutines ISLIT and JSLIT are called to adjust the I indices at the leading and trailing edges and the J indices immediately above and below the airfoil.

5.2.11 Subroutine DIFCOE.- Subroutine DIFCOE computes on the current (X,Y) mesh the difference coefficients throughout the flow field, including the boundaries, that are required for determining the various finite-difference forms of ϕ_x , ϕ_{xx} , and ϕ_{yy} needed for the solution of the partial differential equation for ϕ . In addition, the difference coefficients required to determine the velocities (ϕ_x, ϕ_y), together with special coefficients needed for extrapolation of flow field properties to the airfoil surface, and special difference coefficients for determining ϕ_{yy} near the airfoil surface are also computed and stored.

5.2.12 Function subroutine DRAG(CDFACT).- Function subroutine DRAG computes the pressure drag by integrating $\phi_x \cdot \phi_y$ around the airfoil surface. The pressure drag is defined as

$$\frac{C_D}{2\delta^{5/3}M_\infty^{-n}} = - \oint_{\text{airfoil}} \phi_x \phi_y \tilde{y} dx$$

or equivalently,

$$\frac{C_D}{2 \cdot \text{CDFACT}} = - \int_0^1 \left[\phi_x^+ \frac{dF_u}{dx} - \phi_x^- \frac{dF_l}{dx} \right] dx$$

where \tilde{y} is the transonically scaled y-coordinate, CDFACT ($=\delta^{5/3}M_\infty^{-n}$) is the appropriate transonic scaling factor for the drag coefficient, and

$F_{u,l}$ are the affine ordinates describing the airfoil upper and lower surface. The subroutine uses the latter form and computes the integral by a trapezoidal rule integration through subroutine TRAP.

5.2.13 Subroutine DROOTS.- Subroutine DROOTS computes the constants ALPHA0, ALPHA1, ALPHA2, OMEGA0, OMEGA1, and OMEGA2 which are required for the subsonic far-field doublet in an ideal slotted wind tunnel (BCTYPE = 4). The ALPHA's are computed in an iterative fashion as follows:

$$\text{ALPHA0}^n = \frac{\pi}{2} - \tan^{-1} \left[F \cdot \text{ALPHA0}^{n-1} - \frac{\sqrt{K}}{\tilde{P}} \right]$$

$$\text{ALPHA1}^n = \frac{\pi}{2} - \tan^{-1} \left[F \cdot (\text{ALPHA1}^{n-1} - \pi) - \frac{\sqrt{K}}{\tilde{P}} \right]$$

$$\text{ALPHA2}^n = \frac{\pi}{2} - \tan^{-1} \left[F \cdot (\text{ALPHA2}^{n-1} - 2\pi) - \frac{\sqrt{K}}{\tilde{P}} \right]$$

where F is the tunnel slot parameter, \tilde{P} is the transonically scaled wall porosity factor, and K is the transonic similarity parameter. One hundred iterations are allowed for convergence, with the error criterion,

$$|\text{ALPHA}^n - \text{ALPHA}^{n-1}| < 10^{-5}$$

If any ALPHA fails to converge within 100 iterations, an error message is printed and the program stops. If all three alphas converge, then the OMEGA's are computed as follows:

$$\text{OMEGA0} = \left(1 + \frac{F}{1 + \text{ctn}^2(\text{ALPHA0})} \right)^{-1}$$

with analogous formula for OMEGA1 and OMEGA2.

5.2.14 Subroutine ECHINP.- Subroutine ECHINP reads and prints all of the input cards that are used for the runs to follow.

5.2.15 Function subroutine EMACH1(U).- Function subroutine EMACH1 computes either the local Mach number or the transonic similarity parameter based on local velocity. If the input logical variable $\text{PHYS} = .T.$, the local Mach number is calculated and set equal to EMACH1 according to $\text{EMACH1}(U) = [1 - (AK - (\gamma + 1)U) \delta^{2/3} M^{2m}]^{1/2}$ where m depends on the transonic scaling used (see Table 4-I). For $\text{PHYS} = .F.$, the local transonic

similarity parameter is computed and set equal to EMACH1; that is, EMACH1(U) = AK - (γ + 1)U.

5.2.16 Subroutine EXTRAP.- Subroutine EXTRAP provides initial values of φ for subsonic free-stream flows to subroutine GUESSP at mesh points which lie outside the range of the previous mesh. The formulation is based on the subsonic far-field solution for free air and for various tunnel simulation boundaries (free jet, solid wall, ideal slotted wall, ideal porous/perforated wall). For free air the formulation used is

$$\phi(x, \tilde{y}) = -\frac{\Gamma_{\text{fff}}}{2\pi} \cdot \tan^{-1} \left(\frac{\sqrt{K\tilde{y}}}{x} \right) + \frac{D}{2\pi\sqrt{K}} \frac{x - \frac{1}{2}}{\left(x - \frac{1}{2}\right)^2 + K\tilde{y}^2}$$

and for tunnel wall boundaries

$$\begin{aligned} \phi(x, \tilde{y})_{x \rightarrow +\infty} &= -\frac{\Gamma_{\text{fff}}}{2} \left[1 - \text{sgn}(\eta) + (1 - J)\psi_0 \frac{\sin(\eta\beta_0)}{\beta_0} e^{-\beta_0\xi} \right] \\ &+ \frac{D}{2K\tilde{H}} \left[B + \Omega_0 \cos(\alpha_0\eta) e^{-\alpha_0\xi} \right] \\ \phi(x, \tilde{y})_{x \rightarrow -\infty} &= -\frac{\Gamma_{\text{fff}}}{2} \left[1 - \frac{J\tilde{y}}{H(1+F)} - \psi_2 \frac{\sin[\eta(\pi - \beta_2)]}{\pi - \beta_2} e^{(\pi - \beta_2)\xi} \right] \\ &+ \frac{D}{2K\tilde{H}} \left[(1 - B)\Omega_1 \cos[(\pi - \alpha_1)\eta] e^{(\pi - \alpha_1)\xi} \right] \end{aligned}$$

where

$$\xi = \frac{x - \frac{1}{2}}{H \cdot \sqrt{K}}$$

$$\eta = \frac{\tilde{y}}{H}$$

and the other parameters are defined according to the following table.

	B	α_0	α_1	Ω_0	Ω_1	Ω_2	J	β_0	β_1	ψ_0	ψ_2
Solid Wall	$\frac{1}{2}$	π	π	1	1	1	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	1	1
Free Jet	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	1	1	1	$\frac{1}{2}$	0	0	1	1
Perforated Wall	0	θ_2	θ_2	1	1	1	0	θ_1	θ_1	1	1
Slotted Wall	0	*	*	*	*	*	$\frac{1}{2}$	*	*	*	*

* No simplification possible for ideal slotted wall--must use formulas below.

$$\cot \alpha_0 = F\alpha_0 - \theta \quad 0 \leq \alpha_0 \leq \pi$$

$$\cot \alpha_1 = F(\alpha_1 - \pi) - \theta \quad 0 \leq \alpha_1 \leq \pi$$

$$\Omega_0 = \left(1 + \frac{F}{1 + \cot^2 \alpha_1} \right)^{-1}$$

$$\Omega_1 = \left(1 + \frac{F}{1 + \cot^2 \alpha_1} \right)^{-1}$$

$$\tan \beta_0 = -F\beta_0 + \theta \quad 0 \leq \beta_0 \leq \frac{\pi}{2}$$

$$\tan \beta_2 = -F(\beta_2 - \pi) + \theta \quad -\frac{\pi}{2} \leq \beta_2 \leq \frac{\pi}{2}$$

$$\psi_0 = \left(1 + \frac{F}{1 + \tan^2 \beta_2} \right)^{-1}$$

$$\psi_2 = \left(1 + \frac{F}{1 + \tan^2 \beta_0} \right)^{-1}$$

$$\theta = \frac{\sqrt{K}}{\tilde{P}}$$

$$\theta_1 = \tan^{-1}(\theta)$$

$$\theta_2 = \cot^{-1}(\theta)$$

5.2.17 Subroutine FARFLD.- Subroutine FARFLD computes the appropriate boundary data for the outer boundaries of the computational mesh. In the case of a supersonic free stream ($K \leq 0.0$), the upstream boundary conditions correspond to uniform undisturbed flow, the downstream boundary is required to be supersonic, and for the top and bottom boundaries the simple wave solution is employed. In this case, the subroutine simply defines the variable $FHINV = 1/F*H$ and returns.

For a subsonic free stream, the functional form of the potential on the outer boundaries is prescribed and calculated. These forms represent the asymptotic far-field behavior of the potential solution, and are given by a concentrated unit strength doublet and vortex in free air or various wind-tunnel environments, with the doublet and vortex located at $x = 0.5$, $y = 0$. For an ideal slotted wind tunnel ($BCTYPE = 4$), subroutines DROOTS and VROOTS are called to compute the constants needed in the doublet and vortex solutions; otherwise, the various constants are calculated within the subroutine. We note that the actual boundary values for ϕ are set in subroutines RECIRC and REDUB, where the functional forms determined here are multiplied by the appropriate vortex and doublet strengths.

5.2.18 Subroutine FINDSK(ISTART,IEND,J,ISK).- Subroutine FINDSK identifies the presence of a shock wave on a specified row (J) of the computation mesh between x -points ISTART and IEND. Pairs of ϕ_x values are checked along the row until the condition

$$PX(ISK,J) \leq SONVEL \leq PX(ISK-1,J)$$

is satisfied, where SONVEL is the sonic point streamwise perturbation velocity. If such a pair is found, the subroutine is immediately exited with that value of ISK. If no shock is found, ISK is set equal to -IEND and the subroutine exited.

5.2.19 Subroutine FIXPLT.- Subroutine FIXPLT uses the airfoil surface pressure distribution arrays CPL and CPU to set up the arrays XP, CPUP, CPLO, and CPS for use in subroutine CPLOT.

5.2.20 Subroutine GUESSP.- Subroutine GUESSP initializes the small-disturbance potential ϕ array as follows:

PSTART = 1, ϕ is set to zero

PSTART = 2, ϕ , X, and Y arrays are read from unit 7 in subroutine READIN together with other flow information about the old solution

PSTART = 3, the arrays from the previous case are already in ϕ , XOLD, YOLD

If PSTART = 2 or 3, the old ϕ array on the XOLD, YOLD mesh must be interpolated onto the current X, Y mesh. A check is made to see if the XOLD mesh is the same as the XIN mesh. If XIN and XOLD are the same mesh, the ϕ array may be interpolated by simple deletion of values at mesh points which have been deleted in subroutine CUTOUT. If the old and new X meshes are not the same, then a simple linear interpolation is made from the old ϕ 's providing that each new x-point lies within the range of the old X mesh. If a new x-point lies outside the old X mesh, then for a supersonic free stream ϕ is set to zero, while for a subsonic free stream subroutine EXTRAP is called to extrapolate a new ϕ value using the far-field solution.

The same steps are followed to interpolate ϕ in the y-direction with the following slight difference. If the new y-point lies outside the range of the old Y mesh, then ϕ is set equal to either $\phi_{JMINO,i}$ if the new y value is less than the minimum old y value, or to $\phi_{JMAXO,i}$ if the new y value is greater than the maximum old y value. Only in the case of subsonic free air flow ($K > 0.$, BCTYPE = 1) is subroutine EXTRAP called to extrapolate for ϕ using the far-field formula.

5.2.21 Subroutine INPERR(I).- Subroutine INPERR writes the appropriate error message associated with an error in input and stops the program. There are eight input error messages:

<u>Error Number</u>	<u>Message</u>
1	IMAX or JMAX is greater than 100, not allowed
2	X mesh points not monotonic increasing
3	Y mesh points not monotonic increasing
4	Mach number not in permitted range (0.5,2.0)

<u>Error Number</u>	<u>Message</u>
5	Alpha not in permitted range (-9.0,9.0)
6	Delta not in permitted range (0.0,1.0)
7	AK = 0. Value of AK must be input since PHYS = .F.
8	Mach number too close to 1, not allowed

5.2.22 Subroutine ISLIT(X).- For a given x-mesh array X, subroutine ISLIT computes the I indices associated with the mesh points on the airfoil surface ($0. \leq x \leq 1.$) at or just behind the leading edge (I = ILE) and at or just ahead of the trailing edge (I = ITE).

5.2.23 Subroutine JSLIT(Y).- For a given y-mesh array Y, subroutine JSLIT computes the J indices associated with the rows of mesh points immediately below (J = JLOW) and above (J = JUP) the airfoil ($y = 0.$).

5.2.24 Function LIFT (CLFACT).- The real function LIFT computes the lift coefficient from the jump in potential $\Delta\phi$ at the trailing edge. The lift coefficient is defined as

$$C_L = - \int_0^1 (C_{p_u} - C_{p_l}) dx = +2\delta^{2/3} M_\infty^{-n} \int_0^1 [(\phi_x)_u - (\phi_x)_l] dx$$

or, equivalent, from an integration by parts

$$C_L = 2 \cdot CLFACT \cdot \Delta\phi_{te}$$

where CLFACT ($=\delta^{2/3} M_\infty^{-n}$) is the appropriate transonic scaling factor for the lift coefficient.

5.2.25 Subroutine MACHMP.- Subroutine MACHMP produces a print map of the local Mach numbers, rounded to the nearest 0.1, throughout the flow field. The row of points immediately above the airfoil are designated by (+), the row immediately below by (-), the leading edge by (L), and the trailing edge by (T). Supersonic points are designated by an asterisk (*).

5.2.26 Subroutine MILINE.- Subroutine MILINE determines the coordinates where sonic velocity occurs and outputs them. Beginning with row J = JMAX, pairs of ϕ_x values are checked along the row to determine whether either of the two conditions

$$PX(I,J) \leq SONVEL \leq PX(I-1,J)$$

$$PX(I,J) \geq SONVEL \geq PX(I-1,J)$$

is satisfied, where SONVEL is the streamwise perturbation velocity at a sonic point. If either of these conditions are met, a linear interpolation is made between $X(I,J)$ and $X(I-1,J)$, that computed x-coordinate stored temporarily in array XSONIC and permanently in array XSLPRT, and the y-coordinate loaded into array YSLPRT. After each row is scanned, the XSONIC array and the J index of the row are printed. When the row immediately under the airfoil is reached, the heading "BODY LOCATION" is printed if at least one sonic point has been found.

If more than 200 sonic points are found, a message to this effect is printed and the subroutine is exited. After the entire grid has been checked for sonic points and if at least one has been found, the subroutine checks whether any of the sonic points lie on the outer boundaries. If one does and if the case under study is either a subsonic free-stream calculation or a free air (BCTYPE = 1) supersonic free-stream calculation, the following message is printed out:

* * * * * CAUTION * * * * *

SONIC LINE HAS REACHED A BOUNDARY
THIS VIOLATES ASSUMPTIONS USED TO DERIVE BOUNDARY CONDITIONS
SOLUTION IS PROBABLY INVALID

Finally, the subroutine defines the boundaries and step increments for the printer plot of the sonic line and passes these constants along with the arrays XSLPRT and YSLPRT and the number NPTS of sonic points into a call to the subroutine PLTSON, which plots the sonic line.

5.2.27 Subroutine NEWISK (ISKOLD,J,ISKNEW).- Subroutine NEWISK locates an updated x-position ($I = ISKNEW$) of the shock wave on a specified row J given an initial guess for the location ($I = ISKOLD$). The subroutine searches the range $ISKOLD - 3 \leq I \leq ISKOLD + 2$ testing the condition $PX(ISKNEW - 1, J) \geq SONVEL \geq PX(ISKNEW, J)$. If the condition is met, the subroutine returns with the current value of ISKNEW. Otherwise, with no shock being found, ISKNEW is set negative and the subroutine exited.

5.2.28 Function PITCH (CMFACT).- Function PITCH computes the airfoil pitching moment (positive nose-up) about $x = x_m$, $y = 0$. Pitching moment is defined as

$$C_m = \int_0^1 (x - x_m) (C_{P_u} - C_{P_l}) dx$$

or, equivalently

$$C_m = -2 \cdot \text{CMFACT} \cdot (1 - x_m) \cdot \Delta\phi_{te} - \int_0^1 \Delta\phi dx$$

where $\text{CMFACT} (= \delta^{2/3} M_\infty^{-n})$ is the appropriate transonic scaling for the moment coefficient. The subroutine uses the latter form for C_m and computes the second integral by trapezoidal rule integration through subroutine TRAP. We note that the subroutine sets $x_m = 0.25$, so that the pitching moment is calculated about the quarter chord.

5.2.29 Subroutine PLTSON (X,Y,XAXMIN,XMAX,XINCR,YAXMIN,YMAX,YINCR,NPTS).- Subroutine PLTSON is a modified version of a printer plot routine developed by M. S. Itzkowitz, May 1967. The subroutine provides a printer plot of the sonic lines that are present in the flow field. The plot is provided on a 51×101 point rectangular grid with $-1.0 \leq y \leq 1.5$ and $0.75 \leq x \leq 1.75$, and uses 56 printer lines. The argument list is all input and is defined as follows:

X	array of x-ordinates to be plotted
Y	array of y-ordinates to be plotted
XAXMIN	minimum x (left) grid boundary
XMAX	maximum x (right) grid boundary
XINCR	increment between x-grid marks
YAXMIN	minimum y (lower) grid boundary
YMAX	maximum y (upper) grid boundary
YINCR	increment between y-grid marks
NPTS	dimension of (X,Y) arrays

If either of the incremental step sizes is zero, the program will exit with no plot produced. The input arrays are not destroyed during the calculation.

5.2.30 Subroutine PRBODY.- Subroutine PRBODY prints out the airfoil geometrical characteristics. If logical variable $\text{PHYS} = .T.$, all dimensions are normalized by the airfoil chord; otherwise, all dimensions except x are normalized by chord length times thickness ratio. Quantities printed are maximum thickness, volume of airfoil per unit span,

maximum camber, and for each input x-mesh point along the airfoil, the y-ordinates of the upper and lower surfaces of the airfoil, the thickness distribution, and camberline.

5.2.31 Subroutine PRINT.- Subroutine PRINT is the main print control of program output. The subroutine prints header information regarding:

- (i) choice of printout in similarity on physical variables
- (ii) definition of transonic scaling (Cole, Spreiter, Krupp, or user)
- (iii) boundary condition type (free air, free jet, solid wall, slotted wall, or porous wall)
- (iv) choice of differencing (fully conservative or not conservative) at shock
- (v) choice of Kutta condition or lift specification

and also outputs the Mach number, thickness ratio, angle of attack, transonic similarity parameter, far-field doublet strength, airfoil volume (per unit span), and the values of the transonic scaling parameters used for scaling the coefficients of pressure, lift, drag, and moment, the y-coordinate, and the vertical velocity. The rest of the print output is performed by calling the following subroutines: PRINT1, FIXPLT, PRTWAL, MILINE, PRTFLD, and CDCOLE.

5.2.32 Subroutine PRINT1.- Subroutine PRINT1 outputs the pressure coefficient and local Mach number (or local similarity parameter if $PHYS = .F.$) on the airfoil and dividing streamline ($y = 0.±$) at each x-point of the current mesh, and also provides a printer plot of the pressure coefficient along side the tabulated values. In addition, the y-coordinate array Y is printed as well as the lift, moment, and sonic pressure coefficients. Since PRINT1 is used to putput these quantities on each grid (coarse, medium, and fine), a header is written to identify the current grid. If the local Mach number exceeds 1.3, a warning message is printed indicating that the predicted shock jumps may be in error. Also, if a detached bow shock wave is between IMIN and IMIN + 1, a message is printed indicating that the detached shock will lie upstream of the current x mesh, the ABORT parameter set equal to true and the calculation terminated.

5.2.33 Subroutine PRTFLD.- Subroutine PRTFLD outputs the pressure coefficient, flow angle, and Mach number in the flow field. The number of rows (J lines) printed is set by the user-input value of PRTFLO as follows:

PRTFLO = 1, no J lines are printed
PRTFLO = 2, all J lines are printed
PRTFLO = 3, three J lines around the row where the maximum iteration error in ϕ occurs (i.e., J = JERROR) are printed

5.2.34 Subroutine PRTMC.- Subroutine PRTMC produces a local flow character map of the flow field by ascribing to each point in the input mesh a letter designating whether the flow at that point is elliptic (subsonic), parabolic (sonic), hyperbolic (supersonic), or shock (shock point) according to the convention:

- Elliptic (Subsonic)
P Parabolic (Sonic)
H Hyperbolic (Supersonic)
S Shock Point (Shock)

5.2.35 Subroutine PRTSK (Z,ARG,L,NSHOCK,CDSK,LPRTL).- Subroutine PRTSK provides an inviscid wake profile for all of the shock waves contained within the momentum contour used to calculate the drag. The wave drag contribution $CD(y)$ and total pressure loss $P_0(y)/P_\infty$ along each shock wave is given as a function of the transonically scaled y-coordinate. In addition, the total wave drag for each shock is also provided. If a shock wave extends outside the contour, a message to that effect is printed indicating that shock losses are not available for that portion of the shock.

5.2.36 Subroutine PRTWAL.- Subroutine PRTWAL determines the pressure coefficient and the flow angle on the wind-tunnel walls ($y = \pm H$) according to the particular type of wall, prints that information, and also provides a printer plot of the pressure distribution along side the tabulated printout. Information regarding the wall type, tunnel half height, wall porosity factor (if applicable), tunnel slot parameter (if applicable), and critical pressure coefficient are also provided.

5.2.37 Function PX(I,J).- Function PX computes the streamwise perturbation velocity ϕ_x at the point (I,J). Different differencing forms are used depending whether the point is on the upstream or downstream boundaries

or is an interior point. For the interior points, ϕ_x is given by

$$\phi_x = \frac{1}{2} \left[\frac{\phi_{j,i+1} - \phi_{j,i}}{x_{i+1} - x_i} + \frac{\phi_{j,i} - \phi_{j,i-1}}{x_i - x_{i-1}} \right]$$

while on the upstream boundary ($I = \text{IMIN}$)

$$\phi_x = \frac{3}{2} \left[\frac{\phi_{j,i+1} - \phi_{j,i}}{x_{i+1} - x_i} \right] - \frac{1}{2} \left[\frac{\phi_{j,i+2} - \phi_{j,i+1}}{x_{i+2} - x_{i+1}} \right]$$

and on the downstream boundary ($I = \text{IMAX}$)

$$\phi_x = \frac{3}{2} \left[\frac{\phi_{j,i} - \phi_{j,i-1}}{x_i - x_{i-1}} \right] - \frac{1}{2} \left[\frac{\phi_{j,i-1} - \phi_{j,i-2}}{x_{i-1} - x_{i-2}} \right]$$

5.2.38 Function PY (I,J).- Function PY computes the vertical perturbation velocity ϕ_y at the point (I,J). Different differencing formulas are employed depending whether the point is on the upper or lower boundaries or on the rows immediately above or below the airfoil. In the latter case, separate formulations are involved depending upon whether the point is ahead, over or under, or behind the airfoil. For all of the other points,

$$\phi_y = \frac{1}{2} \left[\frac{\phi_{j+1,i} - \phi_{j,i}}{Y_{j+1} - Y_j} + \frac{\phi_{j,i} - \phi_{j-1,i}}{Y_j - Y_{j-1}} \right]$$

The exceptional cases are:

Lower Boundary ($j = \text{JBOT}$)

$$\phi_y = \frac{3}{2} \left[\frac{\phi_{j+1,i} - \phi_{j,i}}{Y_{j+1} - Y_j} \right] - \frac{1}{2} \left[\frac{\phi_{j+2,i} - \phi_{j+1,i}}{Y_{j+2} - Y_{j+1}} \right]$$

Upper Boundary ($j = \text{JTOP}$)

$$\phi_y = \frac{3}{2} \left[\frac{\phi_{j,i} - \phi_{j-1,i}}{Y_j - Y_{j-1}} \right] - \frac{1}{2} \left[\frac{\phi_{j-1,i} - \phi_{j-2,i}}{Y_{j-1} - Y_{j-2}} \right]$$

Row below airfoil ($j = \text{JLOW}$)

(a) Ahead

$$\phi_y = \frac{1}{2} \left[\frac{\phi_{\text{JUP},i} - \phi_{\text{JLOW},i}}{Y_{\text{JUP}} - Y_{\text{JLOW}}} + \frac{\phi_{\text{JLOW},i} - \phi_{\text{JLOW}-1,i}}{Y_{\text{JLOW}} - Y_{\text{JLOW}-1}} \right]$$

(b) Under

$$\phi_Y = \frac{1}{2} \left[\left(\frac{dF_\ell}{dx} \right)_i - \alpha + \frac{\phi_{JLOW,i} - \phi_{JLOW-1,i}}{Y_{JLOW} - Y_{JLOW-1}} \right]$$

(c) Behind

$$\phi_Y = \frac{1}{2} \left[\frac{\phi_{JUP,i} - \Delta\phi_i - \phi_{JLOW,i}}{Y_{JUP} - Y_{JLOW}} + \frac{\phi_{JLOW,i} - \phi_{JLOW-1,i}}{Y_{JLOW} - Y_{JLOW-1}} \right]$$

Row above airfoil ($j = JUP$)

(a) Ahead

$$\phi_Y = \frac{1}{2} \left[\frac{\phi_{JUP,i} - \phi_{JLOW,i}}{Y_{JUP} - Y_{JLOW}} + \frac{\phi_{JUP+1,i} - \phi_{JUP,i}}{Y_{JUP+1} - Y_{JUP}} \right]$$

(b) Over

$$\phi_Y = \frac{1}{2} \left[\left(\frac{dF_u}{dx} \right)_i - \alpha + \frac{\phi_{JUP+1,i} - \phi_{JUP,i}}{Y_{JUP+1} - Y_{JUP}} \right]$$

(c) Behind

$$\phi_Y = \frac{1}{2} \left[\frac{\phi_{JUP,i} - \Delta\phi_i - \phi_{JLOW,i}}{Y_{JUP} - Y_{JLOW}} + \frac{\phi_{JUP+1,i} - \phi_{JUP,i}}{Y_{JUP+1} - Y_{JUP}} \right]$$

5.2.39 Subroutine READIN.- All user-option input to the program is read in subroutine READIN. The subroutine first calculates and prints the elapsed time in seconds that the previous case took to run, or for the first case sets the start time. Next, a check is made to determine if any test cases are left. If there are and if the user has opted to use the previous result for ϕ as a first guess for the current case (PSTART = 3), then a test is made to determine if the ϕ array in core from the previous case is usable (i.e., if the previous calculation was not aborted). Next, if the default grids for either subsonic free air (XKRUPP,YFREE) or tunnel (XKRUPP,YTUN), or the analytic default mesh (AYMESH = TRUE) are to be used, the subroutine loads those grids into the XIN, YIN arrays and then prints out all of the user-option input parameters, including those having default values. A check is then made to insure that the input mesh does not exceed 100 points in either direction and that the mesh ordinates are monotonic increasing. Next, checks are made on M_∞ , α , and δ to insure they are within the permitted range, and then the indices of the leading

(I = ILE) and trailing (I = ITE) edges and the rows immediately above (J = JUP) and below (J = JLOW) the airfoil are determined. Subroutine CKMESH is then called to insure that certain grid areas have an appropriate (odd or even) number of points. If the case underway is a tunnel simulation calculation, the bounds of the grid are checked and adjusted if necessary. Finally, if the initial guess for the ϕ array is to be read from unit 7 (PSTART = 2), the necessary input from that calculation is read and then printed.

5.2.40 Subroutine RECIRC.- Subroutine RECIRC computes the jump in ϕ at the trailing edge, $\Delta\phi_{te}$, the updated far-field circulation, Γ_{ff}^{n+1} , and the jump in ϕ , $\Delta\phi(x)$, along the slit $y = 0.$, $x > 1.$, according to the following formulas.

For KUTTA = .T.,

$$\Delta\phi_{te} = \phi_{te}^+ - \phi_{te}^- = \frac{\phi_{JUP,ITE} \cdot Y_{JUP+1} - \phi_{JUP+1,ITE} \cdot Y_{JUP}}{Y_{JUP+1} - Y_{JUP}} - \frac{\phi_{JLOW-1,ITE} \cdot Y_{JLOW} - \phi_{JLOW,ITE} \cdot Y_{JLOW-1}}{Y_{JLOW} - Y_{JLOW-1}}$$

$$\Gamma_{ff}^{n+1} = (1. - \omega_{\Gamma}) \cdot \Gamma_{ff}^n + \omega_{\Gamma} \Delta\phi_{te}$$

$$\Delta\phi(x) = \Delta\phi_{te} + \frac{x - 1}{x_{max} - 1} (\Gamma_{ff}^{n+1} - \Delta\phi_{te})$$

where ω_{Γ} is the relaxation factor for circulation, while for KUTTA = .F.,

$$\Delta\phi_{te} = \frac{CLSET}{2}$$

$$\Gamma_{ff} = \frac{CLSET}{2}$$

$$\Delta\phi(x) = \frac{CLSET}{2}$$

where CLSET is the specified lift coefficient.

5.2.41 Subroutine REDUB.- Subroutine REDUB computes the far-field doublet strength. For lifting free air flows (BCTYPE = 1 and ABS(CIRCF) $> 10^{-4}$), the doublet strength is set equal to the model volume. For other flows the nonlinear contribution (DBLSUM = $((\gamma + 1)/4) \cdot \iint \phi_x^2 dx dy$) is added

to the volume. The double integral is computed by considering ϕ_x to be centered midway between mesh points; that is,

$$DBLSUM = \frac{\gamma + 1}{4} \sum_{i=IMIN}^{IMAX} \Delta x_{i+1/2} \int_{Y_{JMIN}}^{Y_{JMAX}} (\phi_x^2)_{i+1/2} dy$$

so that

$$DBLSUM = \frac{\gamma + 1}{4} \sum_{i=IMIN}^{IMAX} \frac{1}{x_{i+1} - x_i} \int_{Y_{JMIN}}^{Y_{JMAX}} (\phi_{j,i+1} - \phi_{j,i})^2 dy$$

and the y-integration is performed using a modified trapezoidal rule.

We note that the doublet strength is required only for $M_\infty < 1$. For subsonic free air lifting flows, the doublet integral DBLSUM is not convergent since $\phi \sim \Gamma_{ff} \tan^{-1}(y/x)$; in this case, a more complicated integral applies (see Krupp, ref. 3). This program neglects that contribution, and while this procedure introduces an error, numerical experimentation has shown it not to be significant.

5.2.42 Subroutine REFINE.- Subroutine REFINE doubles the number of mesh points in both the x- and y-meshes by inserting new mesh points halfway between the old mesh points. The subroutine also linearly interpolates the ϕ array to the new points, determines the new (I,J) indices of the leading (ILE) and trailing (ITE) edges, and the rows immediately above (JUP) and below (JLOW) the airfoil by calls to subroutines ISLIT and JSLIT, respectively, and uses a linear extrapolation to calculate ϕ on the new JLOW and JUP rows.

5.2.43 Subroutine RESET.- Subroutine RESET updates the far-field boundary conditions for subsonic free-stream flows. The potential ϕ is recalculated on the upstream (IMIN) and downstream (IMAX) boundaries for both free air or wind-tunnel simulations. The update is accomplished at the end of each iteration sweep through the flow field. For free air flows (BCTYPE = 1), new boundary values are also calculated on the top (JMAX) and bottom (JMIN) boundaries; while for wind-tunnel simulations, this calculation is omitted.

5.2.44 Subroutine SAVEP.- Subroutine SAVEP automatically stores restart data from the currently completed calculation in an "old data"

block. The subroutine first rescales the parameters back into the input format and then stores the following parameters and arrays: TITLE(I), IMIN, JMIN, IMAX, JMAX, CL, EMACH, ALPHA, DELTA, VOL, DUB, X(I), and YIN(J). If the user has opted to save the restart block (PSAVE = .T.), then the subroutine also writes the above information on Tape 3.

5.2.45 Subroutine SCALE.- Subroutine SCALE scales physical variables that are required for program computation. The option to scale or not is controlled by the logical input variable PHYS. For PHYS = .T., all input and output quantities are in physical units normalized by free-stream values and airfoil chord, and SCALE reduces them to transonic variables according to the following convention:

SIMDEF = 1, Cole scaling
SIMDEF = 2, Spreiter scaling
SIMDEF = 3, Krupp scaling
SIMDEF = 4, User-option scaling; if this option is used,
the definition of local Mach number must also
be adjusted in EMACH1

For PHYS = .F., input is already in scaled variables and no further scaling is done. In this case, the user must input the value of the transonic similarity parameter $AK = (1 - M_\infty^2) / M_\infty^2 m_\infty^{2/3}$ where the exponent m is now user-specified.

Subroutine SCALE then checks to determine that AK has an appropriate value for the case being considered, computes the square root of AK(RTK), the sonic velocity (SONVEL), and the critical pressure coefficient (CPSTAR) and then exits.

5.2.46 Subroutine SETBC.- Subroutine SETBC sets the limits on the range of the I and J indices for solution of the difference equations according to the following schedule:

$M_\infty < 1$				
BCTYPE	IUP	IDOWN	JTOP	JBOT
1	IMIN+1	IMAX-1	JMAX-1	JMIN+1
2	IMIN+1	IMAX-1	JMAX	JMIN
3	IMIN+1	IMAX-1	JMAX-1	JMIN+1
4	IMIN+1	IMAX-1	JMAX	JMIN
5 (POR \leq 1.5)	IMIN+1	IMAX-1	JMAX	JMIN
5 (POR $>$ 1.5)	IMIN+1	IMAX-1	JMAX-1	JMIN+1

$M_\infty > 1$				
BCTYPE	IUP	IDOWN	JTOP	JBOT
1	IMIN+2	IMAX	JMAX	JMIN
2	IMIN+2	IMAX	JMAX	JMIN
3	IMIN+2	IMAX	JMAX-1	JMIN+1
4	IMIN+2	IMAX	JMAX	JMIN
5 (POR \leq 1.5)	IMIN+2	IMAX	JMAX	JMIN
5 (POR $>$ 1.5)	IMIN+2	IMAX	JMAX-1	JMIN+1

The subroutine also computes the body slope boundary conditions at the current x-mesh points on the body, multiplies them by the appropriate mesh spacing constants and stores them into arrays FXUBC(J), FXLBC(I) for use in solving the finite-difference equations at points on the rows immediately above and below the airfoil

5.2.47 Subroutine SIMP (R,X,Y,N,IER).- Subroutine SIMP calculates the integral $\int y dx$ by using Simpson's rule integration. The subroutine argument list is defined as follows:

- R resultant value of the integral
- X vector array of x points
- Y vector array of y values

N length of vector arrays (X,Y)
IER error control index; equal to 1 for satisfactory result;
equal to 2 when only one point is in the interval (N = 1);
equal to 4 when X array not monotonic

Equal point spacing is not required and the number of points N can be either odd or even. In the latter case, a polynomial fit through the first three points is used and an analytical integration performed from X(1) to X(2) in order to obtain an odd number of points for the remaining interval, X(2) to X(N).

5.2.48 Subroutine SOLVE.- Subroutine SOLVE controls the sequence of calculations involved in the finite-difference solution for ϕ . The potential is calculated by using a successive line over-relaxation (SLOR) algorithm and sweeping the flow field from upstream to downstream limits one column at a time. At the start of each sweep, subroutine RECIRC is called to compute the jump in potential at the trailing edge, the circulation for the far-field boundary, and the jump in potential along the slit $y = 0, x > 1$. Next, subroutine SYOR is called to compute a new value of the potential at all the grid points. For subsonic flows, after the program has completed an iteration sweep, and if the current calculation is a free air case, the potential is updated throughout the flow field by adding the increment in potential due to the change in circulation multiplied by the far-field vortex behavior. Then subroutine REDUB is called (but only at every NDUBth iteration sweep) to recompute the far-field doublet strength, and then subroutine RESET is called (at each iteration) to update the far-field boundary conditions. At every IPRTERth iteration, the iteration count (ITERATION), lift coefficient (CL), moment coefficient (CM), I and J locations (IERR,JERR) where the maximum iteration error in ϕ occurs, the absolute magnitude of the maximum iteration error in ϕ (ERROR), the I and J locations (IRL,JRL) of the maximum residual, the absolute magnitude of the maximum residual (BIGRL), the absolute magnitude of the iteration error in circulation (ERCIRC), the I location (ICPU) of the point on the airfoil where the maximum iteration error in upper surface pressure coefficient occurs, the absolute magnitude of the maximum iteration error in upper surface pressure coefficient (CPERRU), and the corresponding results (ICPL,CPERRL) for the maximum iteration error in surface pressure coefficient. The results are then checked for convergence of potential. If the test is met, the message "SOLUTION CONVERGED" is printed and the subroutine exited. If the convergence

criteria is not met, a divergence test on ϕ is performed. If the maximum iteration error in ϕ has become greater than the divergence control (DVERGE), the error message "SOLUTION DIVERGED" is printed and the calculation aborted. If the solution has neither converged nor diverged, the iteration sweeps are continued up to the limit, MAXIT, at which point the message "ITERATION LIMIT REACHED" is printed and the subroutine exited.

5.2.49 Subroutine SPLN1 (X,Y,N).- Subroutine SPLN1 is a cubic spline interpolation subroutine which determines both the ordinate and slope of a function using a cubic spline fit. The argument list consists of a table of (X,Y) ordinates and is defined as follows:

X array of independent variable
 Y array of dependent variable
 N dimension of (X,Y) arrays

There are two entry points to this subroutine. A call to SPLN1 initializes the cubic spline coefficient arrays, while a call to SPLN1X will determine the spline-fit ordinate (YP) and slope (DYP) at the specified point XP. We note that XP,YP, and DXP are passed through a name common block (SPLN) rather than an argument list.

To use SPLN1 it is necessary to specify a derivative of Y at the end points X(1),X(N). Either the first or second derivative at these points may be given. Control for this option is through the parameters (K1,K2) according to the following:

K1 = 1, DY1 = first derivative of Y at X(1)
 K1 = 2, DY1 = second derivative of Y at X(1)
 K2 = 1, DY2 = first derivative of Y at X(N)
 K2 = 2, DY2 = second derivative of Y at X(N)

Values for (K1,K2,DY1,DY2) must be specified prior to the call to SPLN1.

5.2.50 Subroutine SYOR.- Subroutine SYOR determines the potential ϕ at the interior grid points through the flow field; that is, $JBOT \leq J \leq JTOP$, $IUP \leq I \leq IDOWN$. This is accomplished by solving the tridiagonal matrix equation $A\bar{C}_i = f_i$ where \bar{C}_i represents the correction potential and the N dimensional column vector (with $N = JTOP - JBOT + 1$) defined by

$$\bar{C}_i = \begin{pmatrix} C_{JBOT,i} \\ \vdots \\ C_{JTOP,i} \end{pmatrix}$$

and where A is an $N \times N$ dimensional, diagonally-dominant, tridiagonal matrix, and f_i is an N dimensional column vector.

The following solution sequence is repeated for each x -station from IUP to IDOWN. The coefficient $1/(x_i - x_{i-1})^2$ of the ϕ_{xt} term is calculated. Then the array VC_j representing the coefficient of ϕ_{xx} in the partial differential equation; that is, $(K - (\gamma + 1)\phi_x)/2\Delta x$, is computed at each J station. Next, the arrays representing the diagonal ($DIAG_j$), subdiagonal (SUB_j), and superdiagonal (SUP_j) elements of the tridiagonal matrix A are determined, where account is taken in calculating $DIAG_j$ of whether VC_j is greater or less than zero. The array RHS which is equal to the negative of the residual ($-R_{ij}$), and which represents the right-hand side f_i of the matrix equation is then computed. If the iteration sweep in x has proceeded to the airfoil; that is, $ILE \leq I \leq ITE$, the $DIAG_j$, SUB_j , SUP_j , and RHS_j arrays are modified at the points immediately below ($j = JLOW$) and above ($j = JUP$) the x axis to account for the airfoil boundary condition. A similar procedure is used at ($j = JLOW, JUP$) at locations behind the trailing edge to account for the Kutta condition jump in potential along the cut extending from the trailing edge to the downstream boundary. Next subroutine BCEND is called to modify the diagonal ($DIAG_j$) and right-hand-side (RHS_j) arrays to account for the appropriate boundary conditions at JBOT and JTOP. The current column is then scanned to see whether the maximum residual $|R_{ij}|_{\max} = |RHS_j|$ is contained within it. If it is, that value (BIGRL) and the I, J locations (IRL, JRL) are stored. Finally, modifications are made to $DIAG_j$ and RHS_j to add the $\epsilon \Delta t \phi_{xt}/\Delta x$ term according to

$$DIAG_j = DIAG_j - \frac{\epsilon}{(x_i - x_{i-1})^2}$$

$$RHS_j = RHS_j - \epsilon \frac{(\phi_{j,i-1}^{(n+1)} - \phi_{j,i-1}^{(n)})}{(x_i - x_{i-1})^2}$$

The tridiagonal matrix equation $A \bar{C}_i = f_i$ is then solved by using the method of triangular decomposition. The $N \times N$ dimensional, tridiagonal matrix A is factored into the product $A = LU$, where L is a lower triangular matrix containing only subdiagonal and diagonal elements and U is a special upper triangular matrix containing only superdiagonal elements and unit diagonal elements. Thus, the original equation

$$A \bar{C}_i = L U \bar{C}_i = f_i$$

is factored into two equations

$$U \bar{C}_i = Z$$

$$L Z = f_i$$

The latter equation is solved first for Z and then the former equation is solved for \bar{C}_i by back-substitution.

Finally, the new potential is computed by adding the correction to the old potential, and the correction potential \bar{C}_i vector is scanned to determine whether the maximum correction is contained within it. If it is, that value (ERROR) and the (I,J) location (IERROR, JERROR) are stored for output.

5.2.51 Subroutine TRAP (X,Y,N,SUM).- Subroutine TRAP performs an integration of the vector array Y with respect to the ordinates in the vector array X by using the trapezoidal rule. Here, N is the length of the X, Y arrays and SUM is the resultant integral. The formula used is

$$SUM = \frac{1}{2} \sum_{i=1}^{N-1} (x_{i+1} - x_i) (Y_{i+1} + Y_i)$$

5.2.52 Main program TSFOIL (INPUT, OUTPUT, TAPE5 = INPUT, TAPE6 = OUTPUT, TAPE3, TAPE7).- TSFOIL is the main program. No calculations are performed within this routine, which acts only to call the necessary subroutines to read input, perform the required calculations, and print the output. Two tests are made within the routine that determine: (1) when the final computation mesh has been reached; and (2) whether the current calculation has been aborted. If either of these tests are

positive, the program prints the final output for the current case, resets the mesh parameters back to those for the input mesh, terminates the current calculation, and goes on to read the input for the following case.

5.2.53 Subroutine VROOTS.- Subroutine VROOTS computes the constants BETA0, BETA1, BETA2, PSI0, PSI1, and PSI2 which are required for the subsonic far-field vortex in an ideal slotted wind tunnel (BCTYPE = 4). The BETA's are computed in an iterative fashion as follows:

$$\text{BETA0}^n = \tan^{-1} \left[-F \cdot \text{BETA0}^{n-1} + \frac{\sqrt{K}}{\tilde{P}} \right]$$

$$\text{BETA1}^n = \tan^{-1} \left[-F \cdot (\text{BETA1}^{n-1} + \pi) + \frac{\sqrt{K}}{\tilde{P}} \right]$$

$$\text{BETA2}^n = \tan^{-1} \left[-F \cdot (\text{BETA2}^{n-1} - \pi) + \frac{\sqrt{K}}{\tilde{P}} \right]$$

where F is the tunnel slot parameter, K the transonic similarity parameter, and \tilde{P} the transonically scaled wall porosity factor. One hundred iterations are allowed for convergence with the error criteria

$$|\text{BETA}^n - \text{BETA}^{n-1}| < 10^{-5}$$

If any BETA fails to converge within 100 iterations, an error message is printed and the program stops. If all three BETA's converge, then the PSI's are computed as follows:

$$\text{PSI0} = 1 + \frac{F}{1 + \tan^2(\text{BETA0})}$$

with analogous formulas for PSI1 and PSI2.


```

REAL JET
COMMON /COM16/ ALPHA0 , ALPHA1 , ALPHA2 , XSING ,
1 OMEGA0 , OMEGA1 , OMEGA2 , JET
COMMON /COM33/ THETA(100,100)
SUBROUTINE TO COMPUTE THE ANGLE THETA AT
EACH MESH POINT.
R2PI = 1.0 / TWOPI
DO 20 I=1,MN,IMAX
  X = XIN(I) - XSING
  DO 10 J=1,MN,JMAX
    YY = YIN(J) * RTK
    R = SQRT(YIN(J)**2 + XX**2)
    ATM = ATAN2(YY,XX)
    O = PI - SIGN(PI,YY)
    THETA(J,I) = -(ATN + O) * R2PI
  IF (R .LE. 1.0) THETA(J,I) = THETA(J,I) * R
10 CONTINUE
20 CONTINUE
RETURN
END

```

```

FUNCTION ARF (X)
EVALUATES ERF WITH AN ERROR .LT. 1.5E-7 BY RATIONAL
APPROXIMATION 7.1.26 OF HANDBOOK OF MATH. FUNCTIONS
U. S. DEPT. OF COMMERCE. NBS APPL MATH SER 55.
CALLED BY - AYMESH.
DIMENSION C(5)
DATA C /1.061405429,-1.453152027,1.421413741, -.284496736,
.254829592/
Y = X
IF ( X .LT. 0.0 ) Y = -Y
IF ( Y .LT. 10. ) GO TO 10
ARF = 1.0
GO TO 30
10 CONTINUE
T = 1.0 / (1.0 + .3275911*Y)
POLY = 0.0
DO 20 I=1,5
  POLY = (POLY + C(I)) * T
20 CONTINUE
ARF = 1.0 - POLY * EXP(-Y*Y)
30 CONTINUE
IF (X .LT. 0.0) ARF = -ARF
RETURN
END

```

```

SUBROUTINE AYMESH
COMPUTES ANALYTICAL X AND Y MESH POINTS.
CALLED BY - READIN.
COMMON /COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JTOP , JBOT , JMAX , JUP , JLOW ,
2 LOGICAL AMESH
COMMON /COM6/ XIN(100) , YIN(100) , AMESH
INTEGER PSTART
LOGICAL PSAVE
COMMON /COM11/ ALPHA0 , CLQLO , DELTAD , DUBD , EMACHO ,
1 IMINO , IMAXO , IMAXI , JMINO , JMAXO ,
2 JMAXI , PSAVE , PSTART , TITLE(8) , TITLED(8) ,
3 VOLD , XOLD(100) , YOLD(100)
COMMON /COM12/ F , H , HALFFI , PI , RTKPOR ,
1 INTEGER BCTYPE
COMMON /COM20/ BCTYPE , CIRCFF , PHINV , POR , CRACTE
COMMON /COM30/ BXH(401) , REST(3)
DIMENSION XH(401) , BX(100)

```

```

COM16 2
COM16 3
COM16 4
COM33 2
ANGLE 11
ANGLE 12
ANGLE 13
ANGLE 14
ANGLE 15
ANGLE 16
ANGLE 17
ANGLE 18
ANGLE 19
ANGLE 20
ANGLE 21
ANGLE 22
ANGLE 23
ANGLE 24
ANGLE 25
ANGLE 26
ANGLE 27
ANGLE 28

```

```

ARF 2
ARF 3
ARF 4
ARF 5
ARF 6
ARF 7
ARF 8
ARF 9
ARF 10
ARF 11
ARF 12
ARF 13
ARF 14
ARF 15
ARF 16
ARF 17
ARF 18
ARF 19
ARF 20
ARF 21
ARF 22
ARF 23
ARF 24
ARF 25
ARF 26
ARF 27
ARF 28

```

```

AYMESH 2
AYMESH 3
AYMESH 4
COM1 2
COM1 3
COM1 4
COM4 2
COM4 3
COM11 2
COM11 3
COM11 4
COM11 5
COM11 6
COM11 7
COM12 2
COM12 3
COM20 2
COM20 3
AYMESH 10
AYMESH 11
AYMESH 12

```

```

DATA A0/.225/, A1/1.4/, A2/1.6/, A3/.6188/, A4/.75/, A5/30./,
1 A6/.603/, A7/2.0/
DATA CT1/2.0/, CT2/2.0/, CT3/1.0/, CF1/1.0/, CF2/1.0/, CF3/5.2/
IMAXI=81
JMAXI=64
IMA = (IMAXI - 1) / 2
A22 = A2 * A2
A72 = A7 * A7
FACT = HALFFI * .005
DO 10 I=201,401
  EX = TAN(FACT * (I-201) )
  XH(I) = EX
  EX2 = EX * EX
  BXH(I) = A1 * EX + EXP(-A22*EX2) + (1.0-EXP(-A72*EX2))*ARF(A4*EX)
10 CONTINUE
DO 20 I=1,200
  XH(I) = - XH(402-I)
  BXH(I) = -BXH(402-I)
20 CONTINUE
TWOPI = 2.0 / PI

```

C ADD 2/PI ATAN(AK(X+A6)) TO BXH TO GIVE MORE POINTS NEAR THE LEADING EDGE.

```

DO 30 I=1,401
  BXH(I) = BXH(I) * (1.-A0) + TWOPI*A0*ATAN(A5*(XH(I)+A6))
30 CONTINUE
DX = 1.0 / IMA
IM2 = IMA * 2
IM2PI = IM2 * 1
DO 42 I=1,IM2PI
  IF (I .EQ. 1 .OR. I .EQ. IM2PI) GO TO 31
  BX(I) = (I-1) * DX - 1.0
  GO TO 32
31 CONTINUE
  IF (I .EQ. 1) BX(I) = -.999
  IF (I .EQ. IM2PI) BX(I) = .999
32 CONTINUE
  J = 0
33 CONTINUE
  J = J + 1
  IF (BX(I) .GT. BXH(J)) GO TO 33
  IF (BX(I) .LT. BXH(J)) GO TO 34
  XI = XH(J)
  GO TO 40
34 CONTINUE
  BT1 = BX(I) - BXH(J-1)
  BHT1 = BXH(J) - BXH(J-1)
  XHT1 = XH(J) - XH(J-1)
  T1 = XHT1 / BHT1
  XI = XH(J-1) + BT1 * T1
  IF (J .EQ. 2) GO TO 40
  BT2 = BX(I) - BXH(J)
  BHT2 = BXH(J) - BXH(J-2)
  BHT3 = BXH(J-1) - BXH(J-2)
  XHT2 = XH(J-1) - XH(J-2)
  T2 = XHT2 / BHT3
  T12 = T1 - T2
  BT12 = BT1 * BT2
  TBT2 = T12 / BHT2
  XI = XI + BT12 * TBT2
  IF (J .GE. 400) GO TO 40
  BT3 = BX(I) - BXH(J-2)
  BHT4 = BXH(J+1) - BXH(J-2)
  BHT5 = BXH(J+1) - BXH(J)
  BHT6 = BXH(J+1) - BXH(J-1)
  XHT3 = XH(J+1) - XH(J)
  T3 = XHT3 / BHT5
  XI = XI + BT12 * BT3/BHT4 + ((T3-T1)/BHT6 - TBT2)
40 CONTINUE
XIN(I) = XI + A3
42 CONTINUE
CALL ISLIT(XIN)
XFACT=1.0/XIN(ITE)
DO 43 I=1,IMAXI
  XIN(I)=XIN(I)*XFACT
43 CONTINUE
XIN(ITE)=1.0
IMAXI = IM2PI
JM2 = JMAXI

```

```

AYMESH 13
AYMESH 14
AYMESH 15
AYMESH 16
AYMESH 17
AYMESH 18
AYMESH 19
AYMESH 20
AYMESH 21
AYMESH 22
AYMESH 23
AYMESH 24
AYMESH 25
AYMESH 26
AYMESH 27
AYMESH 28
AYMESH 29
AYMESH 30
AYMESH 31
AYMESH 32
AYMESH 33
AYMESH 34
AYMESH 35
AYMESH 36
AYMESH 37
AYMESH 38
AYMESH 39
AYMESH 40
AYMESH 41
AYMESH 42
AYMESH 43
AYMESH 44
AYMESH 45
AYMESH 46
AYMESH 47
AYMESH 48
AYMESH 49
AYMESH 50
AYMESH 51
AYMESH 52
AYMESH 53
AYMESH 54
AYMESH 55
AYMESH 56
AYMESH 57
AYMESH 58
AYMESH 59
AYMESH 60
AYMESH 61
AYMESH 62
AYMESH 63
AYMESH 64
AYMESH 65
AYMESH 66
AYMESH 67
AYMESH 68
AYMESH 69
AYMESH 70
AYMESH 71
AYMESH 72
AYMESH 73
AYMESH 74
AYMESH 75
AYMESH 76
AYMESH 77
AYMESH 78
AYMESH 79
AYMESH 80
AYMESH 81
AYMESH 82
AYMESH 83
AYMESH 84
AYMESH 85
AYMESH 86
AYMESH 87
AYMESH 88
AYMESH 89
AYMESH 90
AYMESH 91
AYMESH 92
AYMESH 93
AYMESH 94
AYMESH 95
AYMESH 96
AYMESH 97
AYMESH 98

```

```

JMA = JM2 / 2
FJ = JMA
IF (BCTYPE .NE. 1) GO TO 44
C1 = CF1
C2 = CF2
C3 = CF3
GO TO 45
44 CONTINUE
C1 = CT1
C2 = CT2
C3 = CT3
45 CONTINUE
DETA = 1.0 / (FJ*C1)
IF (BCTYPF .EQ. 1) DETA = 1.0 / ((FJ+1.0)*C1)
C = C3 / (TAN(HALFPI*DETA*FJ))*C2
DO 50 I=1,JMA
J = I * JMA
ETA1 = I * DETA
YIN(I) = C * (TAN(HALFPI*ETA1))*C2
YIN(I+2*I+1) = -YIN(I)
50 CONTINUE
RETURN
END

```

```

AYMESH 99
AYMESH 100
AYMESH 101
AYMESH 102
AYMESH 103
AYMESH 104
AYMESH 105
AYMESH 106
AYMESH 107
AYMESH 108
AYMESH 109
AYMESH 110
AYMESH 111
AYMESH 112
AYMESH 113
AYMESH 114
AYMESH 115
AYMESH 116
AYMESH 117
AYMESH 118
AYMESH 119
AYMESH 120
AYMESH 121

```

BLOCK DATA

```

COMMON / COM1 / IMIN , IMAX , IUP , IODWN , ILE ,
ITE , JMIN , JMAX , JUP , JLOW ,
JTOP , JBOT
COMMON / COM2 / AK , ALPHA , OUB , GAM1 , RTK
LOGICAL / COM3 / ABORT , ABORT , ICUT , KSTEP
COMMON / COM4 / XIN(100) , YIN(100) , AMESH
COMMON / COM5 / XDIF(100) , YDIF(100)
COMMON / COM6 / FL(100) , FXL(100) , FU(100) , FXU(100) ,
CAMBER(100) , THICK(100) , VOL , XFOIL(100) , IFOIL
COMMON / COM7 / CJUP , CJUP1 , CJLW , CJLW1
COMMON / COM8 / CVERGE , DVERGE , IPRTER , MAXIT ,
WEI(3)
INTEGER / COM9 / BCF0IL , NL , NU , XL(100) , XU(100)
COMMON / COM10 / YLRDIN , YGRDIN , YL(100) , YU(100) , RIGF
LOGICAL XGRDIN , YGRDIN
COMMON / COM11 / YFREE(100) , YTUN(100) , XKRUPP(100) , GAM ,
JMXF , JMXT , XGRDIN , YGRDIN
INTEGER / COM12 / PSTART , PSAVE
LOGICAL / COM13 / ALPHA0 , CLOLD , DELTA0 , DUB0 , EMACH0
COMMON / COM14 / IMIND , IMAX0 , IMAXI , JMIND , JMAX0
1 JMAXI , PSAVE , PSTART , TITLE(6) , TITLE(6) ,
3 VOLJ , XOLD(100) , YOLD(100)
COMMON / COM15 / F , H , HALFPI , PI , RTKPOR ,
TWOP1
COMMON / COM16 / CDFACT , CLFACT , CMFACT , CPFAC , CPSTAR
LOGICAL / COM17 / FCR , KUTTA
COMMON / COM18 / CLSET , FCR , KUTTA , WCIRC , PSIO
COMMON / COM19 / PSI1 , PSI2
REAL JET
COMMON / COM20 / ALPHA0 , ALPHA1 , ALPHA2 , XSING ,
OMEGA0 , OMEGA1 , OMEGA2 , JET
COMMON / COM21 / CY8LC , CY8LD , CY8LU , CY8UC , CY8UD ,
CY8BU , FXLBC(100) , FXUBC(100)
LOGICAL / COM22 / ERROR , I1 , I2 , IERROR , JERROR ,
OUTERR , EMU(100,2) , VC(100)
1 MI , OCIRC , POLD(100,2)
2 COMMON / COM23 / DIAG(100) , RHS(100) , SUB(100) , SUP(100)
COMMON / COM24 / XMD(100) , YMD(100)
COMMON / COM25 / CX(100) , CXL(100) , CXR(100) , CXC(100) , CXXL(100) ,
CXKR(100) , C1(100)
COMMON / COM26 / CYC(100) , CYO(100) , CYYU(100) , IVAL
COMMON / COM27 / DTOP(100) , DBOT(100) , DUP(100) , DDDWN(100) ,
VTOP(100) , VBOT(100) , VUP(100) , VDDWN(100)
COMMON / COM28 / CPL(100) , CPU(100)

```

```

BLKDT 2
BLKDT 3
BLKDT 4
BLANK 2
COM1 2
COM1 3
COM1 4
COM2 2
COM3 2
COM3 3
COM4 2
COM4 3
COM5 2
COM6 2
COM6 3
COM7 2
COM8 2
COM8 3
COM9 2
COM9 3
COM9 4
COM10 2
COM10 3
COM10 4
COM11 2
COM11 3
COM11 4
COM11 5
COM11 6
COM11 7
COM12 2
COM12 3
COM13 2
COM14 2
COM14 3
COM15 2
COM15 3
COM16 2
COM16 3
COM16 4
COM17 2
COM17 3
COM18 2
COM18 3
COM18 4
COM18 5
COM19 2
COM20 2
COM22 2
COM22 3
COM23 2
COM24 2
COM24 3
COM25 2

```

```

COMMON / COM26 / P JUMP(100)
LOGICAL / COM27 / PHYS
INTEGER / COM28 / PRFLO , SIMDEF
COMMON / COM29 / CL , DELTA , DELRT2 , EMACH , EMROOT ,
1 PHYS , PRFFLO , SIMDEF , SONVEL , VFACT ,
2 YFACT
INTEGER / COM30 / BCTYPE
COMMON / COM31 / CIRCFF , FHINV , POR , CIRCTE
COMMON / COM32 / BIGRL , IRL , JRL
COMMON / COM33 / THETA(100,100)
DATA BCF0IL / 3 / , BCTYPE / 1 / , PSTART / 1 / ,
1 PRFLO / 1 / , SIMDEF / 3 /
DATA PHYS / 0. / , PSAVE / 0. / , FCR / 0. / ,
1 KUTTA / 0. / , ABORT / 0. / , AMESH / 0. / ,
DATA EMACH / 0.75 / , DELTA / 0.115 / , ALPHA / 0.12 / ,
1 AK / 0.0 / , GAM/1.4 / , RIGF/0.0 / , EPS/0.2 /
DATA CLSET / 0.0 / , CVERGE/0.0001 / , DVERGE/10. / ,
1 F / 0.0 / , H / 0.0 / , POR / 0.0 /
DATA WCIRC/1.0 / , XE/1.8 / , YGRDIN / F. /
DATA XGRDIN / F. / , YGRDIN / F. /
DATA IMAXI/77 / , JMXF/56 / , MAXIT/500 /
1 NL / 75 / , NU / 100 / , IPRTER / 10 /
DATA PI / 3.14159265 / , HALFPI / 1.570796325 / ,
1 TWOP1 / 6.28318531 /
DATA IMIN / 1 / , JMXT / 1 / , ICUT / 2 /
DATA CPU/1000.0 / , CPL/1000.0 /
DATA YIN / 100*0.0 / , JMAXI / 100 / , JMXT / 48 /
DATA XKRUPP / -1.075 , -0.950 ,
1 -0.825 , -0.7 , -0.575 , -0.45 , -0.35 ,
2 -0.25 , -0.175 , -0.125 , -0.075 , -0.025 ,
3 -0.035 , -0.0225 , -0.015 , -0.0075 , -0.0025 ,
4 0.0025 , 0.0075 , 0.0125 , 0.0175 , 0.0225 ,
5 0.0275 , 0.0325 , 0.0375 , 0.045 , 0.055 ,
6 0.065 , 0.075 , 0.085 , 0.0975 , 0.115 ,
7 0.140625 , 0.171875 , 0.203125 , 0.234375 , 0.265625 ,
8 0.296875 , 0.328125 , 0.359375 , 0.390625 , 0.421875 ,
9 0.453125 , 0.484375 , 0.515625 , 0.546875 , 0.578125 ,
1 0.009375 , 0.040625 , 0.071875 , 0.103125 , 0.134375 ,
2 0.165625 , 0.196875 , 0.228125 , 0.259375 , 0.290625 ,
3 0.321875 , 0.353125 , 0.384375 , 0.415625 , 0.446875 ,
4 0.478125 , 0.509375 , 0.540625 , 0.571875 , 0.603125 ,
5 0.634375 , 0.665625 , 0.696875 , 0.728125 , 0.759375 ,
6 0.790625 , 0.821875 , 0.853125 , 0.884375 , 0.915625 ,
7 0.946875 , 0.978125 , 1.009375 , 1.040625 , 1.071875 ,
8 1.103125 , 1.134375 , 1.165625 , 1.196875 , 1.228125 ,
9 1.259375 , 1.290625 , 1.321875 , 1.353125 , 1.384375 ,
1 1.415625 , 1.446875 , 1.478125 , 1.509375 , 1.540625 ,
2 1.571875 , 1.603125 , 1.634375 , 1.665625 , 1.696875 ,
3 1.728125 , 1.759375 , 1.790625 , 1.821875 , 1.853125 ,
4 1.884375 , 1.915625 , 1.946875 , 1.978125 , 2.009375 ,
5 2.040625 , 2.071875 , 2.103125 , 2.134375 , 2.165625 ,
6 2.196875 , 2.228125 , 2.259375 , 2.290625 , 2.321875 ,
7 2.353125 , 2.384375 , 2.415625 , 2.446875 , 2.478125 ,
8 2.509375 , 2.540625 , 2.571875 , 2.603125 , 2.634375 ,
9 2.665625 , 2.696875 , 2.728125 , 2.759375 , 2.790625 ,
1 2.821875 , 2.853125 , 2.884375 , 2.915625 , 2.946875 ,
2 2.978125 , 3.009375 , 3.040625 , 3.071875 , 3.103125 ,
3 3.134375 , 3.165625 , 3.196875 , 3.228125 , 3.259375 ,
4 3.290625 , 3.321875 , 3.353125 , 3.384375 , 3.415625 ,
5 3.446875 , 3.478125 , 3.509375 , 3.540625 , 3.571875 ,
6 3.603125 , 3.634375 , 3.665625 , 3.696875 , 3.728125 ,
7 3.759375 , 3.790625 , 3.821875 , 3.853125 , 3.884375 ,
8 3.915625 , 3.946875 , 3.978125 , 4.009375 , 4.040625 ,
9 4.071875 , 4.103125 , 4.134375 , 4.165625 , 4.196875 ,
1 4.228125 , 4.259375 , 4.290625 , 4.321875 , 4.353125 ,
2 4.384375 , 4.415625 , 4.446875 , 4.478125 , 4.509375 ,
3 4.540625 , 4.571875 , 4.603125 , 4.634375 , 4.665625 ,
4 4.696875 , 4.728125 , 4.759375 , 4.790625 , 4.821875 ,
5 4.853125 , 4.884375 , 4.915625 , 4.946875 , 4.978125 ,
6 5.009375 , 5.040625 , 5.071875 , 5.103125 , 5.134375 ,
7 5.165625 , 5.196875 , 5.228125 , 5.259375 , 5.290625 ,
8 5.321875 , 5.353125 , 5.384375 , 5.415625 , 5.446875 ,
9 5.478125 , 5.509375 , 5.540625 , 5.571875 , 5.603125 ,
1 5.634375 , 5.665625 , 5.696875 , 5.728125 , 5.759375 ,
2 5.790625 , 5.821875 , 5.853125 , 5.884375 , 5.915625 ,
3 5.946875 , 5.978125 , 6.009375 , 6.040625 , 6.071875 ,
4 6.103125 , 6.134375 , 6.165625 , 6.196875 , 6.228125 ,
5 6.259375 , 6.290625 , 6.321875 , 6.353125 , 6.384375 ,
6 6.415625 , 6.446875 , 6.478125 , 6.509375 , 6.540625 ,
7 6.571875 , 6.603125 , 6.634375 , 6.665625 , 6.696875 ,
8 6.728125 , 6.759375 , 6.790625 , 6.821875 , 6.853125 ,
9 6.884375 , 6.915625 , 6.946875 , 6.978125 , 7.009375 ,
1 7.040625 , 7.071875 , 7.103125 , 7.134375 , 7.165625 ,
2 7.196875 , 7.228125 , 7.259375 , 7.290625 , 7.321875 ,
3 7.353125 , 7.384375 , 7.415625 , 7.446875 , 7.478125 ,
4 7.509375 , 7.540625 , 7.571875 , 7.603125 , 7.634375 ,
5 7.665625 , 7.696875 , 7.728125 , 7.759375 , 7.790625 ,
6 7.821875 , 7.853125 , 7.884375 , 7.915625 , 7.946875 ,
7 7.978125 , 8.009375 , 8.040625 , 8.071875 , 8.103125 ,
8 8.134375 , 8.165625 , 8.196875 , 8.228125 , 8.259375 ,
9 8.290625 , 8.321875 , 8.353125 , 8.384375 , 8.415625 ,
1 8.446875 , 8.478125 , 8.509375 , 8.540625 , 8.571875 ,
2 8.603125 , 8.634375 , 8.665625 , 8.696875 , 8.728125 ,
3 8.759375 , 8.790625 , 8.821875 , 8.853125 , 8.884375 ,
4 8.915625 , 8.946875 , 8.978125 , 9.009375 , 9.040625 ,
5 9.071875 , 9.103125 , 9.134375 , 9.165625 , 9.196875 ,
6 9.228125 , 9.259375 , 9.290625 , 9.321875 , 9.353125 ,
7 9.384375 , 9.415625 , 9.446875 , 9.478125 , 9.509375 ,
8 9.540625 , 9.571875 , 9.603125 , 9.634375 , 9.665625 ,
9 9.696875 , 9.728125 , 9.759375 , 9.790625 , 9.821875 ,
1 9.853125 , 9.884375 , 9.915625 , 9.946875 , 9.978125 ,
2 10.009375 , 10.040625 , 10.071875 , 10.103125 , 10.134375 ,
3 10.165625 , 10.196875 , 10.228125 , 10.259375 , 10.290625 ,
4 10.321875 , 10.353125 , 10.384375 , 10.415625 , 10.446875 ,
5 10.478125 , 10.509375 , 10.540625 , 10.571875 , 10.603125 ,
6 10.634375 , 10.665625 , 10.696875 , 10.728125 , 10.759375 ,
7 10.790625 , 10.821875 , 10.853125 , 10.884375 , 10.915625 ,
8 10.946875 , 10.978125 , 11.009375 , 11.040625 , 11.071875 ,
9 11.103125 , 11.134375 , 11.165625 , 11.196875 , 11.228125 ,
1 11.259375 , 11.290625 , 11.321875 , 11.353125 , 11.384375 ,
2 11.415625 , 11.446875 , 11.478125 , 11.509375 , 11.540625 ,
3 11.571875 , 11.603125 , 11.634375 , 11.665625 , 11.696875 ,
4 11.728125 , 11.759375 , 11.790625 , 11.821875 , 11.853125 ,
5 11.884375 , 11.915625 , 11.946875 , 11.978125 , 12.009375 ,
6 12.040625 , 12.071875 , 12.103125 , 12.134375 , 12.165625 ,
7 12.196875 , 12.228125 , 12.259375 , 12.290625 , 12.321875 ,
8 12.353125 , 12.384375 , 12.415625 , 12.446875 , 12.478125 ,
9 12.509375 , 12.540625 , 12.571875 , 12.603125 , 12.634375 ,
1 12.665625 , 12.696875 , 12.728125 , 12.759375 , 12.790625 ,
2 12.821875 , 12.853125 , 12.884375 , 12.915625 , 12.946875 ,
3 12.978125 , 13.009375 , 13.040625 , 13.071875 , 13.103125 ,
4 13.134375 , 13.165625 , 13.196875 , 13.228125 , 13.259375 ,
5 13.290625 , 13.321875 , 13.353125 , 13.384375 , 13.415625 ,
6 13.446875 , 13.478125 , 13.509375 , 13.540625 , 13.571875 ,
7 13.603125 , 13.634375 , 13.665625 , 13.696875 , 13.728125 ,
8 13.759375 , 13.790625 , 13.821875 , 13.853125 , 13.884375 ,
9 13.915625 , 13.946875 , 13.978125 , 14.009375 , 14.040625 ,
1 14.071875 , 14.103125 , 14.134375 , 14.165625 , 14.196875 ,
2 14.228125 , 14.259375 , 14.290625 , 14.321875 , 14.353125 ,
3 14.384375 , 14.415625 , 14.446875 , 14.478125 , 14.509375 ,
4 14.540625 , 14.571875 , 14.603125 , 14.634375 , 14.665625 ,
5 14.696875 , 14.728125 , 14.759375 , 14.790625 , 14.821875 ,
6 14.853125 , 14.884375 , 14.915625 , 14.946875 , 14.978125 ,
7 15.009375 , 15.040625 , 15.071875 , 15.103125 , 15.134375 ,
8 15.165625 , 15.196875 , 15.228125 , 15.259375 , 15.290625 ,
9 15.321875 , 15.353125 , 15.384375 , 15.415625 , 15.446875 ,
1 15.478125 , 15.509375 , 15.540625 , 15.571875 , 15.603125 ,
2 15.634375 , 15.665625 , 15.696875 , 15.728125 , 15.759375 ,
3 15.790625 , 15.821875 , 15.853125 , 15.884375 , 15.915625 ,
4 15.946875 , 15.978125 , 16.009375 , 16.040625 , 16.071875 ,
5 16.103125 , 16.134375 , 16.165625 , 16.196875 , 16.228125 ,
6 16.259375 , 16.290625 , 16.321875 , 16.353125 , 16.384375 ,
7 16.415625 , 16.446875 , 16.478125 , 16.509375 , 16.540625 ,
8 16.571875 , 16.603125 , 16.634375 , 16.665625 , 16.696875 ,
9 16.728125 , 16.759375 , 16.790625 , 16.821875 , 16.853125 ,
1 16.884375 , 16.915625 , 16.946875 , 16.978125 , 17.009375 ,
2 17.040625 , 17.071875 , 17.103125 , 17.134375 , 17.165625 ,
3 17.196875 , 17.228125 , 17.259375 , 17.290625 , 17.321875 ,
4 17.353125 , 17.384375 , 17.415625 , 17.446875 , 17.478125 ,
5 17.509375 , 17.540625 , 17.571875 , 17.603125 , 17.634375 ,
6 17.665625 , 17.696875 , 17.728125 , 17.759375 , 17.790625 ,
7 17.821875 , 17.853125 , 17.884375 , 17.915625 , 17.946875 ,
8 17.978125 , 18.009375 , 18.040625 , 18.071875 , 18.103125 ,
9 18.134375 , 18.165625 , 18.196875 , 18.228125 , 18.259375 ,
1 18.290625 , 18.321875 , 18.353125 , 18.384375 , 18.415625 ,
2 18.446875 , 18.478125 , 18.509375 , 18.540625 , 18.571875 ,
3 18.603125 , 18.634375 , 18.665625 , 18.696875 , 18.728125 ,
4 18.759375 , 18.790625 , 18.821875 , 18.853125 , 18.884375 ,
5 18.915625 , 18.946875 , 18.978125 , 19.009375 , 19.040625 ,
6 19.071875 , 19.103125 , 19.134375 , 19.165625 , 19.196875 ,
7 19.228125 , 19.259375 , 19.290625 , 19.321875 , 19.353125 ,
8 19.384375 , 19.415625 , 19.446875 , 19.478125 , 19.509375 ,
9 19.540625 , 19.571875 , 19.603125 , 19.634375 , 19.665625 ,
1 19.696875 , 19.728125 , 19.759375 , 19.790625 , 19.821875 ,
2 19.853125 , 19.884375 , 19.915625 , 19.946875 , 19.978125 ,
3 20.009375 , 20.040625 , 20.071875 , 20.103125 , 20.134375 ,
4 20.165625 , 20.196875 , 20.228125 , 20.259375 , 20.290625 ,
5 20.321875 , 20.353125 , 20.384375 , 20.415625 , 20.446875 ,
6 20.478125 , 20.509375 , 20.540625 , 20.571875 , 20.603125 ,
7 20.634375 , 20.665625 , 20.696875 , 20.728125 , 20.759375 ,
8 20.790625 , 20.821875 , 20.853125 , 20.884375 , 20.915625 ,
9 20.946875 , 20.978125 , 21.009375 , 21.040625 , 21.071875 ,
1 21.103125 , 21.134375 , 21.165625 , 21.196875 , 21.228125 ,
2 21.259375 , 21.290625 , 21.321875 , 21.353125 , 21.384375 ,
3 21.415625 , 21.446875 , 21.478125 , 21.509375 , 21.540625 ,
4 21.571875 , 21.603125 , 21.634375 , 21.665625 , 21.696875 ,
5 21.728125 , 21.759375 , 21.790625 , 21.821875 , 21.853125 ,
6 21.884375 , 21.915625 , 21.946875 , 21.978125 , 22.009375 ,
7 22.040625 , 22.071875 , 22.103125 , 22.134375 , 22.165625 ,
8 22.196875 , 22.228125 , 22.259375 , 22.290625 , 22.321875 ,
9 22.353125 , 22.384375 , 22.415625 , 22.446875 , 22.478125 ,
1 22.509375 , 22.540625 , 22.571875 , 22.603125 , 22.634375 ,
2 22.665625 , 22.696875 , 22.728125 , 22.759375 , 22.790625 ,
3 22.821875 , 22.853125 , 22.884375 , 22.915625 , 22.946875 ,
4 22.978125 , 23.009375 , 23.040625 , 23.071875 , 23.103125 ,
5 23.134375 , 23.165625 , 23.196875 , 23.228125 , 23.259375 ,
6 23.290625 , 23.321875 , 23.353125 , 23.384375 , 23.415625 ,
7 23.446875 , 23.478125 , 23.509375 , 23.540625 , 23.571875 ,
8 23.603125 , 23.634375 , 23.665625 , 23.696875 , 23.728125 ,
9 23.759375 , 23.790625 , 23.821875 , 23.853125 , 23.884375 ,
1 23.915625 , 23.946875 , 23.978125 , 24.009375 , 24.040625 ,
2 24.071875 , 24.103125 , 24.134375 , 24.165625 , 24.196875 ,
3 24.228125 , 24.259375 , 24.290625 , 24.321875 , 24.353125 ,
4 24.384375 , 24.415625 , 24.446875 , 24.478125 , 24.509375 ,
5 24.540625 , 24.571875 , 24.603125 , 24.634375 , 24.665625 ,
6 24.696875 , 24.728125 , 24.759375 , 24.790625 , 24.821875 ,
7 24.853125 , 24.884375 , 24.915625 , 24.946875 , 24.978125 ,
8 25.009375 , 25.040625 , 25.071875 , 25.103125 , 25.134375 ,
9 25.165625 , 25.196875 , 25.228125 , 25.259375 , 25.290625 ,
1 25.321875 , 25.353125 , 25.384375 , 25.415625 , 25.446875 ,
2 25.478125 , 25.509375 , 25.540625 , 25.571875 , 25.603125 ,
3 25.634375 , 25.665625 , 25.696875 , 25.728125 , 25.759375 ,
4 25.790625 , 25.821875 , 25.853125 , 25.884375 , 25.915625 ,
5 25.946875 , 25.978125 , 26.009375 , 26.040625 , 26.071875 ,
6 26.103125 , 26.134375 , 26.165625 , 26.196875 , 26.228125 ,
7 26.259375 , 26.290625 , 26.321875 , 26.353125 , 26.384375 ,
8 26.415625 , 26.446875 , 26.478125 , 26.509375 , 26.540625 ,
9 26.571875 , 26.603125 , 26.634375 , 26.665625 , 26.696875 ,
1 26.728125 , 26.759375 , 26.790625 , 26.82
```



```

RHS(JBOT) = RHS(JBOT) - (CYYD(JBOT)*P(JMIN-P(JBOT-1,I)))
RHS(JTOP) = RHS(JTOP) - (CYYU(JTOP)*P(JMAX-P(JTOP+1,I)))
RETURN
      NEUMAN BOUNDARY CONDITIONS
95 CONTINUE
DIAG(JBOT) = DIAG(JBOT) + DFACL
DIAG(JTOP) = DIAG(JTOP) + DFACU
RHS(JBOT) = RHS(JBOT) - RFACL + CYYD(JBOT)*P(JBOT-1,I)
RHS(JTOP) = RHS(JTOP) - RFACU + CYYU(JTOP)*P(JTOP+1,I)
RETURN
END

SUBROUTINE BODY
      COMPUTES BODY GEOMETRY INFORMATION FOR BOUNDARY
      CONDITIONS AND OUTPUT INFORMATION. FOUR CHOICES OF
      BODY DESCRIPTION ARE AVAILABLE AS FOLLOWS
      BCFOIL = 1 NACA 00XX AIRFOIL
      BCFOIL = 2 PARABOLIC ARC AIRFOIL
      BCFOIL = 3 AIRFOIL ORDINATES READ IN
      BCFOIL = 4 EXTRA ADDRESS FOR USERS CHOICE
      BODY ORDINATES AND SLOPES ARE COMPUTED AT THE INPUT
      X MESH LOCATIONS AND ARE DIVIDED BY THE THICKNESS
      RATIO DELTA. THE BODY VOLUME, CAMBER, AND THICKNESS
      ARE ALSO COMPUTED.
      ACTUAL BOUNDARY CONDITION IS SET IN SUBROUTINE SETBC
      CALLED BY - TSFOIL.

COMMON / COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT
COMMON / COM2/ AK , ALP+A , DUB , GAM1 , RTK
LOGICAL AMESH
COMMON / COM4/ XIN(100) , YIN(100) , AMESH
COMMON / COM6/ FL(100) , FX(100) , FU(100) ,
1 CAMBER(100) , THICK(100) , VOL , XFOIL(100) , IFOIL
INTEGER BCFOIL
COMMON / COM9/ BCFOIL , NL , NU , XL(100) , XU(100) ,
1 YL(100) , YU(100) , RIGF
LOGICAL PHYS
INTEGER PRIFLC , SIMDEF
COMMON / COM27/ CL , DELTA , DELRTZ , EMACH , EMROOT ,
1 PHYS , PRIFLO , SIMDEF , SONVEL , VFACT
2 YFACT
COMMON / SPLN/ A(200) , B(200) , DY1 , DY2 , K1 ,
1 K2

      SET NUMBER OF POINTS ON AIRFOIL
      IFOIL = ITE - ILE + 1
      ZERO ALL THICKNESSES AND SLOPES
      DO 10 I=IMIN,IMAX
      FU(I) = 0.
      FL(I) = 0.
      FXU(I) = 0.
      FXL(I) = 0.
      CONTINUE
10 CONTINUE
      BRANCH TO APPROPRIATE AIRFOIL SPECIFICATION
100 GO TO (100,200,300,400), BCFOIL
      CONTINUE
      BCFOIL = 1
      FORMULA FOR NACA 00XX SHAPE
      IC = 0
      DO 125 I = ILE,ITE
      IC = IC + 1
      Z = XIN(I)
      XFOIL(IC) = Z
      RTZ = SQRT(Z)
      Z2 = Z*Z
      Z3 = Z*Z*Z
      Z4 = Z*Z*Z
      FU(IC) = 1.4845*RTZ - .63*Z - 1.750*Z2 + 1.4213*Z3 - .5075*Z4
      FL(IC) = -FU(IC)
      FXU(IC) = .74225*RTZ - .63 - 3.516*Z + 4.2645*Z2 - 2.03*Z3
      FXL(IC) = -FXU(IC)
125 CONTINUE
      GO TO 500
200 CONTINUE
      BCFOIL = 2
      PARABOLIC ARC AIRFOIL*****BI-CONVEX.

```

```

8CEND 104
8CEND 107
8CEND 108
8CEND 109
8CEND 110
8CEND 111
8CEND 112
8CEND 113
8CEND 114
8CEND 115
8CEND 116
8CEND 117

```

```

IC = 0
DO 225 I=ILE,ITE
IC = IC + 1
Z = XIN(I)
XFOIL(IC) = Z
Z2 = Z*Z
FU(IC) = Z*(Z - Z2)
FL(IC) = -FU(IC)
FXU(IC) = Z - 4.*Z
FXL(IC) = -FXU(IC)
225 CONTINUE
GO TO 500
300 CONTINUE
      BCFOIL = 3
      BODY ORDINATES READ IN NAMELIST
      DELINV = 1.
      IF(PHYS) DELINV = 1./DELTA
      COMPUTE ORDINATES AND SLOPES AT X MESH LOCATION ON
      AIRFOIL BY CUBIC SPLINE INTERPOLATION.
      DERIVATIVE END CONDITIONS ARE SPECIFIED AT X=0
      AND X=1
      K1 = 1
      K2 = 1
      UPPER SURFACE
      CALCULATE DY/DX AT END POINTS BY FINITE DIFFERENCE
      FORMULA.
      DY1 = (YU(2) - YU(1)) / (XU(2) - XU(1))
      DY2 = (YU(NU) - YU(NU-1)) / (XU(NU) - XU(NU-1))
      INITIALIZE CUBIC SPLINE INTERPOLATION
      CALL SPLN1(XU,YU,NU)
      CALCULATE ORDINATES AND SLOPES
      IC = 0
      DO 320 I=ILE,ITE
      IC = IC + 1
      XP = XIN(I)
      XFOIL(IC) = XP
      CALL SPLN1(XU,YU,NU)
      SET ORDINATE AND SLOPE OF AIRFOIL TO INTERPOLATED
      VALUE DIVIDED BY THICKNESS RATIO
      FU(IC) = YP*DELINV
      FXU(IC) = DYP*DELINV
      CONTINUE
      LOWER SURFACE
      CALCULATE DY/DX AT END POINTS BY FINITE DIFFERENCE
      FORMULA.
      DY1 = (YL(2) - YL(1)) / (XL(2) - XL(1))
      DY2 = (YL(NL) - YL(NL-1)) / (XL(NL) - XL(NL-1))
      INITIALIZE CUBIC SPLINE INTERPOLATION
      CALL SPLN1(XL,YL,NL)
      CALCULATE ORDINATES AND SLOPES
      IC = 0
      DO 330 I=ILE,ITE
      IC = IC + 1
      XP = XIN(I)
      CALL SPLN1(XL,YL,NL)
      SET ORDINATE AND SLOPE OF AIRFOIL TO INTERPOLATED
      VALUE DIVIDED BY THICKNESS RATIO
      FL(IC) = YP* DELINV
      FXL(IC) = DYP*DELINV
      CONTINUE
      GO TO 500
400 CONTINUE
      BCFOIL = 4
      THIS ADDRESS LEFT FOR USER TO INSERT OWN AIRFOIL
      DESCRIPTION.
      WRITE(6,1000)
1000 FORMAT(33H0ABNORMAL STOP IN SUBROUTINE BODY/
1 24H BCTYPE#4 IS NOT USEABLE)
      STOP
500 CONTINUE
      EXECUTE CALCULATIONS COMMON TO ALL AIRFOILS
      COMPUTE AIRFOIL VOLUME
      CALL SIMP(VOLU,*FOIL,FU,IFOIL,IERR)
      CALL SIMP(VOLL,*FOIL,FL,IFOIL,IERR)
      VOL = VOLU - VOLL
      COMPUTE CAMBER AND THICKNESS
      DO 525 I=1,IFOIL
      CAMBER(I) = .5*(FU(I) + FL(I))
      THICK(I) = .5*(FU(I) - FL(I))
525 CONTINUE
      DO 530 I=1,IFOIL
      FXU(I) = FXU(I) / SORT(1.0 + RIGF *(DELTA*FXU(I))**2 )
      FXL(I) = FXL(I) / SORT(1.0 + RIGF *(DELTA*FXL(I))**2 )
530 CONTINUE
      CALL PRBODY

```

```

BODY 58
BODY 59
BODY 60
BODY 61
BODY 62
BODY 63
BODY 64
BODY 65
BODY 66
BODY 67
BODY 68
BODY 69
BODY 70
BODY 71
BODY 72
BODY 73
BODY 74
BODY 75
BODY 76
BODY 77
BODY 78
BODY 79
BODY 80
BODY 81
BODY 82
BODY 83
BODY 84
BODY 85
BODY 86
BODY 87
BODY 88
BODY 89
BODY 90
BODY 91
BODY 92
BODY 93
BODY 94
BODY 95
BODY 96
BODY 97
BODY 98
BODY 99
BODY 100
BODY 101
BODY 102
BODY 103
BODY 104
BODY 105
BODY 106
BODY 107
BODY 108
BODY 109
BODY 110
BODY 111
BODY 112
BODY 113
BODY 114
BODY 115
BODY 116
BODY 117
BODY 118
BODY 119
BODY 120
BODY 121
BODY 122
BODY 123
BODY 124
BODY 125
BODY 126
BODY 127
BODY 128
BODY 129
BODY 130
BODY 131
BODY 132
BODY 133
BODY 134
BODY 135
BODY 136
BODY 137
BODY 138
BODY 139
BODY 140
BODY 141
BODY 142
BODY 143

```



```

Z(L) = Y(J)
ARG(L) = (PX(ISK+1,J) - PX(ISK-2,J))*3
210 CONTINUE
CALL TRAP(Z,ARG,L,SUM)
CDSK = -GAM1/6.*CDOFACT*SUM
CDWAVE = CDWAVE + CDSK
CALL PRTSK(Z,ARG,L,NSHOCK, CDSK,LPRT1)
220 CONTINUE
      INTEGRATE ALONG SHOCKS ABOVE AIRFOIL
C
ISTART = ILE
225 CONTINUE
CALL FINDSK(ISTART,ITE,JUP,ISK)
IF(ISK.LT. 0) GO TO 250
      SHOCK WAVE FOUND
C
ISTART = ISK + 1
NSHOCK = NSHOCK + 1
LPRT1 = 0
L = 1
Z(L) = 0.
ARG(L) = (CJUP*(PX(ISK+1,JUP) - PX(ISK-2,JUP))
* - CJUP1*(PX(ISK+1,JUP+1) - PX(ISK-2,JUP+1)))*3
DO 230 J = JUP, JT
  L = L+1
  Z(L) = Y(J)
  ARG(L) = (PX(ISK+1,J) - PX(ISK-2,J))*3
  ISKOLD = ISK
  JSK = J + 1
  CALL NEWISK(ISKOLD,JSK,ISK)
  IF(ISK.LT. 0) GO TO 240
  IF(ISK.GT. ID) GO TO 235
230 CONTINUE
235 CONTINUE
LPRT1 = 1
240 CONTINUE
CALL TRAP(Z,ARG,L,SUM)
CDSK = -GAM1/6.*CDOFACT*SUM
CDWAVE = CDWAVE + CDSK
CALL PRTSK(Z,ARG,L,NSHOCK,CDSK,LPRT1)
IF(LPRT1.EQ. 1) LPRT2 = 1
      RETURN TO FIND NEXT SHOCK
C
GO TO 225
      INTEGRATE ALONG SHOCKS BELOW AIRFOIL
250 CONTINUE
ISTART = ILE
260 CONTINUE
CALL FINDSK(ISTART,ITE,JLOW,ISK)
IF(ISK.LT. 0) GO TO 300
      SHOCK WAVE FOUND
C
ISTART = ISK + 1
NSHOCK = NSHOCK + 1
LPRT1 = 0
L = 1
Z(L) = 0.
ARG(L) = (CJLOW*(PX(ISK+1,JLOW) - PX(ISK-2,JLOW))
* - CJLOW1*(PX(ISK+1,JLOW-1) - PX(ISK-2,JLOW-1)))*3
DO 270 JJ = JB,JLOW
  J = JLOW + JB - JJ
  L = L+1
  Z(L) = Y(J)
  ARG(L) = (PX(ISK+1,J) - PX(ISK-2,J))*3
  ISKOLD = ISK
  JSK = J - 1
  CALL NEWISK(ISKOLD,JSK,ISK)
  IF(ISK.LT. 0) GO TO 280
  IF(ISK.GT. ID) GO TO 275
270 CONTINUE
275 CONTINUE
LPRT1 = 1
280 CONTINUE
CALL TRAP(Z,ARG,L,SUM)
CDSK = -GAM1/6.*CDOFACT*(-SUM)
CDWAVE = CDWAVE + CDSK
CALL PRTSK(Z,ARG,L,NSHOCK,CDSK,LPRT1)
IF(LPRT1.EQ. 1) LPRT2 = 1
      RETURN TO FIND NEXT SHOCK
C
GO TO 260
300 CONTINUE
      INTEGRATION ALONG SHOCKS IS COMPLETE
C
      PRINTOUT CD INFORMATION
C
XU = X(IU)
XD = X(ID)
YT = Y(JT)*YFACT
YB = Y(JB)*YFACT
CDC = CDUP + CDTOP + CDBOT + CDDOWN + COBODY

```

```

CDCOLE 157
CDCOLE 158
CDCOLE 159
CDCOLE 160
CDCOLE 161
CDCOLE 162
CDCOLE 163
CDCOLE 164
CDCOLE 165
CDCOLE 166
CDCOLE 167
CDCOLE 168
CDCOLE 169
CDCOLE 170
CDCOLE 171
CDCOLE 172
CDCOLE 173
CDCOLE 174
CDCOLE 175
CDCOLE 176
CDCOLE 177
CDCOLE 178
CDCOLE 179
CDCOLE 180
CDCOLE 181
CDCOLE 182
CDCOLE 183
CDCOLE 184
CDCOLE 185
CDCOLE 186
CDCOLE 187
CDCOLE 188
CDCOLE 189
CDCOLE 190
CDCOLE 191
CDCOLE 192
CDCOLE 193
CDCOLE 194
CDCOLE 195
CDCOLE 196
CDCOLE 197
CDCOLE 198
CDCOLE 199
CDCOLE 200
CDCOLE 201
CDCOLE 202
CDCOLE 203
CDCOLE 204
CDCOLE 205
CDCOLE 206
CDCOLE 207
CDCOLE 208
CDCOLE 209
CDCOLE 210
CDCOLE 211
CDCOLE 212
CDCOLE 213
CDCOLE 214
CDCOLE 215
CDCOLE 216
CDCOLE 217
CDCOLE 218
CDCOLE 219
CDCOLE 220
CDCOLE 221
CDCOLE 222
CDCOLE 223
CDCOLE 224
CDCOLE 225
CDCOLE 226
CDCOLE 227
CDCOLE 228
CDCOLE 229
CDCOLE 230
CDCOLE 231
CDCOLE 232
CDCOLE 233
CDCOLE 234
CDCOLE 235
CDCOLE 236
CDCOLE 237
CDCOLE 238
CDCOLE 239
CDCOLE 240
CDCOLE 241
CDCOLE 242

```

```

CD = CDC + CDWAVE
WRITE(6,1001)
WRITE(6,1002) XU,COUP, XD,CDDOWN, YT,CDTOP, YB,CDBOT
IF(ND.LT. 1.) WRITE(6,1003) XD,CBODY
WRITE(6,1004) CDC
WRITE(6,1005) NSHOCK,CDWAVE
IF(NSHOCK.GT. 0 .AND. LPRT2.EQ. 0) WRITE(6,1007)
IF(NSHOCK.GT. 0 .AND. LPRT2.EQ. 1) WRITE(6,1008)
WRITE(6,1006) CD
RETURN
325 CONTINUE
      SHOCK IS TOO CLOSE TO BODY TO DO CONTOUR INTEGRAL.
      WRITE MESSAGE AND RETURN
C
ULE = PX(ILE,JUP)
IF(ULE.GT. SONVEL) WRITE(6,1011)
IF(ULE.LE. SONVEL) WRITE(6,1012)
CD = DRAG(CDOFACT)
WRITE(6,1013) CD
RETURN
.350 WRITE(6,1014)
      CD=DRAG(CDOFACT)
      WRITE(6,1013) CD
      RETURN
1001 FORMAT(1M1,55X,23HDRAG COEFFICIENT OUTPUT/ 56X,23(1M)//,
1 60H CALCULATION OF DRAG COEFFICIENT BY MOMENTUM INTEGRAL METHOD)
1002 FORMAT(27HBOUNDARIES OF CONTOUR USED,15X,10HCONTRIBUTION TO CD/
* 16H UPSTREAM X =,F12.6,15X,8HCDUP =,F12.6/
* 16H DOWNSTREAM X =,F12.6,15X,8HCDDOWN =,F12.6/
* 16H TOP Y =,F12.6,15X,8HCDTOP =,F12.6/
* 16H BOTTOM Y =,F12.6,15X,8HCDBOT =,F12.6)
1003 FORMAT(16H BODY AFT OF X =,F12.6,15X,8HCBODY =,F12.6)
1004 FORMAT(15X,36HTOTAL CONTRIBUTIONS AROUND CONTOUR =,F12.6)
1005 FORMAT(10HOTHER ARE,13,38H SHOCKS INSIDE CONTOUR. TOTAL CDWAVE =,
* F12.6)
1006 FORMAT(51HDRAG CALCULATED FROM MOMENTUM INTEGRAL CD =,
* F12.6)
1007 FORMAT(43HNOTE = ALL SHOCKS CONTAINED WITHIN CONTOUR/
* 30H CDWAVE EQUALS TOTAL WAVE DRAG)
1008 FORMAT(52HNOTE = ONE OR MORE SHOCKS EXTEND OUTSIDE OF CONTOUR/
* 30H CDWAVE DOES NOT EQUAL TOTAL WAVE DRAG)
1011 FORMAT(31HSHOCK WAVE IS ATTACHED TO BODY/
* 33H MOMENTUM INTEGRAL CANNOT BE DONE/
* 45H DRAG OBTAINED FROM SURFACE PRESSURE INTEGRAL/)
1012 FORMAT(41HDETACHED SHOCK WAVE IS TOO CLOSE TO BODY/
* 33H MOMENTUM INTEGRAL CANNOT BE DONE/
* 45H DRAG OBTAINED FROM SURFACE PRESSURE INTEGRAL/)
1013 FORMAT(4HODD=,F12.6)
1014 FORMAT(1M1,17X,19H***** CAUTION *****16X,16HSOLUTION MAY BE ,
1 7HINVALID/46H DETACHED SHOCK WAVE IS TOO CLOSE TO UPSTREAM,
2 8HBOUNDARY/11X,32HMOMENTUM INTEGRAL CANNOT BE DONE/5X,
3 4HDRAG OBTAINED FROM SURFACE PRESSURE INTEGRAL)
END
SUBROUTINE CKMESH
C
C
C
C
CHECK X MESH AND ADJUST TO CONTAIN ODD NO. OF
POINTS BEFORE TAIL AND ODD NO. AFTER TAIL.
ITE IS INCLUDED IN BOTH COUNTS.
CKMESH IS CALLED ONCE BY READIN.
COMMON / COM1 / IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT
LOGICAL ABORT
COMMON / COM3 / IREF , ABORT , ICUT , KSTEP
LOGICAL AMESH
COMMON / COM4 / XIN(100) , YIN(100) , AMESH
IF (ICUT.GT. 0) GO TO 4
IREF = -1
RETURN
4 CONTINUE
C
C
C
TEST TO BE SURE THAT ADJUSTING THE NO. OF
POINTS WONT MAKE IMAX OR JMAX LARGER THAN 100.
IF (IMAX.LE. 98 .AND. JMAX.LE. 98) GO TO 8
WRITE (6,100)
IREF = -1
RETURN

```

```

CDCOLE 243
CDCOLE 244
CDCOLE 245
CDCOLE 246
CDCOLE 247
CDCOLE 248
CDCOLE 249
CDCOLE 250
CDCOLE 251
CDCOLE 252
CDCOLE 253
CDCOLE 254
CDCOLE 255
CDCOLE 256
CDCOLE 257
CDCOLE 258
CDCOLE 259
CDCOLE 260
CDCOLE 261
CDCOLE 262
CDCOLE 263
CDCOLE 264
CDCOLE 265
CDCOLE 266
CDCOLE 267
CDCOLE 268
CDCOLE 269
CDCOLE 270
CDCOLE 271
CDCOLE 272
CDCOLE 273
CDCOLE 274
CDCOLE 275
CDCOLE 276
CDCOLE 277
CDCOLE 278
CDCOLE 279
CDCOLE 280
CDCOLE 281
CDCOLE 282
CDCOLE 283
CDCOLE 284
CDCOLE 285
CDCOLE 286
CDCOLE 287
CDCOLE 288
CDCOLE 289
CDCOLE 290
CDCOLE 291
CDCOLE 292
CDCOLE 293
CDCOLE 294

```

```

CKMESH 2
CKMESH 3
CKMESH 4
CKMESH 5
CKMESH 6
CKMESH 7
CKMESH 8
COM1 2
COM1 3
COM1 4
COM3 2
COM3 3
COM4 2
COM4 3
CKMESH 12
CKMESH 13
CKMESH 14
CKMESH 15
CKMESH 16
CKMESH 17
CKMESH 18
CKMESH 19
CKMESH 20
CKMESH 21
CKMESH 22
CKMESH 23

```

```

8 CONTINUE
IF (MOD(IYE-IMIN+1, 2) .NE. 0) GO TO 20
C ADD EXTRA MESH POINT AHEAD OF AIRFOIL.
LP = IMAX + IMIN + 1
DO 10 I=IMIN,IMAX
L = LP - I
XIN(L) = XIN(L-1)
10 CONTINUE
IMAX = IMAX + 1
XIN(IMIN) = 2. * XIN(IMIN+1) - XIN(IMIN+2)
CALL ISLIT ( XIN )
20 CONTINUE
C ADD EXTRA MESH POINT AFTER AIRFOIL.
IF (MOD(IMAX-ITE+1, 2) .NE. 0) GO TO 30
IMAX = IMAX + 1
XIN(IMAX) = 2. * XIN(IMAX-1) - XIN(IMAX-2)
30 CONTINUE
C CHECK Y MESH AND ADJUST TO CONTAIN EVEN NO. OF
POINTS ABOVE AND BELOW SLIT.
IF (MOD(JLOW-JMIN, 2) .NE. 0) GO TO 50
C ADD EXTRA MESH POINT BELOW SLIT.
LP = JMAX + JMIN + 1
DO 40 J=JMIN,JMAX
L = LP - J
YIN(L) = YIN(L-1)
40 CONTINUE
JMAX = JMAX + 1
YIN(JMIN) = 2. * YIN(JMIN+1) - YIN(JMIN+2)
CALL JSPLIT ( YIN )
50 CONTINUE
C ADD EXTRA MESH POINT ABOVE SLIT.
IF (MOD(JMAX-JUP, 2) .NE. 0) GO TO 60
JMAX = JMAX + 1
YIN(JMAX) = 2.0 * YIN(JMAX-1) - YIN(JMAX-2)
60 CONTINUE
RETURN
100 FORMAT(9H)D THE MESH CANNOT BE ADJUSTED FOR CUTOUT, BECAUSE IMAX 0
1R JMAX IS TOO CLOSE TO THE LIMIT OF 100./
25X,19HIREF WAS SET TO 0 )
END

```

```

SUBROUTINE CPLOT (X, Y, Z, W, NP)
C SUBROUTINE CPLOT PRODUCES A PRINTER PLOT
C OF CRITICAL PRESSURE VS X.
C CALLED BY - FIXPLT.
LOGICAL AMESH
COMMON / COM4 / XIN(100), YIN(100), AMESH
DIMENSION X(101), Y(101), Z(101), W(101), M(120), ISYM(8)
DATA IC / 1, 1024, 1048576 /
DATA ISYM/1H,1HU,1HL,1HB,1H-,1HU,1HL,1HB/
C NC IS THE NUMBER OF COLUMNS.
C NR IS THE NUMBER OF ROWS.
DATA NC /120/, NR /50/
INITIALIZE RANGES
IF (AMESH) GO TO 3
NPL = 1
NPR = NP - 1
NL5 = 2
GO TO 4
3 CONTINUE
NPL = 2
NPR = NP - 2
NL5 = 3
4 CONTINUE
HL = X(NPL)
HR = X(NPR)
VB=AMIN1(Y(1),Z(1),W(1))
VT=AMAX1(Y(1),Z(1),W(1))
C DETERMINE RANGES
DO 5 I = NL5,NPR
HL=AMIN1(HL,X(I))
HR=AMAX1(HR,X(I))
VB=AMIN1(VB,Y(I),Z(I),W(I))
VT=AMAX1(VT,Y(I),Z(I),W(I))
5 CONTINUE

```

```

CKMESH 24
CKMESH 25
CKMESH 26
CKMESH 27
CKMESH 28
CKMESH 29
CKMESH 30
CKMESH 31
CKMESH 32
CKMESH 33
CKMESH 34
CKMESH 35
CKMESH 36
CKMESH 37
CKMESH 38
CKMESH 39
CKMESH 40
CKMESH 41
CKMESH 42
CKMESH 43
CKMESH 44
CKMESH 45
CKMESH 46
CKMESH 47
CKMESH 48
CKMESH 49
CKMESH 50
CKMESH 51
CKMESH 52
CKMESH 53
CKMESH 54
CKMESH 55
CKMESH 56
CKMESH 57
CKMESH 58
CKMESH 59
CKMESH 60
CKMESH 61
CKMESH 62
CKMESH 63
CKMESH 64
CKMESH 65

```

```

CPLOT 2
CPLOT 3
CPLOT 4
CPLOT 5
CPLOT 6
COM4 2
COM4 3
CPLOT 8
CPLOT 9
CPLOT 10
CPLOT 11
CPLOT 12
CPLOT 13
CPLOT 14
CPLOT 15
CPLOT 16
CPLOT 17
CPLOT 18
CPLOT 19
CPLOT 20
CPLOT 21
CPLOT 22
CPLOT 23
CPLOT 24
CPLOT 25
CPLOT 26
CPLOT 27
CPLOT 28
CPLOT 29
CPLOT 30
CPLOT 31
CPLOT 32
CPLOT 33
CPLOT 34
CPLOT 35
CPLOT 36
CPLOT 37

```

```

C SKIP TO NEW PAGE AND WRITE PLOT HEADING.
WRITE(6,900)
VDEL=(VT-VB)/FLOAT(NR)
HDEL=(HR-HL)/FLOAT(NC)
HDELM = 1.0 / HDEL
VL=VT
DD 100 IROW = 1, NR
VH=VL
VL=FLOAT(NR-IROW)*VDEL+VB
DO 15 I=1,NC
15 M(I)=0
DO 30 I =NPL,NPR
J = MAX0(1,MIN0(NC,1+INT((X(I)-HL) * HDELM)))
A(1) = Y(I)
A(2) = Z(I)
A(3) = W(I)
DO 20 K=1,3
IF (A(K) .GT. VH) GO TO 20
IF (A(K) .GT. VL .OR. (A(K) .LE. VB .AND. IROW .EQ. NR))
1 M(J) = M(J) + IC(K)
20 CONTINUE
30 CONTINUE
DD 90 I=1,NC
J = 1
IF (M(I) .LT. IC(3)) GO TO 70
J = J + 4
M(I) = MOD(M(I),IC(3))
70 CONTINUE
IF (M(I) .LT. IC(2)) GO TO 75
J = J + 2
M(I) = MOD(M(I),IC(2))
75 CONTINUE
IF (M(I) .LE. 0) GO TO 80
J = J + 1
80 CONTINUE
M(I) = ISYM(J)
90 CONTINUE
WRITE(6,901) (M(I),I=1,NC)
100 CONTINUE
RETURN
900 FORMAT(1H1,34X,46HPRINTER PLOT OF CP ON BODY AND DIVIDING STREAM,
1 4HLINE,15X,13HU = CP(UPPER)/35X,50(1H*),15X,13HL = CP(LOWER)/
2 100X,25HB = CP(UPPER) * CP(LOWER)/98X,17H--- = CRITICAL CP)
901 FORMAT(1X,120A1)
END
SUBROUTINE CUTOUT
C SUBROUTINE TO REDUCE THE NUMBER OF MESH POINTS
C FOR THE FIRST CUT AT SOLUTION. THE X-MESH AND
C Y-MESH WILL BE HALVED, AND IF POSSIBLE BE
C HALVED AGAIN.
C IREF =-1, CUTOUT SHOULDNT HAVE BEEN CALLED
C IREF = 0, IF NOT HALVED AT ALL.
C IREF = 1, IF HALVED ONCE.
C IREF = 2, IF HALVED TWICE.
C CALLED BY - TSFOIL.
COMMON / COM1 / IMIN, IMAX, IUP, IDOWN, ILE,
1 ITE, JMIN, JMAX, JUP, JLOW,
2 JTOP, JBOT
LOGICAL ABORT
COMMON / COM3 / IREF, ABORT, ICUT, KSTEP
LOGICAL AMESH
COMMON / COM4 / XIN(100), YIN(100), AMESH
COMMON / COM20 / XMID(100), YMID(100)

```

```

CPLOT 38
CPLOT 39
CPLOT 40
CPLOT 41
CPLOT 42
CPLOT 43
CPLOT 44
CPLOT 45
CPLOT 46
CPLOT 47
CPLOT 48
CPLOT 49
CPLOT 50
CPLOT 51
CPLOT 52
CPLOT 53
CPLOT 54
CPLOT 56
CPLOT 57
CPLOT 58
CPLOT 59
CPLOT 60
CPLOT 61
CPLOT 62
CPLOT 63
CPLOT 64
CPLOT 65
CPLOT 66
CPLOT 67
CPLOT 68
CPLOT 69
CPLOT 70
CPLOT 71
CPLOT 72
CPLOT 73
CPLOT 74
CPLOT 75
CPLOT 76
CPLOT 77
CPLOT 78
CPLOT 79
CPLOT 80
CPLOT 81
CPLOT 82
CPLOT 83
CPLOT 84
CPLOT 85
CPLOT 86
CPLOT 87
CPLOT 88
CPLOT 89
CPLOT 90
CPLOT 91
CPLOT 92
CUTOUT 2
CUTOUT 3
CUTOUT 4
CUTOUT 5
CUTOUT 6
CUTOUT 7
CUTOUT 8
CUTOUT 9
CUTOUT 10
CUTOUT 11
CUTOUT 12
CUTOUT 13
CUTOUT 14
BLANK 2
COM1 2
COM1 3
COM1 4
COM1 4
COM3 2
COM3 3
COM4 2
COM4 3
COM4 2
COM20 2
CUTOUT 19

```

```

C IF (IREF .NE. -1) GO TO 5
      MESH CANNOT BE REFINED, LOAD X*Y,YIN INTO X,Y
C DO 1 I = IMIN,IMAX
      X(I) = XIN(I)
1 CONTINUE
      DO 2 J = JMIN,JMAX
      Y(J) = YIN(J)
2 CONTINUE
      IREF = 0
      RETURN
5 CONTINUE
      K = IMIN - 1
      DO 10 I=IMIN,IMAX,2
      K = K + 1
      XMID(K) = XIN(I)
10 CONTINUE
      IMAX = (IMAX-IMIN) * .5 + IMIN
      CALL ISLIT ( XMID )
      K = JMIN - 1
      JE = JLOW - 1
      DO 15 J=JMIN,JE,2
      K = K + 1
      YMID(K) = YIN(J)
15 CONTINUE
      JST = JUP + 1
      DO 20 J=JST,JMAX,2
      K = K + 1
      YMDIK) = YIN(J)
20 CONTINUE
      JMAX = (JMAX-JMIN) * .5 + JMIN
      CALL JSLIT ( YMID )
      IREF = 1
C FIRST HALVING COMPLETE. CHECK IF NO. OF POINTS
      IS ODD.
C IF (ICUT .EQ. 1) GO TO 30
      IF (MOD(IIE-IMIN+1,2) .EQ. 0) GO TO 30
      IF (MOD(IMAX-IIE+2,2) .EQ. 0) GO TO 30
      IF (MOD(JLOW-JMIN,2) .EQ. 0) GO TO 30
      IF (MOD(JMAX-JUP,2) .NE. 0) GO TO 60
30 CONTINUE
      ONLY ONE MESH REFINEMENT POSSIBLE.
C DO 40 I=IMIN,IMAX
      X(I) = XMID(I)
40 CONTINUE
      DO 50 J=JMIN,JMAX
      Y(J) = YMID(J)
50 CONTINUE
      RETURN
C ALL POINTS ARE ODD SO CUT AGAIN.
60 CONTINUE
      K = IMIN - 1
      DO 70 I=IMIN,IMAX,2
      K = K + 1
      X(K) = XMID(I)
70 CONTINUE
      IMAX = (IMAX-IMIN) * .5 + IMIN
      CALL ISLIT ( X )
      K = JMIN - 1
      JE = JLOW - 1
      DO 75 J=JMIN,JE,2
      K = K + 1
      Y(K) = YMID(J)
75 CONTINUE
      JST = JUP + 1
      DO 80 J=JST,JMAX,2
      K = K + 1
      Y(K) = YMD(J)
80 CONTINUE
      JMAX = (JMAX-JMIN) * .5 + JMIN
      CALL JSLIT ( Y )
      IREF = 2
      RETURN
END

```

```

CUTOUT 20
CUTOUT 21
CUTOUT 22
CUTOUT 23
CUTOUT 24
CUTOUT 25
CUTOUT 26
CUTOUT 27
CUTOUT 28
CUTOUT 29
CUTOUT 30
CUTOUT 31
CUTOUT 32
CUTOUT 33
CUTOUT 34
CUTOUT 35
CUTOUT 36
CUTOUT 37
CUTOUT 38
CUTOUT 39
CUTOUT 40
CUTOUT 41
CUTOUT 42
CUTOUT 43
CUTOUT 44
CUTOUT 45
CUTOUT 46
CUTOUT 47
CUTOUT 48
CUTOUT 49
CUTOUT 50
CUTOUT 51
CUTOUT 52
CUTOUT 53
CUTOUT 54
CUTOUT 55
CUTOUT 56
CUTOUT 57
CUTOUT 58
CUTOUT 59
CUTOUT 60
CUTOUT 61
CUTOUT 62
CUTOUT 63
CUTOUT 64
CUTOUT 65
CUTOUT 66
CUTOUT 67
CUTOUT 68
CUTOUT 69
CUTOUT 70
CUTOUT 71
CUTOUT 72
CUTOUT 73
CUTOUT 74
CUTOUT 75
CUTOUT 76
CUTOUT 77
CUTOUT 78
CUTOUT 79
CUTOUT 80
CUTOUT 81
CUTOUT 82
CUTOUT 83
CUTOUT 84
CUTOUT 85
CUTOUT 86
CUTOUT 87
CUTOUT 88
CUTOUT 89
CUTOUT 90
CUTOUT 91
CUTOUT 92
CUTOUT 93

```

```

SUBROUTINE DIFCOE
C COMPUTE DIFFERENCE COEFFICIENTS IN FIELD
C CALLED BY - TSFOIL.
COMMON / COM1 / IMIN , IMAX , IUP , IDOWN , ILE ,
1 IIE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT , ALPHA , DUB , GAM1 , RTK
COMMON / COM2 / AK
COMMON / COM5 / XDIFF(100),YDIFF(100)
COMMON / COM7 / CJUP , CJLW , CJLOW , CJLOW1
COMMON / COM17 / CYYBLC , CYYBLD , CYYBLU , CYYBUC , CYYBUD ,
1 CYYBUU , FXLBC(100),FXUBC(100)
COMMON / COM22 / CXCL(100) , CXL(100) , CXR(100) , CXXC(100),CXLL(100),
1 CXRR(100) , CII(100)
COMMON / COM23 / CYYC(100), CYYD(100),CYYU(100) , IVAL
C COEFFICIENTS FOR (PIX AND (PI)XX AT IMIN
CXXL(IMIN) = 0.0
CXRR(IMIN) = 0.0
CXXC(IMIN) = 0.0
CXL(IMIN) = 0.0
CXR(IMIN) = 0.0
CXC(IMIN) = 0.0
C COEFFICIENTS FOR (PIX AND (PI)XX FROM I=IMIN+1 TO I=IMAX-1
C2 = GAM1*.5
ISTART = IMIN+1
IEND = IMAX-1
DO 1 I=ISTART,IEND
DXL=X(I)-X(I-1)
DXR=X(I+1)-X(I)
DXC = .5 * (X(I+1) - X(I-1))
C FOR VC
CII(I) = AK/DXC
C FOR (PIX)
CXL(I) = -C2 / (DXL*DXC)
CXR(I) = C2 / (DXR*DXC)
CXC(I) = -CXL(I) - CXR(I)
C FOR (PI)XX
CXXL(I) = 1.0 / DXL
CXRR(I) = 1.0 / DXR
CXXC(I) = CXXL(I) + CXRR(I)
CONTINUE
C COEFFICIENTS FOR (PIX AND (PI)XX AT IMAX
DX = X(IMAX) - X(IMAX-1)
Q = 1.0 / (DX*DX)
CII(IMAX) = AK / DX
CXL(IMAX) = -C2 * Q
CXR(IMAX) = C2 * Q
CXC(IMAX) = 0.0
CXXL(IMAX) = 1.0 / DX
CXRR(IMAX) = 1.0 / DX
CXXC(IMAX) = CXXL(IMAX) + CXRR(IMAX)
C COEFFICIENTS FOR (PI)YY AT JMIN.
DYU = Y(JMIN+1) - Y(JMIN)
CYYD(JMIN) = 2.0 / DYU
CYYU(JMIN) = 2. / (DYU*DYU)
CYYC(JMIN) = CYYU(JMIN)
C COEFFICIENTS FOR (PI)YY FROM J=JMIN+1 TO J=JMAX-1
JSTART = JMIN + 1
JEND = JMAX-1
DO 2 J=JSTART,JEND
DYD=Y(J)-Y(J-1)
DYU=Y(J+1)-Y(J)
DYC=Y(J+1)-Y(J-1)
CYYD(J)=2. / (DYD*DYC)
CYYU(J)=2. / (DYU*DYC)
CYYC(J) = CYYD(J) + CYYU(J)
CONTINUE
C COEFFICIENTS FOR (PI)YY AT JMAX.
DYD = Y(JMAX) - Y(JMAX-1)
CYYD(JMAX) = 2. / (DYD*DYD)
CYYU(JMAX) = 2. / DYD
CYYC(JMAX) = CYYD(JMAX)
C COEFFICIENTS FOR VELOCITY FORMULAS
ISTART = IMIN + 1
DO 3 I = ISTART, IMAX
XDIFF(I) = 1. / (X(I) - X(I-1))
CONTINUE
JSTART = JMIN + 1
DO 5 J = JSTART , JMAX
YDIFF(J) = 1. / (Y(J) - Y(J-1))

```

```

DIFCOE 2
DIFCOE 3
DIFCOE 4
DIFCOE 5
DIFCOE 6
BLANK 2
COM1 2
COM1 3
COM1 4
COM2 2
COM2 2
COM5 2
COM7 2
COM7 2
COM17 2
COM17 3
COM22 2
COM22 3
COM23 3
DIFCOE 15
DIFCOE 16
DIFCOE 17
DIFCOE 18
DIFCOE 19
DIFCOE 20
DIFCOE 21
DIFCOE 22
DIFCOE 23
DIFCOE 24
DIFCOE 25
DIFCOE 26
DIFCOE 27
DIFCOE 28
DIFCOE 29
DIFCOE 30
DIFCOE 31
DIFCOE 32
DIFCOE 33
DIFCOE 34
DIFCOE 35
DIFCOE 36
DIFCOE 37
DIFCOE 38
DIFCOE 39
DIFCOE 40
DIFCOE 41
DIFCOE 42
DIFCOE 43
DIFCOE 44
DIFCOE 45
DIFCOE 46
DIFCOE 47
DIFCOE 48
DIFCOE 49
DIFCOE 50
DIFCOE 51
DIFCOE 52
DIFCOE 53
DIFCOE 54
DIFCOE 55
DIFCOE 56
DIFCOE 57
DIFCOE 58
DIFCOE 59
DIFCOE 60
DIFCOE 61
DIFCOE 62
DIFCOE 63
DIFCOE 64
DIFCOE 65
DIFCOE 66
DIFCOE 67
DIFCOE 68
DIFCOE 69
DIFCOE 70
DIFCOE 71
DIFCOE 72
DIFCOE 73
DIFCOE 74
DIFCOE 75
DIFCOE 76
DIFCOE 77
DIFCOE 78
DIFCOE 79
DIFCOE 80
DIFCOE 81
DIFCOE 82
DIFCOE 83

```


5 CONTINUE

C COEFFICIENTS FOR EXTRAPOLATION FORMULAS FOR AIRFOIL SURFACE PROPERTIES

CJLOW = -Y(JLOW-1) / (Y(JLOW) - Y(JLOW-1))
CJLW1 = -Y(JLOW) / (Y(JLOW) - Y(JLOW-2))
CJUP = Y(JUP+1) / (Y(JUP+1) - Y(JUP))
CJUP1 = Y(JUP) / (Y(JUP+1) - Y(JUP))

C COMPUTE SPECIAL DIFFERENCE COEFFICIENTS FOR PYY
C TO USE FOR AIRFOIL BOUNDARY CONDITION
C DIFFERENCE COEFFICIENTS FOR UPPER SURFACE

CYYBUD = -2. / (Y(JUP+1) + Y(JUP))
CYYBUC = -CYYBUD / (Y(JUP+1) - Y(JUP))
CYYBUU = CYYBUC

C DIFFERENCE COEFFICIENTS FOR LOWER SURFACE

CYYBLU = -2.0 / (Y(JLOW) + Y(JLOW-1))
CYYBLC = CYYBLU / (Y(JLOW) - Y(JLOW-1))
CYYBLD = CYYBLC
RETURN
END

DIFCOE 84
DIFCOE 85
DIFCOE 86
DIFCOE 87
DIFCOE 88
DIFCOE 89
DIFCOE 90
DIFCOE 91
DIFCOE 92
DIFCOE 93
DIFCOE 94
DIFCOE 95
DIFCOE 96
DIFCOE 97
DIFCOE 98
DIFCOE 99
DIFCOE 100
DIFCOE 101
DIFCOE 102
DIFCOE 103

DALPHA = ABS(ALPHA0 - TEMP)
IF(DALPHA .LT. ERROR) GO TO 15
CONTINUE
N = 0
GO TO 9999
CONTINUE

C COMPUTE ALPHA1

ALPHA1 = 0.
DO 20 I = 1,100
TEMP = ALPHA1
Q = F*(TEMP - P1) - RTKPOR
ALPHA1 = HALFPI - ATAN(Q)
DALPHA = ABS(ALPHA1 - TEMP)
IF(DALPHA .LT. ERROR) GO TO 25
CONTINUE

C COMPUTE ALPHA2

ALPHA2 = 0.
DO 30 I = 1,100
TEMP = ALPHA2
Q = F*(TEMP - TWOPI) - RTKPOR
ALPHA2 = HALFPI - ATAN(Q)
DALPHA = ABS(ALPHA2 - TEMP)
IF(DALPHA .LT. ERROR) GO TO 35
CONTINUE

C COMPUTE ALPHA2

ALPHA2 = 0.
DO 30 I = 1,100
TEMP = ALPHA2
Q = F*(TEMP - TWOPI) - RTKPOR
ALPHA2 = HALFPI - ATAN(Q)
DALPHA = ABS(ALPHA2 - TEMP)
IF(DALPHA .LT. ERROR) GO TO 35
CONTINUE

C COMPUTE ALPHA2

ALPHA2 = 0.
DO 30 I = 1,100
TEMP = ALPHA2
Q = F*(TEMP - TWOPI) - RTKPOR
ALPHA2 = HALFPI - ATAN(Q)
DALPHA = ABS(ALPHA2 - TEMP)
IF(DALPHA .LT. ERROR) GO TO 35
CONTINUE

C COMPUTE ALPHA2

ALPHA2 = 0.
DO 30 I = 1,100
TEMP = ALPHA2
Q = F*(TEMP - TWOPI) - RTKPOR
ALPHA2 = HALFPI - ATAN(Q)
DALPHA = ABS(ALPHA2 - TEMP)
IF(DALPHA .LT. ERROR) GO TO 35
CONTINUE

C COMPUTE ALPHA2

ALPHA2 = 0.
DO 30 I = 1,100
TEMP = ALPHA2
Q = F*(TEMP - TWOPI) - RTKPOR
ALPHA2 = HALFPI - ATAN(Q)
DALPHA = ABS(ALPHA2 - TEMP)
IF(DALPHA .LT. ERROR) GO TO 35
CONTINUE

C COMPUTE ALPHA2

ALPHA2 = 0.
DO 30 I = 1,100
TEMP = ALPHA2
Q = F*(TEMP - TWOPI) - RTKPOR
ALPHA2 = HALFPI - ATAN(Q)
DALPHA = ABS(ALPHA2 - TEMP)
IF(DALPHA .LT. ERROR) GO TO 35
CONTINUE

C COMPUTE ALPHA2

ALPHA2 = 0.
DO 30 I = 1,100
TEMP = ALPHA2
Q = F*(TEMP - TWOPI) - RTKPOR
ALPHA2 = HALFPI - ATAN(Q)
DALPHA = ABS(ALPHA2 - TEMP)
IF(DALPHA .LT. ERROR) GO TO 35
CONTINUE

C COMPUTE ALPHA2

ALPHA2 = 0.
DO 30 I = 1,100
TEMP = ALPHA2
Q = F*(TEMP - TWOPI) - RTKPOR
ALPHA2 = HALFPI - ATAN(Q)
DALPHA = ABS(ALPHA2 - TEMP)
IF(DALPHA .LT. ERROR) GO TO 35
CONTINUE

C COMPUTE ALPHA2

ALPHA2 = 0.
DO 30 I = 1,100
TEMP = ALPHA2
Q = F*(TEMP - TWOPI) - RTKPOR
ALPHA2 = HALFPI - ATAN(Q)
DALPHA = ABS(ALPHA2 - TEMP)
IF(DALPHA .LT. ERROR) GO TO 35
CONTINUE

C COMPUTE ALPHA2

ALPHA2 = 0.
DO 30 I = 1,100
TEMP = ALPHA2
Q = F*(TEMP - TWOPI) - RTKPOR
ALPHA2 = HALFPI - ATAN(Q)
DALPHA = ABS(ALPHA2 - TEMP)
IF(DALPHA .LT. ERROR) GO TO 35
CONTINUE

C COMPUTE ALPHA2

ALPHA2 = 0.
DO 30 I = 1,100
TEMP = ALPHA2
Q = F*(TEMP - TWOPI) - RTKPOR
ALPHA2 = HALFPI - ATAN(Q)
DALPHA = ABS(ALPHA2 - TEMP)
IF(DALPHA .LT. ERROR) GO TO 35
CONTINUE

DROOTS 18
DROOTS 19
DROOTS 20
DROOTS 21
DROOTS 22
DROOTS 23
DROOTS 24
DROOTS 25
DROOTS 26
DROOTS 27
DROOTS 28
DROOTS 29
DROOTS 30
DROOTS 31
DROOTS 32
DROOTS 33
DROOTS 34
DROOTS 35
DROOTS 36
DROOTS 37
DROOTS 38
DROOTS 39
DROOTS 40
DROOTS 41
DROOTS 42
DROOTS 43
DROOTS 44
DROOTS 45
DROOTS 46
DROOTS 47
DROOTS 48
DROOTS 49
DROOTS 50
DROOTS 51
DROOTS 52
DROOTS 53
DROOTS 54
DROOTS 55
DROOTS 56
DROOTS 57
DROOTS 58
DROOTS 59
DROOTS 60
DROOTS 61
DROOTS 62

C FUNCTION DRAG(CDFACT)
C COMPUTES PRESSURE DRAG BY INTEGRATING U*W AROUND
C AIRFOIL USING TRAPEZODIAL RULE.
C CALLED BY = CDDOLE.

COMMON / COM1 / P(102,101), X(100), Y(100)
COMMON / COM1 / IMIN, IMAX, IUP, IDOWN, ILE,
1 ITE, IMIN, JMAX, JUP, JLOW,
2 ITOP, JBOT
COMMON / COM6 / FL(100), FXL(100), FU(100), FXU(100),
1 CAMBER(100), THICK(100), VOL, XFOIL(100), IFOIL
COMMON / COM7 / CJUP, CJUP1, CJLOW, CJLDW1
COMMON / COM30 / XI(100), ARG(100), REST(204)

K = 1
ARG(K) = 0.
XI(K) = X(ILE-1)
DO 10 I = ILE,ITE
K = K+1
PXUP = CJUP*PX(I,JUP) - CJUP1*PX(I,JUP+1)
PXLOW = CJLOW*PX(I,JLOW) - CJLOW1*PX(I,JLOW-1)
ARG(K) = FXU(K-1)*PXUP - FXL(K-1)*PXLOW
XI(K) = X(I)

10 CONTINUE
K = K + 1
ARG(K) = 0.
XI(K) = X(ITE+1)
CALL TRAP(XI,ARG,K,SUM)
DRAG = - SUM*CDFACT*2.
RETURN
END

DRAG 2
DRAG 3
DRAG 4
DRAG 5
DRAG 6
BLANK 2
COM1 2
COM1 3
COM1 4
COM6 2
COM6 3
COM7 2
DRAG 11
DRAG 12
DRAG 13
DRAG 14
DRAG 15
DRAG 16
DRAG 17
DRAG 16
DRAG 19
DRAG 20
DRAG 21
DRAG 22
DRAG 23
DRAG 24
DRAG 25
DRAG 26
DRAG 27
DRAG 28
DRAG 29

9999 CONTINUE
WRITE(6,1000) N
1000 FORMAT(35H1ABNORMAL STOP IN SUBROUTINE DROOTS/
1 38HNONCONVERGENCE OF ITERATION FOR ALPHA, I)
STOP
END

C SUBROUTINE ECHINP
C PRINTS INPUT CARDS USED FOR RUN.
C CALLED BY = TSFOIL.

C COMPUTE CONSTANTS ALPHA0, ALPHA1, ALPHA2, OMEGA0, OMEGA1
C OMEGA2 USED IN FORMULA FOR DOUBLE IN SLOTTED WIND
C TUNNEL WITH SUBSONIC FREESTREAM
C CALLED BY = FARFLD.

COMMON / COM12 / F, H, HALFPI, PI, RTKPOR,
1 TWOPI
REAL JET
COMMON / COM16 / ALPHA0, ALPHA1, ALPHA2, XSING,
1 OMEGA0, OMEGA1, OMEGA2, JET

ERROR = .00001
C COMPUTE ALPHA0
ALPHA0 = 0.
DO 10 I = 1,100
TEMP = ALPHA0
Q = F*TEMP - RTKPOR
ALPHA0 = HALFPI - ATAN(Q)

DROOTS 2
DROOTS 3
DROOTS 4
DROOTS 5
DROOTS 6
DROOTS 7
COM12 2
COM12 3
COM16 2
COM16 3
COM16 4
DROOTS 10
DROOTS 11
DROOTS 12
DROOTS 13
DROOTS 14
DROOTS 15
DROOTS 16
DROOTS 17

10 CONTINUE
WRITE(6,110)
READ(5,100) CRD
YES, NO
IF (EDF(5)) 30,20
CONTINUE
WRITE(6,101) CRD
GO TO 10
CONTINUE
REWIND 5
RETURN
100 FORMAT(8A10)
101 FORMAT(1X,8A10)
110 FORMAT(1H1,49X,35H1LISTING OF INPUT CARDS FOR THIS RUN/40X,35(1H*))
1 ///
END

ECHINP 2
ECHINP 3
ECHINP 4
ECHINP 5
ECHINP 6
ECHINP 7
ECHINP 8
ECHINP 9
ECHINP 10
ECHINP 11
ECHINP 12
ECHINP 13
ECHINP 14
ECHINP 15
ECHINP 16
ECHINP 17
ECHINP 18
ECHINP 19
ECHINP 20
ECHINP 21
ECHINP 22
ECHINP 23

```

FUNCTION EMACH1(U)
      FUNCTION EMACH1 COMPUTES LOCAL SIMILARITY PARAMETER
      OR LOCAL MACH NUMBER
      CALLED BY - MACHNP, PRINT1, PRTFLO.
      COMMON / COM2/ AK      , ALPHA , DUB , GAM1 , RTK
      LOGICAL      PHYS
      INTEGER      PRTFLO , SIMDEF , DELRT2 , EMACH , EHR00T ,
      COMMON / COM27/ CL     , DELTA , DELRT2 , EMACH , EHR00T ,
      PHYS , PRTFLO , SIMDEF , SONVEL , VFACT ,
      YFACT
      COMPUTE SIMILARITY PARAMETER BASED ON LOCAL VELOCITY
      AK1 = AK - GAM1*U
      IF (PHYS) GO TO 5
      RETURN VALUE OF LOCAL SIMILARITY PARAMETER
      EMACH1 = AK1
      RETURN
      CONTINUE
      COMPUTE VALUE OF LOCAL MACH NUMBER AND RETURN
      COLE SCALING
      ARG = DELRT2*AK1
      SPREITER SCALING
      IF (SIMDEF .EQ. 2) ARG = ARG*EMR00T*EMR00T
      KRUPP SCALING
      IF (SIMDEF .EQ. 3) ARG = ARG*EMACH
      ARG = 1. - ARG
      EMACH1 = 0.
      IF (ARG .GT. 0.) EMACH1 = SQRT(ARG)
      RETURN
      END

SUBROUTINE EXTRAP(XP,YP,PNEW)
      COMPUTE P AT X, YP USING FAR FIELD SOLUTION
      FOR SUBSONIC FLOW
      CALLED BY - GUESSP.
      COMMON / COM2/ AK      , ALPHA , DUB , GAM1 , RTK
      COMMON / COM12/ F      , H      , HALFPI , PI , RTKPOR ,
      TWOPI
      COMMON / COM15/ B      , BETA0 , BETA1 , BETA2 , PSIO ,
      PSI1 , PSI2
      REAL JET
      COMMON / COM16/ ALPHA1 , ALPHA2 , ALPHA3 , XSING ,
      OMEGA0 , OMEGA1 , OMEGA2 , JET
      INTEGER      BCTYPE
      COMMON / COM28/ BCTYPE , CIRCF , FHINV , POR , CIRCFE
      IF (BCTYPE .NE. 1) GO TO 100
      FREE AIR BOUNDARY CONDITION
      IF (ABS(YP) .LT. 1.E-6) YP = -1.E-6
      XI = XP - XSING
      ETA = YP*RTK
      PNEW = - CIRCF/TWOPI*(ATAN2(ETA,XI) + PI - SIGN(PI,ETA))
      + DUB / TWOPI/RTK*(XI/(XI*XI + ETA*ETA))
      RETURN
      CONTINUE
      TUNNEL WALL BOUNDARY CONDITION
      ETA = YP/H
      XI = (XP - XSING)/(H*RTK)
      IF (XI .LT. 0.) GO TO 200
      XP IS DOWNSTREAM OF AIRFOIL
      TERM = ETA
      IF (BCTYPE .NE. 3) TERM = SIN(ETA*BETA0)/BETA0
      PNEW = - .5*CIRCF*PI - SIGN(1.,ETA)
      + (1.-JET)*PSIO*TERM*EXP(-BETA0*XI)
      + DUB *.5/(AK*H)*(B + OMEGA0*COS(ETA*ALPHA0)
      *EXP(-ALPHA0*XI))
      RETURN
      CONTINUE
      XP IS UPSTREAM OF AIRFOIL
      TERM = 0.
      IF (JET .NE. 0.) TERM = JET*ETA/(1.+F)
      ARG1 = PI - ALPHA1
      ARG2 = PI - BETA2
      PNEW = -.5*CIRCF*(1.-TERM-PSI2*SIN(ETA*ARG2)/ARG2*EXP(ARG2*XI))
      -.5*DUB / (AK*H)*(1.-B)*OMEGA1*COS(ETA*ARG1)*EXP(XI*ARG1)
      RETURN
      END

```

```

EMACH1 2
EMACH1 3
EMACH1 4
EMACH1 5
EMACH1 6
COM2 2
COM27 2
COM27 3
COM27 4
COM27 5
COM27 6
EMACH1 9
EMACH1 10
EMACH1 11
EMACH1 12
EMACH1 13
EMACH1 14
EMACH1 15
EMACH1 16
EMACH1 17
EMACH1 18
EMACH1 19
EMACH1 20
EMACH1 21
EMACH1 22
EMACH1 23
EMACH1 24
EMACH1 25
EMACH1 26
EMACH1 27
EMACH1 28
EXTRAP 2
EXTRAP 3
EXTRAP 4
EXTRAP 5
EXTRAP 6
COM2 2
COM12 2
COM12 3
COM15 3
COM16 2
COM16 3
COM16 4
COM28 2
COM28 3
EXTRAP 12
EXTRAP 13
EXTRAP 14
EXTRAP 15
EXTRAP 16
EXTRAP 17
EXTRAP 18
EXTRAP 19
EXTRAP 20
EXTRAP 21
EXTRAP 22
EXTRAP 23
EXTRAP 24
EXTRAP 25
EXTRAP 26
EXTRAP 27
EXTRAP 28
EXTRAP 29
EXTRAP 30
EXTRAP 31
EXTRAP 32
EXTRAP 33
EXTRAP 34
EXTRAP 35
EXTRAP 36
EXTRAP 37
EXTRAP 38
EXTRAP 39
EXTRAP 40
EXTRAP 41
EXTRAP 42
EXTRAP 43

```

```

SUBROUTINE FARFLD
      SUBROUTINE COMPUTES BOUNDARY DATA FOR OUTER
      BOUNDARIES.
      CALLED BY - TSFOIL.
      COMMON / COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
      ITE , JMIN , JMAX , JUP , JLOW ,
      JTOP , JBOT
      COMMON / COM2/ AK      , ALPHA , DUB , GAM1 , RTK
      LOGICAL      AMESH
      COMMON / COM4/ XIN(100) , YIN(100) , AMESH
      COMMON / COM12/ F      , H      , HALFPI , PI , RTKPOR ,
      TWOPI
      COMMON / COM15/ B      , BETA0 , BETA1 , BETA2 , PSIO ,
      PSI1 , PSI2
      REAL JET
      COMMON / COM16/ ALPHA1 , ALPHA2 , ALPHA3 , XSING ,
      OMEGA0 , OMEGA1 , OMEGA2 , JET
      COMMON / COM24/ DTOP(100) , DBOT(100) , DUP(100) , DDOWN(100) ,
      VTOP(100) , VBOT(100) , VUP(100) , VDOWN(100)
      INTEGER      BCTYPE
      COMMON / COM28/ BCTYPE , CIRCF , FHINV , POR , CIRCFE
      TEST FOR SUPERSONIC OR SUBSONIC FREESTREAM
      IF (AK .GT. 0.) GO TO 99
      IF (F .NE. 0.0 .AND. H .NE. 0.0) GO TO 10
      FHINV = 1.0
      RETURN
      10 CONTINUE
      FHINV = 1.0 / (F * H)
      SUPERSONIC FREESTREAM
      UPSTREAM BOUNDARY CONDITIONS CORRESPOND TO UNIFORM
      UNDISTURBED FLOW
      DOWNSTREAM BOUNDARY REQUIRED TO BE SUPERSONIC
      TOP AND BOTTOM BOUNDARIES USE SIMPLE WAVE SOLUTION.
      RETURN
      99 CONTINUE
      SUBSONIC FREESTREAM
      FUNCTIONAL FORM OF THE POTENTIAL ON OUTER BOUNDARIES
      IS PRESCRIBED. EQUATIONS REPRESENT ASYMPTOTIC FORM
      FOR DOUBLET AND VORTEX IN FREE AIR AND WIND TUNNEL
      ENVIRONMENT. DOUBLET AND VORTEX ARE LOCATED AT
      X*XSING, Y*0.
      ACTUAL BOUNDARY VALUES ARE SET IN SUBROUTINES RECIRC
      AND REDUB WHERE THE FUNCTIONAL FORMS ARE MULTIPLIED
      BY THE VORTEX AND DOUBLET STRENGTHS. THE BOUNDARY
      CONDITIONS ARE CALCULATED HEREIN FOR THE INPUT X AND
      Y MESH AND VALUES ARE DELETED FOR THE COARSE MESH IN
      SUBROUTINE SETBC.
      SET LOCATION OF SINGULAR VORTEX AND DOUBLET
      XSING = .5
      SET DEFAULT VALUES FOR TUNNEL WALL PARAMETERS
      B = 0.
      OMEGA0 = 1.
      OMEGA1 = 1.
      OMEGA2 = 1.
      JET = 0.
      PSIO = 1.
      PSI1 = 1.
      PSI2 = 1.
      BRANCH TO APPROPRIATE FORMULAS DEPENDING ON BCTYPE
      GO TO (100,200,300,400,500,600),BCTYPE
      CONTINUE
      BCTYPE = 1
      FREE AIR BOUNDARY CONDITION
      SET BOUNDARY ORDINATES
      YT = YIN(IMAX)*RTK
      YB = YIN(JMIN)*RTK
      XU = XIN(IMIN) - XSING
      XD = XIN(IMAX) - XSING
      YT2 = YT*YT
      YB2 = YB*YB
      XU2 = XU*XU
      XD2 = XD*XD
      COEF1 = 1./TWOPI
      COEF2 = 1.0/(TWOPI*RTK)
      COMPUTE DOUBLET AND VORTEX TERMS ON TOP AND BOTTOM
      BOUNDARIES.
      DO 110 I=IMIN,IMAX
      XP = XIN(I) - XSING
      XP2 = XP*XP
      DTOP(I) = XP/(XP2 + YT2)*COEF2

```

```

FARFLD 2
FARFLD 3
FARFLD 4
FARFLD 5
FARFLD 6
COM1 2
COM1 3
COM1 4
COM2 2
COM4 2
COM4 3
COM12 2
COM12 3
COM15 2
COM15 3
COM16 2
COM16 3
COM16 4
COM24 2
COM24 3
COM28 2
COM28 3
FARFLD 15
FARFLD 16
FARFLD 17
FARFLD 18
FARFLD 19
FARFLD 20
FARFLD 21
FARFLD 22
FARFLD 23
FARFLD 24
FARFLD 25
FARFLD 26
FARFLD 27
FARFLD 28
FARFLD 29
FARFLD 30
FARFLD 31
FARFLD 32
FARFLD 33
FARFLD 34
FARFLD 35
FARFLD 36
FARFLD 37
FARFLD 38
FARFLD 39
FARFLD 40
FARFLD 41
FARFLD 42
FARFLD 43
FARFLD 44
FARFLD 45
FARFLD 46
FARFLD 47
FARFLD 48
FARFLD 49
FARFLD 50
FARFLD 51
FARFLD 52
FARFLD 53
FARFLD 54
FARFLD 55
FARFLD 56
FARFLD 57
FARFLD 58
FARFLD 59
FARFLD 60
FARFLD 61
FARFLD 62
FARFLD 63
FARFLD 64
FARFLD 65
FARFLD 66
FARFLD 67
FARFLD 68
FARFLD 69
FARFLD 70
FARFLD 71
FARFLD 72
FARFLD 73
FARFLD 74
FARFLD 75
FARFLD 76
FARFLD 77
FARFLD 78

```

```

DDOT(I) = XP/(XP2 + YB2)*COEF2
VTOP(I) = -ATAN2(YT,XP)*COEF1
VBOT(I) = -ATAN2(YB,XP) + TWOPI)*COEF1
110 CONTINUE
      COMPUTE DOUBLET AND VORTEX TERMS ON UPSTREAM
      AND DOWNSTREAM BOUNDARIES
      DO 120 J=JMIN,JMAX
      YJ = YMIN(J)*RTK
      YJ2 = YJ*YJ
      DUP(J) = XU/(XU2 + YJ2)*COEF2
      DDOWN(J) = XD/(XD2 + YJ2)*COEF2
      O = PI - SIGN(PI,YJ)
      VUP(J) = -ATAN2(YJ,XU) + O)*COEF1
      VDOWN(J) = -ATAN2(YJ,XD) + O)*COEF1
120 CONTINUE
      IF (AK .GT. 0.0) CALL ANGLE
      RETURN
      COMPUTE WALL CONSTANTS FOR VARIOUS WIND TUNNEL
      CONDITIONS.
200 CONTINUE
      BCTYPE = 2
      SOLID WALL TUNNEL
      PDR = 0.
      SET CONSTANTS FOR DOUBLET SOLUTION
      B = .5
      ALPHAO = PI
      ALPHA1 = PI
      ALPHA2 = PI
      SET CONSTANTS FOR VORTEX SOLUTION
      BETA0 = HALFPI
      BETA1 = HALFPI
      BETA2 = HALFPI
      GO TO 700
300 CONTINUE
      BCTYPE = 3
      FREE JET
      F = 0.
      RTKPOR = 0.
      SET CONSTANTS FOR DOUBLET SOLUTION
      ALPHAO = HALFPI
      ALPHA1 = HALFPI
      ALPHA2 = HALFPI
      SET CONSTANTS FOR VORTEX SOLUTION
      JET = .5
      BETA0 = 0.
      BETA1 = 0.
      BETA2 = 0.
      GO TO 700
400 CONTINUE
      BCTYPE = 4
      IDEAL SLOTTED WALL
      RTKPOR = 0.
      FHINV = 1.0 / ( F * H )
      SET CONSTANTS FOR DOUBLET SOLUTION
      CALL DROOTS
      SET CONSTANTS FOR VORTEX SOLUTION
      JET = .5
      CALL VROOTS
      GO TO 700
500 CONTINUE
      BCTYPE = 5
      IDEAL PERFORATED/POROUS WALL
      F = 0.
      RTKPOR = RTK/PDR
      SET CONSTANTS FOR DOUBLET SOLUTION
      ALPHAO = HALFPI
      ALPHA1 = ALPHAO
      ALPHA2 = ALPHAO
      SET CONSTANTS FOR VORTEX SOLUTION
      BETA0 = ATAN(RTK/PDR)
      BETA1 = BETA0
      BETA2 = BETA1
      GO TO 700
600 CONTINUE
      BCTYPE = 6
      GENERAL HOMOGENEOUS WALL BOUNDARY CONDITION
      BOUNDARY CONDITION IS NOT OPERABLE YET IN FINITE
      DIFFERENCE SUBROUTINES.
      FAR FIELD SOLUTION HAS BEEN DERIVED AND IS INCLUDED
      HERE FOR FUTURE USE
      RTKPOR = RTK/PDR
      CALL DROOTS
      CALL VROOTS
      WRITE(6,1000)
1000 FORMAT(35H18RNORMAL STOP IN SUBRDUTINE FARFLD/
1 24H BCTYPE=6 IS NOT USEABLE)

```

```

FARFLD 79
FARFLD 80
FARFLD 81
FARFLD 82
FARFLD 83
FARFLD 84
FARFLD 85
FARFLD 86
FARFLD 87
FARFLD 88
FARFLD 89
FARFLD 90
FARFLD 91
FARFLD 92
FARFLD 93
FARFLD 94
FARFLD 95
FARFLD 96
FARFLD 97
FARFLD 98
FARFLD 99
FARFLD 100
FARFLD 101
FARFLD 102
FARFLD 103
FARFLD 104
FARFLD 105
FARFLD 106
FARFLD 107
FARFLD 108
FARFLD 109
FARFLD 110
FARFLD 111
FARFLD 112
FARFLD 113
FARFLD 114
FARFLD 115
FARFLD 116
FARFLD 117
FARFLD 118
FARFLD 119
FARFLD 120
FARFLD 121
FARFLD 122
FARFLD 123
FARFLD 124
FARFLD 125
FARFLD 126
FARFLD 127
FARFLD 128
FARFLD 129
FARFLD 130
FARFLD 131
FARFLD 132
FARFLD 133
FARFLD 134
FARFLD 135
FARFLD 136
FARFLD 137
FARFLD 138
FARFLD 139
FARFLD 140
FARFLD 141
FARFLD 142
FARFLD 143
FARFLD 144
FARFLD 145
FARFLD 146
FARFLD 147
FARFLD 148
FARFLD 149
FARFLD 150
FARFLD 151
FARFLD 152
FARFLD 153
FARFLD 154
FARFLD 155
FARFLD 156
FARFLD 157
FARFLD 158
FARFLD 159
FARFLD 160
FARFLD 161
FARFLD 162
FARFLD 163
FARFLD 164

```

```

700 STOP
      CONTINUE
      COMPUTE FUNCTIONAL FORMS FOR UPSTREAM AND DOWNSTREAM
      BOUNDARY CONDITIONS FOR DOUBLET AND VORTEX
      XU = (XIN(I)MIN) - XSING)/(RTK*H)
      XD = (XIN(I)MAX) - XSING)/(RTK*H)
      DOUBLET TERMS
      COEF1 = .5/AK/H
      ARG0 = ALPHA0
      ARG1 = PI - ALPHA1
      ARG2 = TWOPI - ALPHA2
      EXARG0 = EXP(-ARG0*XD)
      EXARG1 = EXP(ARG1*XU)
      EXARG2 = EXP(ARG2*XU)
      DO 720 J = JMIN,JMAX
      YJ = YMIN(J) / H
      DDOWN(J) = COEF1*(B + OMEGA0*COS(YJ*ARG0)*EXARG0)
      DUP(J) = -COEF1*(1.-B)*OMEGA1*COS(YJ*ARG1)*EXARG1 +
1 OMEGA2*COS(YJ*ARG2)*EXARG2)
720 CONTINUE
      VORTEX TERMS
      ARG0 = BETA0
      ARG1 = PI + BETA1
      ARG2 = PI - BETA2
      EXARG0 = EXP(-ARG0*XD)
      EXARG1 = EXP(-ARG1*XD)
      EXARG2 = EXP(ARG2*XU)
      DO 740 J = JMIN,JMAX
      YJ = YMIN(J) / H
      TERM = YJ
      IF (JET .EQ. 0.0) TERM = SIN(YJ*ARG0) / ARG0
      VDOWN(J) = -.5*(1. - SIGN(1.,YJ) + (1. - JET)*PSI0*TERM*EXARG0 +
1 PSII*SIN(YJ*ARG1)*EXARG1/ARG1)
      TERM = 0.
      IF (JET .NE. 0.0) TERM = JET * YJ / (1.0 + F)
      VUP(J) = -.5*(1. - TERM - PSII*SIN(YJ*ARG2)*EXARG2/ARG2)
740 CONTINUE
      RETURN
      END
      SUBROUTINE FINOSK(I,START,IEND,J,ISK)
      SUBROUTINE LOCATES SHOCK WAVE ON LINE J BETWEEN
      ISTART AND IEND. SHOCK IS LOCATED AT SHOCK POINT.
      IF NO SHOCK FOUND, ISK IS SET NEGATIVE.
      CALLED BY - CDDOLE.
      LOGICAL PHYS
      INTEGER PRTFLO , SIMDEF
      COMMON /COM27/ CL , DELTA , DELRT2 , EMACH , EMROOT ,
1 PHYS , PRTFLO , SIMDEF , SONVEL , VFACT ,
2 YFACT
      ISK = ISTART - 1
      U2 = PX(ISK,J)
      ISK = ISK + 1
      U1 = U2
      U2 = PX(ISK,J)
      IF(U1 .GT. SONVEL .AND. U2 .LE. SONVEL) GO TO 10
      IF (ISK .LE. IEND) GO TO 5
      ISK = - IEND
      CONTINUE
      RETURN
      END
      SUBROUTINE FIXPLT
      SETS UP ARRAYS FOR SUBROUTINE CPLOT.
      CALLED BY - PRINT.
      COMMON /COM1/ P(102,101),X(100) , Y(100)
      COMMON /COM1/ IMIN , IMAX , IUP , IODWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT
      COMMON /COM13/ CDFACT , CLFACT , CMFACT , CPFAC , CPSTAR
      COMMON /COM25/ CPL(100) , CPU(100)

```

```

FARFLD 165
FARFLD 166
FARFLD 167
FARFLD 168
FARFLD 169
FARFLD 170
FARFLD 171
FARFLD 172
FARFLD 173
FARFLD 174
FARFLD 175
FARFLD 176
FARFLD 177
FARFLD 178
FARFLD 179
FARFLD 180
FARFLD 181
FARFLD 182
FARFLD 183
FARFLD 184
FARFLD 185
FARFLD 186
FARFLD 187
FARFLD 188
FARFLD 189
FARFLD 190
FARFLD 191
FARFLD 192
FARFLD 193
FARFLD 194
FARFLD 195
FARFLD 196
FARFLD 197
FARFLD 198
FARFLD 199
FARFLD 200
FARFLD 201
FARFLD 202
FARFLD 203
      FINOSK 2
      FINOSK 3
      FINOSK 4
      FINOSK 5
      FINOSK 6
      COM27 2
      COM27 3
      COM27 4
      COM27 5
      COM27 6
      FINOSK 8
      FINOSK 9
      FINOSK 10
      FINOSK 11
      FINOSK 12
      FINOSK 13
      FINOSK 14
      FINOSK 15
      FINOSK 16
      FINOSK 17
      FINOSK 18
      FIXPLT 2
      FIXPLT 3
      FIXPLT 4
      FIXPLT 5
      FIXPLT 6
      BLANK 2
      COM1 2
      COM1 3
      COM1 4
      COM13 2
      COM25 2

```

APPENDIX

```

COMMON /COM30/ CPUP(101), CPLO(101), CPS(101), XP(101)
YMX = 5.*CPFACT
YMN = -5.*CPFACT
K = 0
DO 150 I = IMIN, IMAX
  K = K + 1
  QCP = -CPU(I)
  QCP = AMAX1(QCP, YMN)
  QCP = AMIN1(QCP, YMX)
  CPUP(K) = QCP
  QC1 = -CPL(I)
  QC1 = AMAX1(QC1, YMN)
  QC1 = AMIN1(QC1, YMX)
  CPLO(K) = QC1
  QC2 = -CPSTAR
  QC2 = AMAX1(QC2, YMN)
  QC2 = AMIN1(QC2, YMX)
  CPS(K) = QC2
  XP(K) = X(I)
150 CONTINUE
IMP = K + 1
CPUP(IMP) = YMX
CPLO(IMP) = YMN
CPS(IMP) = 0.0
XP(IMP) = XI(IMAX) + .001
CALL CPLDIT (XP, CPUP, CPLO, CPS, IMP)
RETURN
END

SUBROUTINE GUESSP
SUBROUTINE INITIALIZES P ARRAY AS FOLLOWS
PSTART = 1 P SET TO ZERO
PSTART = 2 P READ FROM UNIT 7
PSTART = 3 P SET TO VALUES IN CORE
IF PSTART = 2 OR 3, SOLUTION IS INTERPOLATED FROM
XOLD, YOLD TO X,Y
BOUNDARY CONDITIONS FOR P ON OUTER BOUNDARIES ARE
AUTOMATICALLY SET DURING INITIALIZATION
CALLED BY - TSP0IL.

COMMON / COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT ,
COMMON / COM2/ AK , ALPHA , DUB , GAM1 , RTK
LOGICAL ABORT
COMMON / COM3/ IREF , ABORT , ICUT , KSTEP
LOGICAL AMESH
COMMON / COM4/ XIN(100) , YIN(100) , AMESH
INTEGER PSTART
LOGICAL PSAVE
COMMON /COM11/ ALPHA , CLOLD , DELTAD , DUBO , EMACHO ,
1 IMNO , IMAXO , IMAXI , JMNO , JMAXO ,
2 JMAKI , PSAVE , PSTART , TITLE(8) , TITLEI(8) ,
3 VOLD , XOLD(100),YOLD(100)
COMMON /COM13/ CDFACT , CMFACT , CPFAC , CPSTAR
INTEGER BCTYPE
COMMON /COM28/ BCTYPE , CIRCFF , FMINV , POR , CIRCTE
COMMON /COM30/ PT(100) , REST(304)

BRANCH TO APPROPRIATE LOCATION
GO TO ( 200, 300, 400) , PSTART
200 CONTINUE
PSTART = 1
P SET TO ZERO
DO 205 I=1,101
DO 205 J=1,102
PT(I,J) = 0.
205 CONTINUE
DUB = 0.
CIRCFF = 0.
CIRCTE=0.0
RETURN
300 CONTINUE
PSTART = 2
P, X, Y ARRAYS READ FROM UNIT 7 IN SUBROUTINE READIN
TOGETHER WITH INFORMATION ABOUT OLD SOLUTION

```

```

FIXPLT 11
FIXPLT 12
FIXPLT 13
FIXPLT 14
FIXPLT 15
FIXPLT 16
FIXPLT 17
FIXPLT 18
FIXPLT 19
FIXPLT 20
FIXPLT 21
FIXPLT 22
FIXPLT 23
FIXPLT 24
FIXPLT 25
FIXPLT 26
FIXPLT 27
FIXPLT 28
FIXPLT 29
FIXPLT 30
FIXPLT 31
FIXPLT 32
FIXPLT 33
FIXPLT 34
FIXPLT 35
FIXPLT 36
FIXPLT 37
FIXPLT 38
FIXPLT 39

```

```

400 CONTINUE
PSTART = 3
ARRAYS FROM PREVIOUS CASE ARE ALREADY IN P,XOLD,YOLD
DUB = DUBO
CIRCFF = CLOLD/CLFACT
CIRCTE=CIRCFF
FOR PSTART = 2 OR 3, OLD P-ARRAY ON XOLD, YOLD MESH
MUST BE INTERPOLATED ONTO NEW X Y MESH.
INTERPOLATE P FROM XOLD,YOLD TO X,YOLD
CHECK TO SEE IF XOLD AND XIN ARE THE SAME MESH
IF(IMAXI .NE. IMAXO) GO TO 450
DO 410 I=IMIN, IMAXI
TEST = ABS(XIN(I) - XOLD(I))
IF(TEST .GT. .0001) GO TO 450
410 CONTINUE
XIN AND XOLD ARE SAME MESH.
P ARRAY MAY BE INTERPOLATED BY SIMPLE DELETION OF
VALUES AT MESH POINTS DELETED IN SUBROUTINE CUTOUT
IF IREF .LE. ZERO, NO INTERPOLATION IS NEEDED
IF(IREF .LE. 0) GO TO 500
ISTEP = 2*IREF
DO 430 J = JMINO, JMAXO
IMEW = 0
DO 420 I = IMINO, IMAXO, ISTEP
IMEW = IMEW + 1
P(I,J,IMEW) = P(I,J,I)
420 CONTINUE
430 CONTINUE
INTERPOLATION IN X DIRECTION COMPLETE IF XIN AND
XOLD ARE THE SAME.
GO TO 500
450 CONTINUE
INTERPOLATE FROM XOLD TO X FOR ARBITRARY CASE
DO 490 J = JMINO, JMAXO
YP = YOLD(J)
DO 475 I = IMIN, IMAX
XP = X(I)
IF(XP .LT. XOLD(IMINO)) GO TO 470
IF(XP .GT. XOLD(IMAXO)) GO TO 470
NEW X MESH POINT WITHIN RANGE OF OLD X MESH
FIND VALUE OF XOLD .GT. XP
XZ = XOLD(I)
K = 0
455 K = K + 1
X1 = XZ
X2 = XOLD(K)
IF (X2 .LT. XP) GO TO 455
IF (X2 .GT. XP) GO TO 460
GO TO 475
460 CONTINUE
P1 = P(J,K-1)
P2 = P(J,K)
PT(I) = P1 + (P2 - P1)/(X2 - X1) * (XP - X1)
GO TO 475
470 CONTINUE
NEW X MESH POINT IS OUTSIDE RANGE OF OLD X MESH
FOR SUPERSONIC FREESTREAM SET P=0, FOR SUBSONIC
FREESTREAM, EXTRAPOLATE USING FAR FIELD SOLUTION
PT(I) = 0.
IF(AK .GT. 0.) CALL EXTRAP(XP,YP,PT(I))
475 CONTINUE
WRITE NEW VALUES FOR P INTO P ARRAY
DO 480 I = IMIN, IMAX
P(I,J) = PT(I)
480 CONTINUE
490 CONTINUE
500 CONTINUE
INTERPOLATE FROM X,YOLD TO X,Y
CHECK TO SEE IF YIN AND YOLD ARE THE SAME MESH
IF(JMAXI .NE. JMAXO) GO TO 510
DO 510 J = JMIN, JMAXI
TEST = ABS(YIN(J) - YOLD(J))
IF(TEST .GT. .0001) GO TO 510
510 CONTINUE
YIN AND YOLD ARE THE SAME MESH
P ARRAY MAY BE INTERPOLATED BY SIMPLE DELETION OF
VALUES AT MESH POINTS DELETED IN SUBROUTINE CUTOUT
IF IREF .LE. ZERO, NO INTERPOLATION IS NEEDED
IF(IREF .LE. 0) GO TO 600
JSTEP = 2*IREF
DO 530 I = IMIN, IMAX

```

```

GUESSP 40
GUESSP 41
GUESSP 42
GUESSP 43
GUESSP 44
GUESSP 45
GUESSP 46
GUESSP 47
GUESSP 48
GUESSP 49
GUESSP 50
GUESSP 51
GUESSP 52
GUESSP 53
GUESSP 54
GUESSP 55
GUESSP 56
GUESSP 57
GUESSP 58
GUESSP 59
GUESSP 60
GUESSP 61
GUESSP 62
GUESSP 63
GUESSP 64
GUESSP 65
GUESSP 66
GUESSP 67
GUESSP 68
GUESSP 69
GUESSP 70
GUESSP 71
GUESSP 72
GUESSP 73
GUESSP 74
GUESSP 75
GUESSP 76
GUESSP 77
GUESSP 78
GUESSP 79
GUESSP 80
GUESSP 81
GUESSP 82
GUESSP 83
GUESSP 84
GUESSP 85
GUESSP 86
GUESSP 87
GUESSP 88
GUESSP 89
GUESSP 90
GUESSP 91
GUESSP 92
GUESSP 93
GUESSP 94
GUESSP 95
GUESSP 96
GUESSP 97
GUESSP 98
GUESSP 99
GUESSP 100
GUESSP 101
GUESSP 102
GUESSP 103
GUESSP 104
GUESSP 105
GUESSP 106
GUESSP 107
GUESSP 108
GUESSP 109
GUESSP 110
GUESSP 111
GUESSP 112
GUESSP 113
GUESSP 114
GUESSP 115
GUESSP 116
GUESSP 117
GUESSP 118
GUESSP 119
GUESSP 120
GUESSP 121
GUESSP 122
GUESSP 123
GUESSP 124

```

```

JNEW = 0
DD 520 J = JMIN, JMAX, JSTEP
JNEW = JNEW + 1
P(JNEW, I) = P(J, I)
520 CONTINUE
530 CONTINUE
      INTERPOLATION IN Y DIRECTION COMPLETE IF YIN AND
      YOLD ARE THE SAME.
C
GO TO 600
540 CONTINUE
      INTERPOLATE YOLD TO Y FOR ARBITRARY CASE
C
DD 590 I = IMIN, IMAX
XP = X(I)
K = 2
Y1 = YOLD(I)
DD 575 J = JMIN, JMAX
YP = Y(J)
IF(YP .LT. YOLD(JMIN)) GO TO 570
IF(YP .GT. YOLD(JMAX)) GO TO 571
      NEW Y MESH POINT WITHIN RANGE OF OLD Y MESH
      FIND VALUE OF YOLD .GT. YP
C
Y2 = Y1
K = K - 1
555 K = K + 1
Y1 = Y2
Y2 = YOLD(K)
IF(Y2 .LE. YP) GO TO 555
P1 = P(K-1, I)
P2 = P(K, I)
PT(J) = P1 + (P2 - P1) / (Y2 - Y1) * (YP - Y1)
GO TO 575
C
      NEW Y MESH POINT OUTSIDE RANGE OF OLD Y MESH.
      SET P(J, I) = P(JMIN, I) OR, IF SUBSONIC FREESTREAM
      FREE AIRFLOW, EXTRAPOLATE FOR P(J, I) USING FARFLD
      FORMULA.
C
570 CONTINUE
PT(J) = P(JMIN, I)
GO TO 572
571 PT(J) = P(JMAX, I)
572 CONTINUE
IF(AK .GT. 0. AND. BCTYPE .EQ. 1) CALL EXTRAP(XP, YP, PT(J))
575 CONTINUE
      PUT NEW P VALUES INTO P ARRAY
C
DD 580 J = JMIN, JMAX
P(J, I) = PT(J)
580 CONTINUE
590 CONTINUE
600 CONTINUE
RETURN
END

```

```

GUESSP 125
GUESSP 126
GUESSP 127
GUESSP 128
GUESSP 129
GUESSP 130
GUESSP 131
GUESSP 132
GUESSP 133
GUESSP 134
GUESSP 135
GUESSP 136
GUESSP 137
GUESSP 138
GUESSP 139
GUESSP 140
GUESSP 141
GUESSP 142
GUESSP 143
GUESSP 144
GUESSP 145
GUESSP 146
GUESSP 147
GUESSP 148
GUESSP 149
GUESSP 150
GUESSP 151
GUESSP 152
GUESSP 153
GUESSP 154
GUESSP 155
GUESSP 156
GUESSP 157
GUESSP 158
GUESSP 159
GUESSP 160
GUESSP 161
GUESSP 162
GUESSP 163
GUESSP 164
GUESSP 165
GUESSP 166
GUESSP 167
GUESSP 168
GUESSP 169
GUESSP 170
GUESSP 171
GUESSP 172
GUESSP 173
GUESSP 174

```

```

SUBROUTINE ISLIT ( X )
C
      ISLIT COMPUTES ILE AND ITE FOR ANY X MESH ARRAY.
      CALLED BY - CKMESH, CUTOUT, READIN, REFINE.
C
COMMON / COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT

DIMENSION X(100)
I = IMIN - 1
10 CONTINUE
I = I + 1
IF (X(I) .LT. 0.0) GO TO 10
ILE = I
20 CONTINUE
I = I + 1
IF (X(I) .LE. 1.0) GO TO 20
ITE = I - 1
RETURN
END

```

```

ISLIT 2
ISLIT 3
ISLIT 4
ISLIT 5
ISLIT 6
COM1 2
COM1 3
COM1 4
ISLIT 8
ISLIT 9
ISLIT 10
ISLIT 11
ISLIT 12
ISLIT 13
ISLIT 14
ISLIT 15
ISLIT 16
ISLIT 17
ISLIT 18
ISLIT 19
ISLIT 20

```

```

SUBROUTINE JSLIT ( Y )
C
      JSLIT COMPUTES JLOW AND JUP FROM Y ARRAY.
      CALLED BY - CKMESH, CUTOUT, READIN, REFINE.
C
COMMON / COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT

DIMENSION Y(100)
J = JMIN - 1
10 CONTINUE
J = J + 1
IF (Y(J) .LT. 0.0) GO TO 10
JLOW = J - 1
JUP = J
RETURN
END

```

```

JSLIT 2
JSLIT 3
JSLIT 4
JSLIT 5
JSLIT 6
COM1 2
COM1 3
COM1 4
JSLIT 8
JSLIT 9
JSLIT 10
JSLIT 11
JSLIT 12
JSLIT 13
JSLIT 14
JSLIT 15
JSLIT 16
JSLIT 17

```

```

SUBROUTINE INPERR ( I )
C
      FATAL ERROR IN INPUT. WRITE MESSAGE AND STOP.
      CALLED BY - READIN, SCALE.
C
IF (I .EQ. 1) WRITE (6, 901)
IF (I .EQ. 2) WRITE (6, 902)
IF (I .EQ. 3) WRITE (6, 903)
IF (I .EQ. 4) WRITE (6, 904)
IF (I .EQ. 5) WRITE (6, 905)
IF (I .EQ. 6) WRITE (6, 906)
IF (I .EQ. 7) WRITE (6, 907)
STOP
901 FORMAT(1H0, 5X, 45HIMAX OR JMAX IS GREATER THAN 100, NOT ALLOWED.)
902 FORMAT(1H0, 5X, 39HX MESH POINTS NOT MONOTONIC INCREASING.)
903 FORMAT(1H0, 5X, 39HY MESH POINTS NOT MONOTONIC INCREASING.)
904 FORMAT(1H0, 5X, 44HMACH NUMBER NOT IN PERMITTED RANGE. (.5, 2.0))
905 FORMAT(1H0, 5X, 41HALPHA NOT IN PERMITTED RANGE. (-9.0, 9.0))
906 FORMAT(1H0, 5X, 41HDELTA NOT IN PERMITTED RANGE. ( 0.0, 1.0))
907 FORMAT(1H0, 5X, 45HAK=0. VALUE OF AK MUST BE INPUT SINCE PHYS=F.)
END

```

```

INPERR 2
INPERR 3
INPERR 4
INPERR 5
INPERR 6
INPERR 7
INPERR 8
INPERR 9
INPERR 10
INPERR 11
INPERR 12
INPERR 13
INPERR 14
INPERR 15
INPERR 16
INPERR 17
INPERR 18
INPERR 19
INPERR 20
INPERR 21
INPERR 22

```

```

REAL FUNCTION CLIFT (CLFACT)
C
      COMPUTE LIFT COEFFICIENT FROM JUMP IN P AT TRAILING
      EDGE OF AIRFOIL
      CALLED BY - PRINT1, SOLVE.
C
COMMON P(102, 101), X(100) , Y(100)
COMMON / COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT
COMMON / COM7/ CJUP , CJUP1 , CJLOW , CJLOW1

PTOP = CJUP*P(JUP, ITE) - CJUP1*P(JUP + 1, ITE)
PBOT = CJLOW*P(JLOW, ITE) - CJLOW1*P(JLOW - 1, ITE)
LIFT = 2.*CLFACT*(PTOP - PBOT)
RETURN
END

```

```

LIFT 2
LIFT 3
LIFT 4
LIFT 5
LIFT 6
BLANK 2
COM1 2
COM1 3
COM1 4
COM7 2
LIFT 10
LIFT 11
LIFT 12
LIFT 13
LIFT 14
LIFT 15

```

```

SUBROUTINE MACHMP
SUBROUTINE TO PRINT MAP OF MACH NO. ROUNDED
TO NEAREST .1. CALLED BY = PRINT.

COMMON / COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
1 IIE , IJTE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT , JSDT
COMMON / COM30/ MM(100) , REST(304)

INTEGER CHAR
DIMENSION CHAR(37)
DATA CHAR/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HA,1HB,1HC,1HD,
1 1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT,
2 1HU,1HV,1HW,1HX,1HY,1HZ,1H*/
DATA IB/1H / , IP/1H+ / , IM/1- / , IL/1HL / , IT/1HT /

WRITE (6,400)
DO 60 K=2,JMAX
J = JMAX - K + 2
IJC = IB
IF ( J .EQ. JUP ) IJC = IP
IF ( J .EQ. JLOW ) IJC = IM
DO 10 I=1,IMAX
MM(I) = IB
10 CONTINUE
DO 50 I=2,IMAX
U = PX(I,J)
EM = EMACH1(U)
IF (EM .GT. 0.0) GO TO 20
MM(I) = CHAR(1)
20 CONTINUE
KAR=INT(10.*EM+1.5)
IF (KAR .GT. 37) KAR=37
MM(I)=CHAR(KAR)
30 CONTINUE
WRITE (6,410) IJC, (MM(I),I=2,IMAX)
40 CONTINUE
DO TO 1=1,IMAX
MM(I) = IB
IF (I .EQ. ILE) MM(I) = IL
IF (I .EQ. ITE) MM(I) = IT
70 CONTINUE
WRITE (6,420) (MM(I),I=2,IMAX)
RETURN

400 FORMAT(1H1,5HX,15HMACH NUMBER MAP/59X,15(1H*)///3X,4(6HSYMBOL,5X,
1 5HRANGE,14X)/5X,1HO,10X,9HM,LE=0.09,10X,20HA 0.95,LT,M,LE=1.05,
2 1OX,20HK 1.95,LT,M,LE=2.05,10X,20HU 2.95,LT,M,LE=3.05/5X,
3 20H1 0.05,LT,M,LE=0.15,10X,20HB 1.05,LT,M,LE=1.15,10X,
4 20HL 2.05,LT,M,LE=2.15,10X,20HV 3.05,LT,M,LE=3.15/5X,
5 20H2 0.15,LT,M,LE=0.25,10X,20HC 1.15,LT,M,LE=1.25,10X,
6 20HM 2.15,LT,M,LE=2.25,10X,20HH 3.15,LT,M,LE=3.25/5X,
7 20H3 0.25,LT,M,LE=0.35,10X,20HD 1.25,LT,M,LE=1.35,10X,
8 20HN 2.25,LT,M,LE=2.35,10X,20HE 3.25,LT,M,LE=3.35/5X,
9 20H4 0.35,LT,M,LE=0.45,10X,20HE 1.35,LT,M,LE=1.45,10X,
A 20HD 2.35,LT,M,LE=2.45,10X,20HY 3.35,LT,M,LE=3.45/5X,
B 20H5 0.45,LT,M,LE=0.55,10X,20HF 1.45,LT,M,LE=1.55,10X,
C 20HP 2.45,LT,M,LE=2.55,10X,20HZ 3.45,LT,M,LE=3.55/5X,
D 20H6 0.55,LT,M,LE=0.65,10X,20HG 1.55,LT,M,LE=1.65,10X,
E 20HQ 2.55,LT,M,LE=2.65,10X,1HP,10X,9HM,GT=3.55/5X,
F 20H7 0.65,LT,M,LE=0.75,10X,20HH 1.65,LT,M,LE=1.75,10X,
G 20HP 2.65,LT,M,LE=2.75/5X,20HS 0.75,LT,M,LE=0.85,10X,
H 20HI 1.75,LT,M,LE=1.85,10X,20HS 2.75,LT,M,LE=2.85/5X,
I 20H9 0.85,LT,M,LE=0.95,10X,20HJ 1.85,LT,M,LE=1.95,10X,
J 20HT 2.85,LT,M,LE=2.95//)

410 FORMAT(11X,100A1)
420 FORMAT (12X,99A1)

END

```

```

SUBROUTINE MILINE
PRINTS COORDINATES WHERE SONIC VELOCITY IS COMPUTED
LINEAR INTERPOLATION BETWEEN MESH POINTS IS USED
CALLED BY = PRINT.
*****

COMMON P(102,101),X(100) , Y(100)
COMMON / COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
1 IIE , IJTE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT , JSDT
COMMON / COM2/ AK , ALPHA , OUB , GAM1 , RTK
LOGICAL PHY5
INTEGER PRTFLO , SIMDEF
COMMON / COM27/ CL , DELTA , DELRT2 , EMACH , EMROOT ,
1 PHY5 , PRTFLO , SIMDEF , SONVEL , VFACT
2 YFACT
INTEGER BCTYPE
COMMON / COM28/ BCTYPE , CIRCFF , FHINV , PQR , CIRCTE
COMMON / COM30/ XSLPRT(200) , YSLPRT(200) , REST(4)

DIMENSION XSONIC(10)
NPTS = 0
KMIN = JMIN
KMAX = JMAX
JP = JMAX + JMIN
DO 200 K=KMIN,KMAX
J = JP - K
YPR = YFACT*(YJ)
PX2 = PX(IMIN,J)
H = 0
IF (J .NE. JLOW) GO TO 170
IF (NPTS .NE. 0) WRITE (6,2070)
170 CONTINUE
IMM = IMIN + 1
DO 180 I=IMM,IMAX
PX1 = PX2
PX2 = PX(I,J)
IF (PX1 .GT. SONVEL .AND. PX2 .GT. SONVEL) GO TO 180
IF (PX1 .LT. SONVEL .AND. PX2 .LT. SONVEL) GO TO 180
IF (NPTS .EQ. 0) WRITE (6,2000)
M = M+1
RATIO = (SONVEL-PX1)/(PX2-PX1)
XSONIC(M) = X(I-1) + (X(I)-X(I-1))*RATIO
NPTS = NPTS + 1
XSLPRT(NPTS) = XSONIC(M)
YSLPRT(NPTS) = YPR
IF (NPTS .GE. 200) GO TO 250
180 CONTINUE
IF (M .EQ. 0) GO TO 200
WRITE(6,2040) YPR,(XSONIC(L),L=1,M)
200 CONTINUE
GO TO 300
250 CONTINUE
WRITE(6,2060)
GO TO 400
300 CONTINUE
IF (NPTS .EQ. 0) GO TO 400
YM = Y(JMIN)
YX = Y(JMAX)
DO 320 N=1,NPTS
IF (YSLPRT(N) .NE. YM .AND. YSLPRT(N) .NE. YX) GO TO 320
IF ( AK .GT. 0.0 ) WRITE (6,2350)
IF ( AK .LT. 0.0 .AND. BCTYPE .EQ. 1 ) WRITE (6,2050)
320 CONTINUE
XMIN = -1.75
XMAX = 1.75
XINCR = .25
YMIN = -1.0
YMAX = 1.5
YINCR = .5
CALL PLTSON(XSLPRT,YSLPRT,XMIN,XMAX,XINCR,YMIN,YMAX,YINCR,NPTS)
CONTINUE
RETURN

2000 FORMAT(1H1,5HX,17HSONIC LINE OUTPUT/59X,17(1H*)/
1 23H SONIC LINE COORDINATES//6X,1HY,10X,7HX=SDNIC/)
2040 FORMAT(11F10.5)
2050 FORMAT(20H0***** CAUTION *****/
* 34H SONIC LINE HAS REACHED A BOUNDARY/
1 61H THIS VIOLATES ASSUMPTIONS USED TO DERIVE BOUNDARY CONDITIONS/
2 29H SOLUTION IS PROBABLY INVALID)
2060 FORMAT(20H0***** CAUTION *****/36H NUMBER OF SONIC POINTS EXCEEDED
1 200/25H ARRAY DIMENSION EXCEEDED/42H EXECUTION OF SUBROUTINE MILI
2NE TERMINATED)
2070 FORMAT(2X,13HBODY LOCATION)

END

```

```

MILLINE 2
MILLINE 3
MILLINE 4
MILLINE 5
MILLINE 6
MILLINE 7
BLANK 2
COM1 2
COM1 3
COM1 4
COM1 4
COM2 2
COM27 2
COM27 3
COM27 4
COM27 5
COM27 6
COM28 2
COM28 3
MILLINE 13
MILLINE 14
MILLINE 15
MILLINE 16
MILLINE 17
MILLINE 18
MILLINE 19
MILLINE 20
MILLINE 21
MILLINE 22
MILLINE 23
MILLINE 24
MILLINE 25
MILLINE 26
MILLINE 27
MILLINE 28
MILLINE 29
MILLINE 30
MILLINE 31
MILLINE 32
MILLINE 33
MILLINE 34
MILLINE 35
MILLINE 36
MILLINE 37
MILLINE 38
MILLINE 39
MILLINE 40
MILLINE 41
MILLINE 42
MILLINE 43
MILLINE 44
MILLINE 45
MILLINE 46
MILLINE 47
MILLINE 48
MILLINE 49
MILLINE 50
MILLINE 51
MILLINE 52
MILLINE 53
MILLINE 54
MILLINE 55
MILLINE 56
MILLINE 57
MILLINE 58
MILLINE 59
MILLINE 60
MILLINE 61
MILLINE 62
MILLINE 63
MILLINE 64
MILLINE 65
MILLINE 66
MILLINE 67
MILLINE 68
MILLINE 69
MILLINE 70
MILLINE 71
MILLINE 72
MILLINE 73
MILLINE 74
MILLINE 75
MILLINE 76
MILLINE 77
MILLINE 78
MILLINE 79
MILLINE 80
MILLINE 81

```

```

SUBROUTINE NEWISK(ISKOLD,J,ISKNEW)
C      FIND NEW LOCATION OF SHOCKWAVE (ISKNEW) ON LINE J
C      GIVEN AN INITIAL GUESS FOR LOCATION (ISKOLD).
C      SHOCK LOCATION IS DEFINED AS LOCATION OF SHOCK POINT.
C      IF NO SHOCK IS FOUND, ISKNEW IS SET NEGATIVE.
C      CALLED BY = CDCOLE.

LOGICAL      PHYS
INTEGER      PRFLD , SIMDEF
COMMON /COM27/ CL , DELTA , DELRTZ , EMACH , EPRODT ,
1            PHYS , PRFLD , SIMDEF , SONVEL , VFACT ,
2            YFACT
I2 = ISKOLD + 2
ISKNEW = ISKOLD - 3
U2 = #X(ISKNEW,J)
ISKNEW = ISKNEW + 1
U1 = U2
U2 = #X(ISKNEW,J)
IF(U1 .GT. SONVEL .AND. U2 .LE. SONVEL) GO TO 10
IF(ISKNEW .LT. I2) GO TO 5
C      NO SHOCK POINT FOUND, TIP OF SHOCK REACHED.
10 ISKNEW = - ISKNEW
RETURN
END

```

```

NEWISK 2
NEWISK 3
NEWISK 4
NEWISK 5
NEWISK 6
NEWISK 7
NEWISK 8
NEWISK 9
NEWISK 10
NEWISK 11
NEWISK 12
NEWISK 13
NEWISK 14
NEWISK 15
NEWISK 16
NEWISK 17
NEWISK 18
NEWISK 19
NEWISK 20
NEWISK 21

```

```

FUNCTION PITCH(CMFACT)
C      COMPUTE AIRFOIL PITCHING MOMENT ABOUT X = XM, Y=0
C      THE INTEGRAL OF(X-XM)(CPUPPER-CPLOWER) IS INTEGRATED
C      BY PARTS TO GIVE AN INTEGRAL IN THE JUMP IN P PLUS A
C      CONSTANT EQUAL TO THE JUMP IN P AT X = 1. TIMES THE
C      MOMENT ARM AT THE TRAILING EDGE
C      CALLED BY = PRINT1, SOLVE.

COMMON / COM1 / IMIN , IMAX , IUP , IDOWN , ILE ,
1           ITE , JMIN , JMAX , JUP , JLOW ,
2           JTOP , JBOT
COMMON / COM7 / CJUP , CJUP1 , CJLOW , CJLOW1
COMMON / COM30 / XI(100) , ARG(160) , REST(204)

C      SET XM TO QUARTER CHORD
XM = .25
K = 0
DO 10 I=ILE,ITE
K = K + 1
PTOP = CJUP*P(JUP,I) - CJUP1*P(JUP+1,I)
PBOT = CJLOW*P(JLOW,I) - CJLOW1*P(JLOW+1,I)
ARG(K) = PTOP - PBOT
XI(K) = X(I)
10 CONTINUE
CALL TRAP(XI,ARG,K,SUM)
PITCH = CMFACT*(1.-XM)*ARG(K) - SUM * (-2.)
RETURN
END

```

```

PITCH 2
PITCH 3
PITCH 4
PITCH 5
PITCH 6
PITCH 7
PITCH 8
PITCH 9
PITCH 0
BLANK 2
COM1 2
COM1 3
COM1 4
COM7 2
PITCH 13
PITCH 14
PITCH 15
PITCH 16
PITCH 17
PITCH 18
PITCH 19
PITCH 20
PITCH 21
PITCH 22
PITCH 23
PITCH 24
PITCH 25
PITCH 26
PITCH 27
PITCH 28

```

```

SUBROUTINE PLTSON (X,Y,XAXMIN,XMAX,XINCR,YAXMIN,YMAX,YINCR,NPTS)
C      THIS IS A MODIFIED VERSION OF PRNPLT AND
C      XMAX, XINCR, YMAX, YINCR MUST BE SUPPLIED.
C      XMIN, AND YMIN MUST ALSO BE SUPPLIED.
C      PRINTER PLOT ROUTINE M. S. ITZKOWITZ MAY, 1967
C      PLOTS THE SMPSS POINTS GIVEN BY, X(I),Y(I)
C      ON A 31 X 101 GRID USING A TOTAL OF 36 LINES ON
C      THE PRINTER.
C      IF EITHER INCREMENTAL STEP SIZE IS ZERO, THE
C      PROGRAM EXITS.
C      NEITHER OF THE INPUT ARRAYS ARE DESTROYED.
C      CALLED BY = MLINE.

DIMENSION X(NPTS), Y(NPTS), IGRID(105), XAXIS(11)
DIMENSION IBS(9)

```

```

PLTSON 2
PLTSON 3
PLTSON 4
PLTSON 5
PLTSON 6
PLTSON 7
PLTSON 8
PLTSON 9
PLTSON 10
PLTSON 11
PLTSON 12
PLTSON 13
PLTSON 14
PLTSON 15
PLTSON 16
PLTSON 17
PLTSON 18

```

```

INTEGER BLANK , DOT , STAR , DASH
DATA BLANK , DOT , STAR , DASH / 1H , 1H0 , 1H*, 1H-/
DATA IBS/1H0,1H0,1H0,1H*,1H , 1H*,1H1,1H1,1H1/

WRITE(6,900)
WRITE(6,910)
IF (XINCR .EQ. 0. .OR. YINCR .EQ. 0.) GO TO 800
YRNGE = YMAX - YAXMIN
XRNGE = XMAX - XAXMIN
YVAL = YRNGE * .02
XVAL = XRNGE * .01
JSLT = IFIX( 51.*(YMAX/YRNGE) ) + 1
OXV = 1.0 / XVAL
OYV = 1.0 / YVAL
IZERO = YMAX * OYV + 1.5
JZERO = 103.5 - XMAX * OXV
IF (JZERO .GT. 103 .OR. JZERO .LT. 4) JZERO = 2
FIZERO = IZERO
FJZERO = JZERO
IGRID(1) = BLANK
IGRID(2) = DOT
IGRID(104) = DOT
IGRID(105) = BLANK
FIZERS = FIZERO + .5
FJZERS = FJZERO + .5

C      NOPW IS THE NUMBER OF PRINT WHEELS USED TO
C      SPAN THE BODY LENGTH.
NOPW = OXV

C      JS = THE NUMBER OF PRT WHEEL POSITIONS TO BE
C      FILLED AT EACH END OF THE WORD +BODY SLIT+
JS = 0
IF (NOPW .GT. 9) JS = (NOPW-9) / 2

WRITE (6,905)
WRITE (6,903)

C      LOOP TO SET UP ONE LINE EACH TIME THRU.
DO 30 I=1,51
C      BLANK THE LINE TO BE PRINTED.
DO 2 J=3,103
IGRID(J) = BLANK
2 CONTINUE

C      SEARCH ARRAY FOR POINTS ON THIS LINE.
DO 10 K=1,NPTS
ITEST = FIZERS - Y(K) * OYV
IF (ITEST .NE. I) GO TO 7
J = FJZERS + X(K) * OXV
IF (J .GT. 103) J = 103
IF (J .LT. 3) J = 3
IGRID(J) = STAR
7 CONTINUE
IF (JSLT .EQ. I) GO TO 9
GO TO 10
9 CONTINUE

C      WRITE +BODY SLIT+ IF THIS IS THE JO LINE.
J = JZERO + 1
IF (JS .EQ. 0) GO TO 4
DO 3 L=1,JS
IF (IGRID(J) .NE. STAR) IGRID(J) = DASH
J = J + 1
3 CONTINUE
4 CONTINUE
DO 5 L=1,9
IF (IGRID(J) .NE. STAR) IGRID(J) = IBS(L)
J = J + 1
5 CONTINUE
IF (JS .EQ. 0) GO TO 10
DO 6 L=1,JS
IF (IGRID(J) .NE. STAR) IGRID(J) = DASH
J = J + 1
6 CONTINUE
10 CONTINUE
IF (MOD(I,10) .EQ. 1) GO TO 13
WRITE (6,901) IGRID
GO TO 30
13 CONTINUE
FI = I - 1
YAXIS = YMAX - FI * YINCR * .1
IF (ABS(YAXIS) .LT. YAXMIN) YAXIS = 0.
WRITE (6,902) YAXIS, (IGRID(J),J=1,105)
30 CONTINUE
WRITE (6,903)
WRITE (6,905)
DO 40 M=1,11

```

PLTSON 19
PLTSON 20
PLTSON 21
PLTSON 22
PLTSON 23
PLTSON 24
PLTSON 25
PLTSON 26
PLTSON 27
PLTSON 28
PLTSON 29
PLTSON 30
PLTSON 31
PLTSON 32
PLTSON 33
PLTSON 34
PLTSON 35
PLTSON 36
PLTSON 37
PLTSON 38
PLTSON 39
PLTSON 40
PLTSON 41
PLTSON 42
PLTSON 43
PLTSON 44
PLTSON 45
PLTSON 46
PLTSON 47
PLTSON 48
PLTSON 49
PLTSON 50
PLTSON 51
PLTSON 52
PLTSON 53
PLTSON 54
PLTSON 55
PLTSON 56
PLTSON 57
PLTSON 58
PLTSON 59
PLTSON 60
PLTSON 61
PLTSON 62
PLTSON 63
PLTSON 64
PLTSON 65
PLTSON 66
PLTSON 67
PLTSON 68
PLTSON 69
PLTSON 70
PLTSON 71
PLTSON 72
PLTSON 73
PLTSON 74
PLTSON 75
PLTSON 76
PLTSON 77
PLTSON 78
PLTSON 79
PLTSON 80
PLTSON 81
PLTSON 82
PLTSON 83
PLTSON 84
PLTSON 85
PLTSON 86
PLTSON 87
PLTSON 88
PLTSON 89
PLTSON 90
PLTSON 91
PLTSON 92
PLTSON 93
PLTSON 94
PLTSON 95
PLTSON 96
PLTSON 97
PLTSON 98
PLTSON 99
PLTSON 100
PLTSON 101
PLTSON 102
PLTSON 103

APPENDIX

```

FM = 11 - M
XAXIS(M) = XMAX - KINCR * FM
IF (XAXIS(M) .LT. XAXMIN) XAXIS(M) = XAXMIN
40 CONTINUE
WRITE (6,904) XAXIS, NPTS
RETURN

800 WRITE (6,9800)
CALL EXIT

900 FORMAT(1M1,58X,17HSONIC LINE OUTPUT/39X,17(1H*))
901 FORMAT(14X,109A1)
902 FORMAT(1X,F10.1,2X,1H*,105A1,1H*)
903 FORMAT(15X,103(1H*))
904 FORMAT(7X,11(F10.2),2H (,14,5H PTS) )
905 FORMAT(10X,11(1H*,9X))
910 FORMAT(35X,15HSONIC LINE PLOT,10X,6HY VS X,10X,
1 16H * = SONIC POINT/)
9600 FORMAT(46H1SCALING ERROR IN PRNPLT, EXECUTION TERMINATED )

END

```

```

SUBROUTINE PRBODY
PRINTS OUT BODY GEOMETRY
IF PHYS = .TRUE. ALL DIMENSIONS ARE NORMALIZED BY
AIRFOIL CHORD
IF PHYS = .FALSE. ALL DIMENSIONS EXCEPT X ARE
NORMALIZED BY CHORD LENGTH AND
THICKNESS RATIO.
CALLED BY - BODY.

COMMON / COM6 / FL(100) , FXL(100) , FU(100) , FXU(100) ,
CAMBER(100) , THICK(100) , VOL , XFOIL(100) , IFOIL
LOGICAL PHYS
INTEGER PRTFLO , SIMDEF
COMMON /COM27/ CL , DELTA , DELRT2 , EMACH , EMROOT ,
1 PHYS , PRTFLO , SIMDEF , SONVEL , VFACT ,
2 YFACT

WRITE(6,1000)
C FIND MAXIMUM THICKNESS AND CAMBER
THMAX = 0.
CAMAX = 0.
DO 10 I = 1, IFOIL
THMAX = AMAX1(THMAX, THICK(I))
CAMAX = AMAX1(CAMAX, CAMBER(I))
10 CONTINUE
THMAX = 2.*THMAX
IF (PHYS) GO TO 100
PRINTOUT IN SIMILARITY VARIABLES
WRITE(6,10G1) THMAX
WRITE(6,10G2) VOL, CAMAX
WRITE(6,10G3)
DO 40 I=1, IFOIL
WRITE (6,10G5) XFOIL(I), FU(I), FXU(I), FL(I), FXL(I), THICK(I) ,
1 CAMBER(I)
40 CONTINUE
RETURN
100 CONTINUE
PRINTOUT IN PHYSICAL VARIABLES
THMAX = DELTA*THMAX
CAMAX = DELTA*CAMAX
WRITE(6,10G6) THMAX
VOLUME = VOL*DELTA
WRITE(6,10G2) VOLUME, CAMAX
WRITE(6,10G4)
DO 110 I=1, IFOIL
YUP = DELTA*FU(I)
YXUP = DELTA*FXU(I)
YLO = DELTA*FL(I)
YXLO = DELTA*FXL(I)
TH = DELTA*THICK(I)
CA = DELTA*CAMBER(I)
WRITE(6,10G5) XFOIL(I), YUP, YXUP, YLO, YXLO, TH, CA)
110 CONTINUE
RETURN
1000 FORMAT(1M1,55X,23H AIRFOIL GEOMETRY OUTPUT/56X,23(1H*))
1001 FORMAT(33H PRINTOUT IN SIMILARITY VARIABLES,
* 48X, 15HMAX THICKNESS =,F12.8)

```

PLTSON 104
PLTSON 105
PLTSON 106
PLTSON 107
PLTSON 108
PLTSON 109
PLTSON 110
PLTSON 111
PLTSON 112
PLTSON 113
PLTSON 114
PLTSON 115
PLTSON 116
PLTSON 117
PLTSON 118
PLTSON 119
PLTSON 120
PLTSON 121
PLTSON 122
PLTSON 123
PLTSON 124

PRBODY 2
PRBODY 3
PRBODY 4
PRBODY 5
PRBODY 6
PRBODY 7
PRBODY 8
PRBODY 9
PRBODY 10
COM6 2
COM6 3
COM27 2
COM27 3
COM27 4
COM27 5
COM27 6
PRBODY 13
PRBODY 14
PRBODY 15
PRBODY 16
PRBODY 17
PRBODY 18
PRBODY 19
PRBODY 20
PRBODY 21
PRBODY 22
PRBODY 23
PRBODY 24
PRBODY 25
PRBODY 26
PRBODY 27
PRBODY 28
PRBODY 29
PRBODY 30
PRBODY 31
PRBODY 32
PRBODY 33
PRBODY 34
PRBODY 35
PRBODY 36
PRBODY 37
PRBODY 38
PRBODY 39
PRBODY 40
PRBODY 41
PRBODY 42
PRBODY 43
PRBODY 44
PRBODY 45
PRBODY 46
PRBODY 47
PRBODY 48
PRBODY 49
PRBODY 50
PRBODY 51
PRBODY 52
PRBODY 53

```

1002 FORMAT(16H AIRFOIL VOLUME=,F12.8,53X,15HMAX CAMBER =,F12.8//
1 2X,11HINPUT GRID,10X,14HUPPER SURFACE,17X,14HLOWER SURFACE)
1003 FORMAT(8X,1H,1X,2(15X,1HF,9X,5HDF/DX),11X,9HTHICKNESS,4X,
1 6HCAMBER//)
1004 FORMAT(8X,1H,1X,2(15X,1HY,9X,5HODY/DX),11X,9HTHICKNESS,4X,
1 6HCAMBER//)
1005 FORMAT(1X,F12.8,5X,2F12.8,2(6X,2F12.8))
1006 FORMAT(58H PRINTOUT IN PHYSICAL VARIABLES NORMALIZED BY CHORD LENG
1TH, 23X, 15HMAX THICKNESS =,F12.8)
END

```

```

SUBROUTINE PRINT
C SUBROUTINE FOR MAIN OUTPUT PRINT CONTROL.
C PRINTS RELATIVE PARAMETERS AND CALLS
C SPECIALIZED PRINT/PLOT SUBROUTINES AS REQUIRED.

C PRINTOUT DESCRIPTION
C IF PHYS IS .TRUE. THE FOLLOWING INFORMATION IS
C PRINTED IN PHYSICALLY SCALED VARIABLES (I.E. USUAL
C AERODYNAMIC TERMS)
C X,Y ARE COORDINATES NON-DIMENSIONALIZED BY
C AIRFOIL CHORD.
C CP = (P - P(INF))/Q(INF) = -2.*(DP/DX)
C CD = DRAG COEFF = D/Q(INF)*AIRFOIL CHORD
C CPSTAR = CRITICAL PRESSURE COEFFICIENT
C IF PHYS IS .FALSE. THE ABOVE INFORMATION IS PRINTED
C IN TRANSONIC SIMILARITY VARIABLES.

C PRINTOUT OF SONIC LINE ORDINATES IN SUBR. MILINE
C INCLUDING PRINTPLOT OF SONIC LINE ORDINATES
C CALLED BY - TSFOIL.
C *****PRINT 23*****
COMMON / COM2 / AK , ALPHA , DUB , GAM1 , RTK
LOGICAL ABORT
COMMON / COM3 / IREF , ABORT , ICUT , KSTEP
COMMON / COM6 / FL(100) , FXL(100) , FU(100) , FXU(100) ,
CAMBER(100) , THICK(100) , VOL , XFOIL(100) , IFOIL
1 INTEGER PSTART
LOGICAL PSAVE
COMMON /COM11/ ALPHAO , CLOLD , DELTAD , DUBD , EMACHO ,
IMINO , EMAXD , IMAXI , JMIND , JMAXO ,
2 JMAXI , PSAVE , PSTART , TITLE(8) , TITLEO(8) ,
3 VOLO , XOLO(100) , YOLO(100)
COMMON /COM13/ CDFACT , CLFACT , CMFACT , CPFAC , CPSTAR
LOGICAL FCR , KUTTA
COMMON /COM14/ CLSET , FCR , KUTTA , WCIRC
LOGICAL PHYS
INTEGER PRTFLO , SIMDEF
COMMON /COM27/ CL , DELTA , DELRT2 , EMACH , EMROOT ,
1 PHYS , PRTFLO , SIMDEF , SONVEL , VFACT ,
2 YFACT
INTEGER BCTYPE
COMMON /COM28/ BCTYPE , CIRCFF , PHINV , POR , CIRCTE

DIMENSION TPH(6) , SIM(8) , BCT(15) , FCP(14)
DATA TPH /4HSIMI,4HLARI,2HTY,4H PHY,4HSICA,2HL /
DATA SIM /4HCOLE,4H ,4HSPRE,4HITER,4HKRUP,4HP ,
1 4HUSER,4H /
DATA BCT /4HFREE,4H AIR,4H ,4HSOLI,4HD WA,4HLL ,4HFREE,
1 4H JET,4H ,4HSLOT,4HTED ,4HWALL,4HPORO,4HUS W,4HALL /
DATA FCP /4HFULL,4HY CO,4HNSER,4HVAT,4HVE ,4H ,4H ,
1 4HNOT ,4HCONS,4HERVA,4HTIVE,4H AT ,4HSHOC,4HK. /

WRITE (6,900)
IS = 1
IF (PHYS) IS = 4
IE = IS + 2
WRITE (6,901) (TPH(I), I=IS, IE)
IE = 2 * SIMDEF
IS = IE - 1
WRITE (6,902) (SIM(I), I=IS, IE)
IE = 3 * BCTYPE
IS = IE - 2
WRITE (6,903) (BCT(I), I=IS, IE)
IS = 8
IF (FCR) IS = 1

```

PRBODY 54
PRBODY 55
PRBODY 56
PRBODY 57
PRBODY 58
PRBODY 59
PRBODY 60
PRBODY 61
PRBODY 62
PRBODY 63
PRINT 2
PRINT 3
PRINT 4
PRINT 5
PRINT 6
PRINT 7
PRINT 8
PRINT 9
PRINT 10
PRINT 11
PRINT 12
PRINT 13
PRINT 14
PRINT 15
PRINT 16
PRINT 17
PRINT 18
PRINT 19
PRINT 20
PRINT 21
PRINT 22
PRINT 23
PRINT 24
COM2 2
COM3 2
COM3 3
COM6 2
COM6 3
COM11 2
COM11 3
COM11 4
COM11 5
COM11 6
COM11 7
COM13 2
COM14 2
COM14 3
COM27 2
COM27 3
COM27 4
COM27 5
COM28 2
COM28 3
PRINT 33
PRINT 34
PRINT 35
PRINT 36
PRINT 37
PRINT 38
PRINT 39
PRINT 40
PRINT 41
PRINT 42
PRINT 43
PRINT 44
PRINT 45
PRINT 46
PRINT 47
PRINT 48
PRINT 49
PRINT 50
PRINT 51
PRINT 52
PRINT 53
PRINT 54
PRINT 55
PRINT 56


```

IE = IS + 6
WRITE (6,9C4) (FCP(I),I=IS,IE)
IF (KUTTA) GO TO 10
WRITE (6,905)
GO TO 20
10 CONTINUE
WRITE (6,906)
20 CONTINUE
ALPHVF = ALPHA * VFACT
WRITE (6,911)
IF ( PHYS ) WRITE (6,909) EMACH, DELTA
WRITE (6,907) ALPHVF, AK
DUBDEL=DUB*DELTA
VOLDEL=VOL*DELTA
IF (AK .GT. 0.) WRITE(6,909) DUBDEL,VOLDEL
IF ( PHYS ) WRITE(6,910)CPFACT,CDFACT,CMFACT,CLFACT,YFACT,VFACT
CALL PRINT1
CALL PRINC
CALL MACHMP
IF ( ABORT ) GO TO 60
CALL FIXPLT
IF (BCTYPE .EQ. 1) GO TO 40
IF (BCTYPE .EQ. 3) GO TO 40
CALL PRYAL
40 CONTINUE
CALL MLINE
IF (PRTFLO .EQ. 1) GO TO 50
CALL PRFLO
50 CONTINUE
CALL CDCOLE
60 CONTINUE
RETURN
C
900 FORMAT(1H1,5X,18(1H*)/59X,1H*,16X,1H*/59X,18H* FINAL OUTPUT */
1 59X,1H*,16X,1H*/59X,18(1H*)
901 FORMAT(14H0 PRINTOUT IN 2A4,A2,11H VARIABLES.)
902 FORMAT(41H0 DEFINITION OF SIMILARITY PARAMETERS BY 2A4)
903 FORMAT(25H0 BOUNDARY CONDITION FOR 3A4)
904 FORMAT(27H0 DIFFERENCE EQUATIONS ARE 7A4)
905 FORMAT(37H0 LIFT COEFFICIENT SPECIFIED BY USER.)
906 FORMAT(30H0 KUTTA CONDITION IS ENFORCED.)
907 FORMAT(13X,7HALPHA =,F12.7/17X,3HK =,F12.7)
908 FORMAT(12X,18H0DOUBLET STRENGTH =,F12.7/4X,16HAIRFOIL VOLUME =,
1 F12.7)
909 FORMAT(14X,6HMACH =,F12.7/13X,7HDELTA =,F12.7)
910 FORMAT(1H0,8X,38HPARAMETERS USED TO TRANSFORM VARIABLES/
1 18X,20H0 TRANSMIC SCALING/12X,8HCPFACT =F12.7/
2 12X,8HCDFACT =,F12.7/12X,8HCMFACT =,F12.7/12X,8HCLFACT =,F12.7
3 /13X,7HVFACT =,F12.7/ 13X,7HVFACT =F12.7)
911 FORMAT(1H0)
END
SUBROUTINE PRINT1
C
C PRINTS PRESSURE COEFFICIENT AND MACH NUMBER
C ON Y=0 LINE, AND PLOTS CP ALONG SIDE OF PRINT.
C CALLED BY = ISFOLL, PRINT.
COMMON /COM1/ P(102,101),X(100) , Y(100)
COMMON / COM1 / IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT
LOGICAL
COMMON / COM3 / IREF , ABORT , ICUT , KSTEP
COMMON / COM7 / CJUP , CJUP1 , CJLW , CJLW1
COMMON / COM13 / CDFACT , CLFACT , CMFACT , CPFACT , CPSTAR
COMMON / COM25 / CPL(100) , CPU(100)
LOGICAL
INTEGER
COMMON / COM27 / CL , SIMDEF , DELTA , DELRT2 , EMACH , EMROOT ,
1 PHYS , PRFLO , SIMDEF , SONVEL , VFACT ,
2 YFACT
COMMON / COM30 / EML(100), EMLU(100), YM(100), REST(104)
REAL LIFT
DIMENSION LINE1(60) , THAC(2)
DATA THAC / 2H1/, 2H1/, 1B /1H /, 1L / 1HL/,1U /1HU/, 1S/1H*/
DATA 188/ 1H8/

```

```

PRINT 57
PRINT 58
PRINT 59
PRINT 60
PRINT 61
PRINT 62
PRINT 63
PRINT 64
PRINT 65
PRINT 66
PRINT 67
PRINT 68
PRINT 69
PRINT 70
PRINT 71
PRINT 72
PRINT 73
PRINT 74
PRINT 75
PRINT 76
PRINT 77
PRINT 78
PRINT 79
PRINT 80
PRINT 81
PRINT 82
PRINT 83
PRINT 84
PRINT 85
PRINT 86
PRINT 87
PRINT 88
PRINT 89
PRINT 90
PRINT 91
PRINT 92
PRINT 93
PRINT 94
PRINT 95
PRINT 96
PRINT 97
PRINT 98
PRINT 99
PRINT 100
PRINT 101
PRINT 102
PRINT 103
PRINT 104
PRINT 105
PRINT 106
PRINT 107
PRINT 108
PRINT 12
PRINT 13
PRINT 14
PRINT 15
PRINT 16
PRINT 17
PRINT 18
PRINT 19
PRINT 20
PRINT 21
PRINT 22

```

```

CL = LIFT (CLFACT)
CM = PITCH (CMFACT)
CPMIN = 1.0E37
CPMAX = -CPMIN
IEM = 0
CJ01 = -Y(JLOW)/(Y(JUP)-Y(JLOW))
CJ02 = Y(JUP)/(Y(JUP)-Y(JLOW))
DO 10 I=IMIN,IMAX
UL = CJLW*PX(I,JLOW) - CJLW1*PX(I,JLOW-1)
IF (I .GT. ITE) UL=CJ01*PX(I,JUP) + CJ02*PX(I,JLOW)
IF (I .LT. ILE) UL=CJ01*PX(I,JUP) + CJ02*PX(I,JLOW)
CPL(I) = -2.0 * UL * CPFACT
EML(I) = EMACH(UL)
IF (EML(I) .GT. 1.3) IEM = 1
UU = CJUP *PX(I,JUP) - CJUP1 *PX(I,JUP+1)
IF (I .GT. ITE) UU = UL
IF (I .LT. ILE) UU = UL
CPU(I) = -2.0 * UU * CPFACT
EMLU(I) = EMACH(UU)
IF (EMLU(I) .GT. 1.3) IEM = 1
CPMAX = AMAX1(CPMAX,CPU(I), CPL(I))
CPMIN = AMIN1(CPMIN,CPU(I), CPL(I))
10 CONTINUE
CPLARG = AMAX1(CPMAX,ABS(CPMIN))
UNPCOL = CPLARG / 29.
C LOCATE CP* FOR PRINTER PLOT
COL = -CPSTAR / UNPCOL
NCOL = SIGN((ABS(COL)+.5), COL)
NCOLS = NCOL + 30
C PRINT SINGLE VARIABLES
IF (IREF .EQ. 2) WRITE (6,270)
IF (IREF .EQ. 1) WRITE (6,280)
IF (IREF .EQ. 0) WRITE (6,290)
WRITE (6,260)
WRITE (6,200) CL, CM, CPSTAR
IF (CPL(IMIN) .LT. CPSTAR .AND. CPL(IMIN+1) .GT. CPSTAR) GO TO 70
WRITE (6,210)
KT = 2
IF ( PHYS ) KT = 1
C PRINT COLUMN HEADERS.
WRITE (6,220) THAC(KT) , THAC(KT)
DO 40 I=IMIN,IMAX
DO 30 K=1,60
LINE1(K) = IB
30 CONTINUE
COL = -CPU(I) / UNPCOL
NCOL = SIGN((ABS(COL) + .5) , COL)
NCOLU = NCOL + 30
LINE1(NCOLU) = IU
COL = -CPL(I) / UNPCOL
NCOL = SIGN((ABS(COL) + .5) , COL)
NCOLL = NCOL + 30
LINE1(NCOLL) = IL
IF (NCOLL .EQ. NCOLU) LINE1(NCOLL) = 18B
IF (IABS(NCOLS) .LT. 61) LINE1(NCOLS) = IS
IF (I .EQ. ILE) WRITE (6,230)
WRITE (6,250) I, X(I), CPL(I), EML(I), CPU(I), EMLU(I), LINE1
IF (I .EQ. ITE) WRITE (6,240)
40 CONTINUE
IF ( IEM .EQ. 0 ) GO TO 50
IF ( PHYS ) WRITE (6,300)
50 CONTINUE
DO 60 J=JMIN,JMAX
YM(J) = Y(J) * YFACT
60 CONTINUE
WRITE (6,310) JMIN,JMAX,(YM(J),J=JMIN,JMAX)
RETURN
70 CONTINUE
WRITE (6,320)
ABORT = .TRUE.
RETURN

```

```

PRINT1 23
PRINT1 24
PRINT1 25
PRINT1 26
PRINT1 27
PRINT1 28
PRINT1 29
PRINT1 30
PRINT1 31
PRINT1 32
PRINT1 33
PRINT1 34
PRINT1 35
PRINT1 36
PRINT1 37
PRINT1 38
PRINT1 39
PRINT1 40
PRINT1 41
PRINT1 42
PRINT1 43
PRINT1 44
PRINT1 45
PRINT1 46
PRINT1 47
PRINT1 48
PRINT1 49
PRINT1 50
PRINT1 51
PRINT1 52
PRINT1 53
PRINT1 54
PRINT1 55
PRINT1 56
PRINT1 57
PRINT1 58
PRINT1 59
PRINT1 60
PRINT1 61
PRINT1 62
PRINT1 63
PRINT1 64
PRINT1 65
PRINT1 66
PRINT1 67
PRINT1 68
PRINT1 69
PRINT1 70
PRINT1 71
PRINT1 72
PRINT1 73
PRINT1 74
PRINT1 75
PRINT1 76
PRINT1 77
PRINT1 78
PRINT1 79
PRINT1 80
PRINT1 81
PRINT1 82
PRINT1 83
PRINT1 84
PRINT1 85
PRINT1 86
PRINT1 87
PRINT1 88
PRINT1 89
PRINT1 90
PRINT1 91
PRINT1 92
PRINT1 93
PRINT1 94
PRINT1 95
PRINT1 96
PRINT1 97
PRINT1 98
PRINT1 99
PRINT1 100
PRINT1 101
PRINT1 102
PRINT1 103
PRINT1 104
PRINT1 105
PRINT1 106
PRINT1 107
PRINT1 108

```

APPENDIX

```

200 FORMAT(85X,28HB = BEFORE OR BEHIND AIRFOIL/10X,4HCL =,F10.6,61X,
1 25HL = UPPER AIRFOIL SURFACE/10X,4HCM =,F10.6,61X,
2 25HL = LOWER AIRFOIL SURFACE/10X,4HCP =,F10.6,61X,
3 21HP = CRITICAL PRESSURE)
210 FORMAT(140,27X,5HLOWER,23X,5HUPPER/28X,4HY=0,-24X,4HY=0+)
220 FORMAT(3X,1H1,8X,1HX,10X,2HCP,10X,A2,14X,2HCP,10X,A2/)
230 FORMAT(25X,20HAIRFOIL LEADING EDGE,49X,20HAIRFOIL LEADING EDGE)
240 FORMAT(25X,21HAIRFOIL TRAILING EDGE,44X,21HAIRFOIL TRAILING EDGE)
250 FORMAT(1H ,13,3F12.6,4X,2F12.6,2X,60A1)
260 FORMAT(60H FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUM
1BER /2X,59HGR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM L
2INE.)
270 FORMAT(1H1,52X,28HFINAL OUTPUT FOR COARSE MESH//)
280 FORMAT(1H1,52X,28HFINAL OUTPUT FOR MEDIUM MESH//)
290 FORMAT(1H1,53X,26HFINAL OUTPUT FOR FINE MESH//)
300 FORMAT(20H0***** CAUTION *****/
1 32H MAXIMUM MACH NUMBER EXCEEDS 1.3/
2 69H SHOCK JUMPS IN ERROR IF UPSTREAM NORMAL MACH NUMBER GREATER T
3HAN 1.3)
310 FORMAT(///16X,13HY=GRID VALUES,5X,7HY(J) J=,I4,3H TO,I4/
1 (10X,8F12.6))
320 FORMAT(1H0,///,60H DETACHED SHOCK WAVE UPSTREAM OF X=MESH,SOLUTION
1TERMINATED.)

END

SUBROUTINE PRFLD
PRINTS PRESSURE COEFFICIENT, FLOW ANGLE AND
MACH NUMBER IN FLOW FIELD. NUMBER OF J LINES
PRINTED IS DETERMINED FROM THE INPUT VALUE OF
PRFLD.
PRFLD = 1, NONE.
PRFLD = 2, ALL J LINES EXCEPT JO.
PRFLD = 3, THREE J LINES AROUND JERROR.
CALLED BY - PRINT.

COMMON P(102,101),X(100), Y(100)
COMMON / COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT
COMMON /COM13/ CDFACT , CLFACT , CMFACT , CPFAC , CPSTAR
LOGICAL OUTERR
COMMON /COM18/ ERROR , I1 , I2 , IERROR , JERROR ,
1 OUTERR , EMU(100,2) , VC(100) ,
2 WT , DCIRC , POLD(100,2)
LOGICAL PHYS
INTEGER PRFLD , SINDEF
COMMON /COM27/ CL , DELTA , DELRT2 , EMACH , EMROOT ,
1 PHYS , PRFLD , SINDEF , SONVEL , VFAC ,
2 YFACT
COMMON /COM30/ JLIN(100), REST(304)

DIMENSION THAC(2), CPPR(3), PYPR(3), YPRINT(3), PRT(10), EM1(3)
DATA THAC / 2HM1,2HK1/
DATA PRT / 4HACH+H NUM,4HBERS,4H ,4H ,
1 4HSIM,4HLAI,4HTY P,4HARAM,4HETER/
IF (PRFLD.EQ. 2) GO TO 25
LOCATE LINES AROUND JERROR.
JL = 3
IF (JERROR .EQ. JMIN) GO TO 10
IF (JERROR .EQ. JLOW) GO TO 20
IF (JERROR .EQ. JUP) GO TO 13
IF (JERROR .EQ. JMAX) GO TO 20
JLIN(1) = JERROR - 1
JLIN(2) = JERROR
JLIN(3) = JERROR + 1
GO TO 30
10 CONTINUE
JLIN(1) = JERROR
JLIN(2) = JERROR + 1
JLIN(3) = JERROR + 2
GO TO 30
20 CONTINUE
JLIN(1) = JERROR - 2
JLIN(2) = JERROR - 1
JLIN(3) = JERROR
GO TO 30

```

```

PRINT1 109
PRINT1 110
PRINT1 111
PRINT1 112
PRINT1 113
PRINT1 114
PRINT1 115
PRINT1 116
PRINT1 117
PRINT1 118
PRINT1 119
PRINT1 120
PRINT1 121
PRINT1 122
PRINT1 123
PRINT1 124
PRINT1 125
PRINT1 126
PRINT1 127
PRINT1 128
PRINT1 129
PRINT1 130
PRINT1 131
PRINT1 132
PRINT1 133

```

```

PRFLD 2
PRFLD 3
PRFLD 4
PRFLD 5
PRFLD 6
PRFLD 7
PRFLD 8
PRFLD 9
PRFLD 10
PRFLD 11
PRFLD 12
BLANK 2
COM1 2
COM1 3
COM1 4
COM13 2
COM18 2
COM18 3
COM18 4
COM18 5
COM27 2
COM27 3
COM27 4
COM27 5
COM27 6
PRFLD 18
PRFLD 19
PRFLD 20
PRFLD 21
PRFLD 22
PRFLD 23
PRFLD 24
PRFLD 25
PRFLD 26
PRFLD 27
PRFLD 28
PRFLD 29
PRFLD 30
PRFLD 31
PRFLD 32
PRFLD 33
PRFLD 34
PRFLD 35
PRFLD 36
PRFLD 37
PRFLD 38
PRFLD 39
PRFLD 40
PRFLD 41
PRFLD 42
PRFLD 43
PRFLD 44

```

```

25 CONTINUE
JL = JMAX - JMIN + 1
FILL JLIN ARRAY WITH VALUES OF J.
K = 1
DO 28 J=JMIN,JMAX
JLIN(K) = J
K = K + 1
28 CONTINUE
30 CONTINUE

C
C PRINT FLOW FIELD IN 3 J LINES PER PAGE.
DO 100 MPR = 1,JL,3
MPREND = MINO(MPR+2,JL)
DO 40 M=MPR,MPREND
MQ = M-MPR+1
JQ = JLIN(M)
YPRINT(MQ) = Y(JQ)*YFACT
40 CONTINUE
WRITE PAGE HEADER.
IS = 1
IF ( PHYS ) IS = 6
IE = IS + 4
WRITE (6,900) (PRT(I),I=IS,IE) , CPSTAR
WRITE(6, 901) (JLIN(M),M=MPR,MPREND)
WRITE(6, 902)(YPRINT(M),M=1,MQ)
KT = 2
IF ( PHYS ) KT = 1
WRITE (6,903) THAC(KT), THAC(KT), THAC(KT)
DO 60 I=IMIN,IMAX
DO 50 M=MPR,MPREND
MQ = M-MPR+1
J = JLIN(M)
U = PX(I,J)
CPPR(MQ) = -2.0 + CPFAC * U
PYPR(MQ) = VFAC*PY(I,J)
EM1(MQ) = EMACH(IU)
50 CONTINUE
WRITE (6,904) (I,X(I)),(CPPR(M),PYPR(M),EM1(M)),M=1,MQ)
60 CONTINUE
100 CONTINUE
RETURN

900 FORMAT(47H1PRESSURE COEFFICIENTS, FLOW ANGLES, AND LOCAL ,9A4/
1 20H ON Y=CONSTANT LINES/9H CPSTAR =,F12.7//)
901 FORMAT(13X,3(15X,2HJ=,I4,15X))
902 FORMAT(13X,3(12X,2HY=,F10.6,12X))
903 FORMAT(4HD I,8X,1HX,5X,3(6X,2HCP,8X,5HTHETA,7X,A2,6X)//)
904 FORMAT(1X,I3,2X,F10.6,1X,3(2X,3F11.6,1X))

END

SUBROUTINE PRMTC
SUBROUTINE TO PRINT A CHARACTER FOR EACH
POINT IN THE GRID DESCRIBING THE TYPE OF
FLOW. S , SHOCK POINT
H , HYPERBOLIC POINT
P , PARABOLIC POINT
- , ELLIPTIC POINT
CALLED BY - PRINT.

COMMON P(102,101),X(100), Y(100)
COMMON / COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT
COMMON / COM2/ AK , ALPHA , DUB , GAM1 , RTK
COMMON /COM22/ CXK(100), CXL(100), CXK(100), CXXC(100),CXKL(100),
1 CXXR(100), C1(100)

COMMON /COM30/ IPC(100) , VT(100,2), REST(104)
DATA [HP/IHP/, IHH/IHH/, IHS/IHS/, IHD/IH-/, I8/IH /

WRITE (6,100)
DO 5 I=1,50,2
IPC(I) = 18
IPC(I+1) = 18
5 CONTINUE
DO 10 J=JMIN,JMAX
VT(J,1) = C1(2)
10 CONTINUE

```

```

PRFLD 45
PRFLD 46
PRFLD 47
PRFLD 48
PRFLD 49
PRFLD 50
PRFLD 51
PRFLD 52
PRFLD 53
PRFLD 54
PRFLD 55
PRFLD 56
PRFLD 57
PRFLD 58
PRFLD 59
PRFLD 60
PRFLD 61
PRFLD 62
PRFLD 63
PRFLD 64
PRFLD 65
PRFLD 66
PRFLD 67
PRFLD 68
PRFLD 69
PRFLD 70
PRFLD 71
PRFLD 72
PRFLD 73
PRFLD 74
PRFLD 75
PRFLD 76
PRFLD 77
PRFLD 78
PRFLD 79
PRFLD 80
PRFLD 81
PRFLD 82
PRFLD 83
PRFLD 84
PRFLD 85
PRFLD 86
PRFLD 87
PRFLD 88
PRFLD 89
PRFLD 90
PRFLD 91
PRFLD 92
PRFLD 93
PRFLD 94

```

```

PRMTC 2
PRMTC 3
PRMTC 4
PRMTC 5
PRMTC 6
PRMTC 7
PRMTC 8
PRMTC 9
PRMTC 10
BLANK 2
COM1 2
COM1 3
COM1 4
COM2 2
COM22 2
COM22 3
PRMTC 15
PRMTC 16
PRMTC 17
PRMTC 18
PRMTC 19
PRMTC 20
PRMTC 21
PRMTC 22
PRMTC 23
PRMTC 24
PRMTC 25
PRMTC 26

```

```

DO 60 K=JMIN,JMAX
J = JMAX - K + 1
DO 50 I=IUP, IDOWN
VT(J,2) = VT(J,1)
VT(J,1) = C1(I) - CXL(I)*P(J,I-1) + CXC(I)*P(J,I) + CXR(I)*P(J,I+1)
IF (VT(J,1) .GT. 0.0) GO TO 30
IF (VT(J,2) .LT. 0.0) GO TO 20
C
      PARABOLIC POINT (SONIC)
IPC(I) = IHP
GO TO 30
20 CONTINUE
      HYPERBOLIC POINT (SUPERSONIC)
IPC(I) = IHM
GO TO 30
30 CONTINUE
IF (VT(J,2) .LT. 0.0) GO TO 40
C
      ELLIPTIC POINT (SUBSONIC)
IPC(I) = IHD
GO TO 50
40 CONTINUE
      SHOCK POINT
IPC(I) = IHS
50 CONTINUE
WRITE (6,110) J, (IPC(I),I=IUP, IDOWN)
60 CONTINUE
RETURN
100 FORMAT(1H1,30X,10HFLOW CHARACTER MAP/57X,18(1H*)//28X,
1 13HP = PARABOLIC/28X,14HH = HYPERBOLIC/28X,9HS = SHOCK/28X,
2 12H = ELLIPTIC//1
110 FORMAT(10X,13,5X,100A1)
END

```

```

PRTMC 27
PRTMC 28
PRTMC 29
PRTMC 30
PRTMC 31
PRTMC 32
PRTMC 33
PRTMC 34
PRTMC 35
PRTMC 36
PRTMC 37
PRTMC 38
PRTMC 39
PRTMC 40
PRTMC 41
PRTMC 42
PRTMC 43
PRTMC 44
PRTMC 45
PRTMC 46
PRTMC 47
PRTMC 48
PRTMC 49
PRTMC 50
PRTMC 51
PRTMC 52
PRTMC 53
PRTMC 54
PRTMC 55
PRTMC 56
PRTMC 57
PRTMC 58
PRTMC 59

```

SUBROUTINE PRTWAL

PRINTS PRESSURE COEFFICIENT AND FLOW ANGLE
ON Y=H AND Y=+H, AND PLOTS CP ALONG SIDE OF
TABULATION.
CALLED BY = PRINT.

```

COMMON / COM1/ TMIN, IMAX, IUP, IDOWN, ILE,
1 ITE, JMIN, JMAX, JUP, JLOW,
2 JTOP, JBOT
COMMON / COM5/ XDIFF(100), YDIFF(100)
COMMON / COM12/ F, H, HALFPI, PI, RTKPOR,
1 TWOPI
COMMON / COM13/ CDFACT, CLFACT, CMFACT, CPFACT, CPSTAR
LOGICAL PHYS
INTEGER PRTFLO, SIMDEF
COMMON / COM27/ CL, DELTA, DELRT2, EMACH, EMROOT,
1 PHYS, PRTFLO, SIMDEF, SONVEL, VFACT,
2 YFACT
INTEGER BCTYPE
COMMON / COM28/ BCTYPE, CIRCFF, FHINV, POR, CIRCTE
COMMON / COM30/ CPLW(100), CPUW(100), VWL(100), VUW(100), REST(4)
DIMENSION BCT(15), LINE1(60)

```

```

DATA IB /3H /, IL / 1HL/, IU /1HU/, IS / 1H/, IBB /1HB/,
DATA BCT /4H /, 4HFREE,4H AIR,4H SD,4HLID,4H WALL,4H /,
1 4HFREE,4H JET,4H SLOT,4H TED,4H WALL,4H POR,4H OUS,4H WALL/

```

PRINT SINGLE VARIABLES

```

I2 = 3 * BCTYPE
I1 = I2 - 2
WRITE (6,903) (BCT(I),I=I1,I2)
THH = H * YFACT
WRITE(6,904) THH,H
IF ( BCTYPE .LT. 5 ) GO TO 4
PORF = POR / YFACT
WRITE(6,905) PORF,POR
4 CONTINUE
IF ( BCTYPE .NE. 4 .AND. BCTYPE .NE. 6 ) GO TO 6
WRITE (6,906) F
6 CONTINUE
WRITE (6,907) CPSTAR

```

```

CPMIN = 1.0E37
CPMAX = -CPMIN
CPT = -2.0 * CPFACT
DO 10 I=IUP, IDOWN
CPLW(I) = CPT * PX(I,JMIN)
CPUW(I) = CPT * PX(I,JMAX)
CPMAX = AMAX1(CPMAX,CPUW(I), CPLW(I))
CPMIN = AMIN1(CPMIN,CPUW(I), CPLW(I))
10 CONTINUE
DO 20 I=IUP, IDOWN
IF (BCTYPE .NE. 2) GO TO 11

```

```

      SOLID WALL
VWL(I) = 0.0
VUW(I) = 0.0
GO TO 20
11 CONTINUE
IF (BCTYPE .NE. 3) GO TO 12
      FREE JET
VWL(I) = VFACT * PY(I,JMIN)
VUW(I) = VFACT * PY(I,JMAX)
GO TO 20
12 CONTINUE
IF (BCTYPE .NE. 4) GO TO 13

```

```

      SLOTTED WALL
VWL(I) = VFACT * FHINV * (P(JBOT,I) + .75 * CIRCFF)
VUW(I) = -VFACT * FHINV * (P(JTOP,I) - .25 * CIRCFF)
GO TO 20
13 CONTINUE
      POROUS WALL
IF (POR .GT. 1.5) GO TO 14
VWL(I) = VFACT * POR * XDIFF(I)*(P(JMIN,I)-P(JMIN,I-1))
VUW(I) = -VFACT * POR * XDIFF(I)*(P(JMAX,I)-P(JMAX,I-1))
GO TO 20
14 CONTINUE
VWL(I) = VFACT * .25*(P(JMIN+1,I+1)+2.*P(JMIN+1,I)+P(JMIN+1,I-1)
- P(JMIN,I+1)-2.*P(JMIN,I)-P(JMIN,I-1))
/ (Y(JMIN+1)-Y(JMIN))
VUW(I) = VFACT * .25*(P(JMAX,I+1) +2.*P(JMAX,I) +P(JMAX,I-1))

```

```

SUBROUTINE PRTSK(7,ARG,L,NSHOCK,CDSK,LPRTI)
PRINTOUT WAVE DRAG CONTRIBUTION AND TOTAL PRESSURE
LOSS ALONG SHOCK WAVE
CALLED BY = CDCCLE.
COMMON / COM2/ AK, ALPHA, DUB, GAM1, RTK
COMMON / COM13/ CDFACT, CLFACT, CMFACT, CPFACT, CPSTAR
LOGICAL PHYS
INTEGER PRTFLO, SIMDEF
COMMON / COM27/ CL, DELTA, DELRT2, EMACH, EMROOT,
1 PHYS, PRTFLO, SIMDEF, SONVEL, VFACT,
2 YFACT
DIMENSION Z(1), ARG(1)
CDYCDF = -CDFACT*GAM1/(6.*YFACT)
POYCDF = DELTA*DELTA*GAM1*(GAM1-1.)/12.
IF(NSHOCK .EQ.1) WRITE(6,1001)
WRITE(6,1002) NSHOCK, CDSK
DO 10 K = 1,L
YY = Z(K)*YFACT
CDY = CDYCDF*ARG(K)
POY = 1. + POYCDF*ARG(K)
WRITE(6,1003) YY,CDY,POY
10 CONTINUE
IF(LPRTI .EQ. 1) WRITE(6,1004)
1001 FORMAT(1H1,35X,43HSHOCK WAVE DRAG AND TOTAL PRESSURE PROFILE ,
1 6HOUTPUT/36X,49(1H*)//39H INVISCID WAKE PROFILES FOR INDIVIDUAL ,
2 39HSHOCK WAVES WITHIN MOMENTUM CONTOUR)
1002 FORMAT(10HSHOCK,13/26H WAVE DRAG FOR THIS SHOCK*,F12.6/
* 6X,1HY,9X,5HCD(Y),8X,8HP0(POINF/))
1003 FORMAT(1X,3F12.8)
1004 FORMAT(39HSHOCK WAVE EXTENDS OUTSIDE CONTOUR/
* 61H PRINTOUT OF SHOCK LOSSES ARE NOT AVAILABLE FOR REST OF SHOCK)
END

```

```

PRTSK 2
PRTSK 3
PRTSK 4
PRTSK 5
PRTSK 6
COM2 2
COM13 2
COM27 2
COM27 2
COM27 3
COM27 4
COM27 5
COM27 6
PRTSK 10
PRTSK 11
PRTSK 12
PRTSK 13
PRTSK 14
PRTSK 15
PRTSK 16
PRTSK 17
PRTSK 18
PRTSK 19
PRTSK 20
PRTSK 21
PRTSK 22
PRTSK 23
PRTSK 24
PRTSK 25
PRTSK 26
PRTSK 27
PRTSK 28
PRTSK 29
PRTSK 30
PRTSK 31

```

```

PRTWAL 2
PRTWAL 3
PRTWAL 4
PRTWAL 5
PRTWAL 6
PRTWAL 7
PRTWAL 8
PRTWAL 9
PRTWAL 10
PRTWAL 11
PRTWAL 12
PRTWAL 13
PRTWAL 14
PRTWAL 15
PRTWAL 16
PRTWAL 17
PRTWAL 18
PRTWAL 19
PRTWAL 20
PRTWAL 21
PRTWAL 22
PRTWAL 23
PRTWAL 24
PRTWAL 25
PRTWAL 26
PRTWAL 27
PRTWAL 28
PRTWAL 29
PRTWAL 30
PRTWAL 31
PRTWAL 32
PRTWAL 33
PRTWAL 34
PRTWAL 35
PRTWAL 36
PRTWAL 37
PRTWAL 38
PRTWAL 39
PRTWAL 40
PRTWAL 41
PRTWAL 42
PRTWAL 43
PRTWAL 44
PRTWAL 45
PRTWAL 46
PRTWAL 47
PRTWAL 48
PRTWAL 49
PRTWAL 50
PRTWAL 51
PRTWAL 52
PRTWAL 53
PRTWAL 54
PRTWAL 55
PRTWAL 56
PRTWAL 57
PRTWAL 58
PRTWAL 59
PRTWAL 60
PRTWAL 61
PRTWAL 62
PRTWAL 63
PRTWAL 64
PRTWAL 65
PRTWAL 66
PRTWAL 67
PRTWAL 68
PRTWAL 69
PRTWAL 70
PRTWAL 71
PRTWAL 72
PRTWAL 73
PRTWAL 74
PRTWAL 75
PRTWAL 76
PRTWAL 77
PRTWAL 78

```

xcn2020

```

1          - P(JMAX-1,I+1)-2.*P(JMAX-1,I)-P(JMAX-1,I-1)
2          / (Y(JMAX)-Y(JMAX-1))
20 CONTINUE
CPLARG = ANAX1CPHAX,ABS(CPHIN)
UNPCOL = CPLARG / 25.
C          LOCATE CP* FOR PRINTER PLOT
COL      = -CPSTAR / UNPCOL
NCOL     = SIGN(ABS(COL)+.5), COL
NCOLS    = NCOL + 30
WRITE (6,210)
C          PRINT COLUMN HEADERS.
WRITE (6,220)
DO 40 I=1,UP,IDOWN
DO 30 K=1,60
LINEI(K) = 18
30 CONTINUE
COL      = -CPW(I) / UNPCOL
NCOL     = SIGN(ABS(COL) + .5), COL
NCOLU    = NCOL + 30
LINEI(NCOLU) = IU
COL      = -CPLW(I) / UNPCOL
NCOL     = SIGN(ABS(COL) + .5), COL
NCOLL    = NCOL + 30
LINEI(NCOLL) = IL
IF (NCOLL.EQ. NCOLU) LINEI(NCOLL) = IBB
IF (IABS(NCOLS) .LT. 61) LINEI(NCOLS) = 15
WRITE (6,250) I, X(I), CPL(I), VLM(I), CPW(I), VUK(I), LINEI
40 CONTINUE
RETURN
210 FORMAT(1H0,27X,5HLOWER,23X,5HUPPER/28X,4HY=-H,24X,4HY=+H)
220 FORMAT(3X,1H1,8X,1HX,10X,2HCP,9X,5HTHETA,12X,2HCP,9X,5HTHETA/)
250 FORMAT(1H ,13,2F12.6,4X,2F12.6,2X,60A1)
903 FORMAT(2H1 ,3A,20H BOUNDARY CONDITION.)
904 FORMAT(1H0,10X,25HH (TUNNEL HALF HEIGHT) =,F9.6,15X,
1 10HSCALED H =,F9.6)
905 FORMAT(1H0,11X,23HPOR (POROSITY FACTOR) =,F9.6,13X,
1 12HSCALED POR =,F9.6)
906 FORMAT(1H0,14X,20HF (SLOT PARAMETER) =,F9.6)
907 FORMAT(1H0,29X,2HCP =,F9.6)
END
FUNCTION PX(I,J)
C          FUNCTION PX COMPUTES U = DP/DX AT POINT I,J
C          CALLED BY - CDCOLE, DRAG, FINSK, MACHMP, MLINE,
C          NEWSK, PRINT1, PRTFLD, PRTVAL.
COMMON / COM1 / IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JSOT
COMMON / COM5 / XDIFF(100),YDIFF(100)
C          TEST TO LOCATE END POINTS
IF(I.EQ. IMIN) GO TO 19
IF(I.EQ. IMAX) GO TO 20
C          INTERIOR MESH POINT
PJI = P(J,I)
PX = .5*(XDIFF(I+1)*(P(J,I+1)-PJI) + XDIFF(I)*(PJI-P(J,I-1)))
RETURN
10 CONTINUE
C          UPSTREAM BOUNDARY
PX = 1.5*XDIFF(I+1)*(P(J,I+1)-P(J,I)) -
1 0.5*XDIFF(I+2)*(P(J,I+2) - P(J,I+1))
RETURN
20 CONTINUE
C          DOWNSTREAM BOUNDARY
PX = 1.5*XDIFF(I)*(P(J,I) - P(J,I-1))
1 -0.5*XDIFF(I-1)*(P(J,I-1) - P(J,I-2))
RETURN
END
PRTVAL 79
PRTVAL 80
PRTVAL 81
PRTVAL 82
PRTVAL 83
PRTVAL 84
PRTVAL 85
PRTVAL 86
PRTVAL 87
PRTVAL 88
PRTVAL 89
PRTVAL 90
PRTVAL 91
PRTVAL 92
PRTVAL 93
PRTVAL 94
PRTVAL 95
PRTVAL 96
PRTVAL 97
PRTVAL 98
PRTVAL 99
PRTVAL 100
PRTVAL 101
PRTVAL 102
PRTVAL 103
PRTVAL 104
PRTVAL 105
PRTVAL 106
PRTVAL 107
PRTVAL 108
PRTVAL 109
PRTVAL 110
PRTVAL 111
PRTVAL 112
PRTVAL 113
PRTVAL 114
PRTVAL 115
PRTVAL 116
PRTVAL 117
PRTVAL 118
PRTVAL 119
PRTVAL 120
PRTVAL 121
PRTVAL 122
PRTVAL 123
PRTVAL 124
PRTVAL 125
PRTVAL 126
PRTVAL 127
PRTVAL 128
PRTVAL 129
PRTVAL 130
PRTVAL 131

```

```

FUNCTION PY(I,J)
C          FUNCTION PY COMPUTES V = DP/DY AT POINT I,J
C          CALLED BY - CDCOLE, PRTFLD, PRTVAL.
COMMON / COM1 / IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JSOT
COMMON / COM2 / AK , ALPHA , DUB , GAM1 , RTK
COMMON / COM5 / XDIFF(100),YDIFF(100)
COMMON / COM6 / FL(100) , FXL(100) , FU(100) , FXU(100),
1 CAMBER(100) , THICK(100),VOL , XFOIL(100), IFOIL
COMMON /COM26/ PJUMP(100)
INTEGER 8CTYPE
COMMON /COM28/ 8CTYPE , CIRCOFF , FHINV , POR , CIRCTE
C          TEST FOR END POINTS OR POINTS NEAR AIRFOIL SLIT
IF(J.EQ. JMIN) GO TO 10
IF(J.EQ. JLOW) GO TO 20
IF(J.EQ. JUP) GO TO 40
IF(J.EQ. JMAX) GO TO 50
C          I,J IS AN INTERIOR POINT
PJI = P(J,I)
PY = .5*(YDIFF(I+1)*(P(J,I+1)-PJI) + YDIFF(I)*(PJI-P(J-1,I)))
RETURN
10 CONTINUE
C          I,J IS ON LOWER BOUNDARY. USE ONE SIDED DERIVATIVE
PY = 1.5*YDIFF(J+1)*(P(I+1,I) - P(J,I)) -
1 0.5*YDIFF(J+2)*(P(I+2,I) - P(J+1,I))
RETURN
20 CONTINUE
C          I,J IS ON ROW OF MESH POINTS BELOW AIRFOIL.
VMINUS = YDIFF(J)*(P(J,I) - P(J-1,I))
C          TEST TO SEE IF I,J IS AHEAD, UNDER, OR BEHIND SLIT.
IF(I.LT. ILE) GO TO 21
IF(I.GT. ITE) GO TO 22
C          I,J IS UNDER AIRFOIL. USE DERIVATIVE BOUNDARY
CONDITION.
IC = I - ILE + 1
PY = .5 * (FXL(IC) - ALPHA + VMINUS)
RETURN
21 CONTINUE
C          I,J IS AHEAD OF AIRFOIL.
PY = .5*((P(JUP,I) - P(JLOW,I)) * YDIFF(JUP) + VMINUS)
RETURN
22 CONTINUE
C          I,J IS BEHIND AIRFOIL
PY = .5*((P(JUP,I) - PJUMP(I) - P(JLOW,I)) * YDIFF(JUP) + VMINUS)
RETURN
40 CONTINUE
C          I,J IS ON ROW OF MESH POINTS ABOVE AIRFOIL
VPLUS = YDIFF(J+1)*(P(J+1,I) - P(J,I))
C          TEST TO SEE IF I IS AHEAD OF, OVER, OR BEHIND
AIRFOIL SLIT.
IF(I.LT. ILE) GO TO 41
IF(I.GT. ITE) GO TO 42
IC = I - ILE + 1
PY = .5 * (VPLUS + FXU(IC) - ALPHA)
RETURN
41 CONTINUE
C          I,J IS AHEAD OF AIRFOIL.
PY = .5*((P(JUP,I) - P(JLOW,I)) * YDIFF(JUP) + VPLUS)
RETURN
42 CONTINUE
C          I,J IS BEHIND AIRFOIL
PY = .5*((P(JUP,I) - PJUMP(I) - P(JLOW,I)) * YDIFF(JUP) + VPLUS)
RETURN
50 CONTINUE
C          I,J IS ON TOP ROW OF MESH POINTS. USE ONE SIDED
FORMULA.
PY = 1.5*YDIFF(J)*(P(J,I) - P(J-1,I))
1 - 0.5*YDIFF(J-1)*(P(J-1,I) - P(J-2,I))
RETURN
END
PY 2
PY 3
PY 4
PY 5
BLANK 1
COM1 2
COM1 3
COM1 4
COM2 2
COM5 2
COM6 2
COM6 3
COM26 2
COM28 2
COM28 3
PY 13
PY 14
PY 15
PY 16
PY 17
PY 18
PY 19
PY 20
PY 21
PY 22
PY 23
PY 24
PY 25
PY 26
PY 27
PY 28
PY 29
PY 30
PY 31
PY 32
PY 33
PY 34
PY 35
PY 36
PY 37
PY 38
PY 39
PY 40
PY 41
PY 42
PY 43
PY 44
PY 45
PY 46
PY 47
PY 48
PY 49
PY 50
PY 51
PY 52
PY 53
PY 54
PY 55
PY 56
PY 57
PY 58
PY 59
PY 60
PY 61
PY 62
PY 63
PY 64
PY 65
PY 66
PY 67
PY 68
PY 69
PY 70
PY 71
PY 72
PY 73
PY 74

```

```

SUBROUTINE READIN
C
C      CALLED BY - TSFOIL.
C
C      INPUT EXPLANATION
C
C      *****
C
C      ALL INPUT IS READ IN THIS SUBROUTINE. THE ORDER IS AS DESCRIBED
C      BELOW.
C
C      1.) ONE CARD OF TITLE INFORMATION, AN *** (ALPHANUMERIC) FORMAT
C      IS USED TO READ AND WRITE THIS INFORMATION. MULTIPLE CASES
C      MAY BE RUN WITH THIS PROGRAM AND THE DATA FOR EACH CASE
C      MUST START WITH THIS CARD. THE LAST CARD OF THE INPUT MUST
C      BE A CARD WITH THE WORD *FINISHED* IN THE FIRST 8 COLUMNS.
C
C      2.) NAMELIST CONTAINING THESE PARAMETERS IS NOW READ. (SEE
C      FORTRAN MANUAL FOR DESCRIPTION OF NAMELIST INPUT). THE
C      BLOCK DATA SUBROUTINE SETS A DEFAULT VALUE, AS NOTED BELOW,
C      FOR ALL OF THESE PARAMETERS.
C
C      ONLY THE VALUES WHICH ARE DIFFERENT FROM THE PREVIOUS CASE
C      (OR DEFAULT) MUST BE INCLUDED, ALTHOUGH AT LEAST ONE VALUE
C      MUST BE INPUT BY NAMELIST FOR EACH CASE.
C
C      * (F) = FLOATING POINT *
C      * (I) = INTEGER *
C      * (L) = LOGICAL *
C
C      DEFAULT
C      VALUE
C
C      AMESH (L) OPTION FOR ANALYTICAL MESH CALC. .F.
C      .TRUE. X AND Y MESH VALUES ARE COMPUTED
C      ACCORDING TO AN ANALYTIC
C      FORMULATION. THE GRID USED IS
C      IMAXI=31, JMAXI=69, AND ALLOWED
C      FOR 1-1/2 PITCH CUTS.
C      .FALSE. X AND Y POINTS ARE THE DEFAULT
C      VALUES OR THE VALUES SUPPLIED
C      BY THE USER THRU NAMELIST.
C
C      EMACH (F) FREESTREAM MACH NUMBER. .75
C      NOTE****EMACH MAY NOT BE = 1.0
C
C      DELTA (F) BODY THICKNESS RATIO. .215
C
C      ALPHA (F) ANGLE OF ATTACK (DEGREES IF PHYS=T) .12
C
C      AK (F) TRANSONIC SIMILARITY PARAMETER. 0.0
C      (INPUT REQUIRED ONLY IF PHYS = .F.)
C
C      GAM (F) RATIO OF SPECIFIC HEATS. 1.4
C
C      SIMOEF (I) SIMILARITY DEFINITION. 3
C      =1 COLE
C      =2 SPREITER
C      =3 KRUPP
C      =4 USER
C
C      PRTFLO (I) OPTION FOR PRINT OF FINAL FLOW FIELD. 1
C      =1 NO FLOW FIELD PRINT
C      =2 ALL J LINES PRINTED.
C      =3 PRINT 3 J LINES AROUND MAXIMUM ERROR
C
C      PSTART (I) OPTION FOR INITIALIZING P ARRAY. 1
C      =1 SET TO ZERO.
C      =2 READ P FROM UNIT 7
C      =3 USE P IN CORE (PREVIOUS CASE).
C
C      PSAVE (L) OPTION FOR SAVING RESTART BLOCK OF .F.
C      VALUES ON UNIT 3.
C      .TRUE. SAVE FOR RESTART.
C      .FALSE. DO NOT SAVE.
C
C      FCR (L) FULLY CONSERVATIVE RELAXATION OPTION .T.
C      .TRUE. DIFFERENCE EQUATIONS ARE
C      FULLY CONSERVATIVE FORM.
C      .FALSE. DIFFERENCE EQUATIONS NOT
C      CONSERVATIVE AT SHOCK WAVES.
C
C      KUTTA (L) KUTTA CONDITION OPTION. .T.
C      .TRUE. KUTTA CONDITION IS ENFORCED
C      BY USER.
C      .FALSE. LIFT COEFFICIENT SPECIFIED
C      BY USER.
C
C      CLSET (F) LIFT COEFFICIENT, USED IF KUTTA .0
C      IS FALSE.
C
C      BCFOIL (I) OPTION FOR FOIL OR BODY GEOMETRY. 3
C      =1 HACA00XX
C      =2 PARABOLIC ARC.
C      =3 ORDINATES (READ LATER IN NAMELIST
C      IF DIFFERENT FROM DEFAULT VALUES WHICH
C      ARE FOR THE KORN AIRFOIL).
C      =4 USER OPTION.
C
C      BCTYPE (I) DESCRIBES THE TYPE OF FLOW TO BE 1
C      COMPUTED.
C      =1 FREE AIR.
C      =2 SOLID WALL.
C      =3 FREE JET.
C      =4 SLOTTED WALL.
C      =5 POROUS WALL.
C
C      F (F) TUNNEL SLOT PARAMETER. 0.
C
C      READIN 2 C
C      READIN 3 C
C      READIN 4 C
C      READIN 5 C
C      READIN 6 C
C      READIN 7 C
C      READIN 8 C
C      READIN 9 C
C      READIN 10 C
C      READIN 11 C
C      READIN 12 C
C      READIN 13 C
C      READIN 14 C
C      READIN 15 C
C      READIN 16 C
C      READIN 17 C
C      READIN 18 C
C      READIN 19 C
C      READIN 20 C
C      READIN 21 C
C      READIN 22 C
C      READIN 23 C
C      READIN 24 C
C      READIN 25 C
C      READIN 26 C
C      READIN 27 C
C      READIN 28 C
C      READIN 29 C
C      READIN 30 C
C      READIN 31 C
C      READIN 32 C
C      READIN 33 C
C      READIN 34 C
C      READIN 35 C
C      READIN 36 C
C      READIN 37 C
C      READIN 38 C
C      READIN 39 C
C      READIN 40 C
C      READIN 41 C
C      READIN 42 C
C      READIN 43 C
C      READIN 44 C
C      READIN 45 C
C      READIN 46 C
C      READIN 47 C
C      READIN 48 C
C      READIN 49 C
C      READIN 50 C
C      READIN 51 C
C      READIN 52 C
C      READIN 53 C
C      READIN 54 C
C      READIN 55 C
C      READIN 56 C
C      READIN 57 C
C      READIN 58 C
C      READIN 59 C
C      READIN 60 C
C      READIN 61 C
C      READIN 62 C
C      READIN 63 C
C      READIN 64 C
C      READIN 65 C
C      READIN 66 C
C      READIN 67 C
C      READIN 68 C
C      READIN 69 C
C      READIN 70 C
C      READIN 71 C
C      READIN 72 C
C      READIN 73 C
C      READIN 74 C
C      READIN 75 C
C      READIN 76 C
C      READIN 77 C
C      READIN 78 C
C      READIN 79 C
C      READIN 80 C
C      READIN 81 C
C      READIN 82 C
C      READIN 83 C
C      READIN 84 C
C      READIN 85 C
C      READIN 86 C
C      READIN 87 C
C
C      H (F) TUNNEL HALF HEIGHT/CHORD RATIO. 0.
C
C      POR (F) WALL POROSITY FACTOR. .0
C
C      PHYS (L) TYPE OF SCALING TO USE FOR I/O. .T.
C      .TRUE. PHYSICALLY SCALED VALUES.
C      .FALSE. TRANSONICALLY SCALED VALUES
C      FOR PHYS = .F., ALSO INPUT VALUE FOR AK.
C
C      IMAXI (I) NUMBER OF X-MESH POINTS (L.E. 100) 77
C
C      JMAXI (I) NUMBER OF Y-MESH POINTS (L.E. 100) 56
C
C      IMIN (I) X MESH POINT WHERE CALC IS TO START 1
C
C      JMIN (I) Y MESH POINT WHERE CALC IS TO START 1
C
C      ICUT (I) CONTROL FOR MESH CUT AND REFINEMENT. 2
C      = 0 INPUT MESH IS USED TO CONVERGENCE.
C      = 1 INPUT MESH MAY BE CUT ONCE.
C      = 2 INPUT MESH MAY BE CUT TWICE.
C
C      WE (F) 3 VALUES FOR RELAXATION FACTOR FOR 1-8
C      ELLIPTIC PTS. 1-ST FOR COARSE MESH, 1.9
C      2-ND FOR MED. MESH AND 3-RD FOR 1.95
C      FINE MESH. DEFAULT VALUES ARE SUGGESTED
C      VALUES, IN ORDER, IF SPECIFIED IN INPUT
C      ALL THREE VALUES MUST BE GIVEN.
C
C      WCIRC (F) RELAXATION FACTOR FOR CIRCULATION. 1.0
C
C      MAXIT (I) MAXIMUM NUMBER OF ITERATION CYCLES 500
C      ALLOWED.
C
C      CVERGE (F) CONVERGENCE CRITERION FOR RESIDUALS <00001
C      OF P.
C
C      DIVER (F) DIVERGENCE CRITERION FOR RESIDUALS 10.
C
C      RIGF (F) REIGLES RULE FOR BODY SLOPE. 0.0
C
C      EPS (F) COEFFICIENT OF PXT .2
C
C      IPRTER (I) CONTROL FOR FREQUENCY OF PRINT OF
C      LINE IN MESH WHERE ERROR IS LARGEST.
C      I.E. IPRTER=10, LINE WILL BE PRINTED
C      EVERY 10-TH ITERATION.
C
C      NOTE***** WHEN ARRAYS ARE READ BY NAMELIST THE FULL ARRAY MUST BE
C      SET, I.E. IF ALL VALUES ARE NOT REQUIRED THE ARRAY MAY
C      BE FILLED USING MULTIPLE ZEROS. (N=0,0)
C
C      XU (F) ARRAY - X VALUES FOR UPPER BODY. USED
C      IF BCFOIL = 3. KORN AIRFOIL USING ALL
C      100 PTS.(UPPER) AND 75 (LOWER) IS DEFAULT.
C
C      XL (F) ARRAY - X VALUES FOR LOWER BODY.
C
C      YU (F) ARRAY - Y VALUES FOR UPPER BODY.
C
C      YL (F) ARRAY - Y VALUES FOR LOWER BODY.
C
C      NU (I) NUMBER OF POINTS TO USE FOR UPPER BODY 100
C
C      NL (I) NUMBER OF POINTS TO USE FOR LOWER BODY 75
C
C      ***** NOTE THIS PROGRAM USES A MESH REFINEMENT METHOD FOR DECREASING
C      COMPUTER TIME. FOLLOW THE RULES BELOW FOR CONSTRUCTING THE
C      X AND Y MESH TO TAKE FULL ADVANTAGE OF THIS FEATURE.
C
C      IMAXI - ITE SHOULD BE A MULTIPLE OF 4.
C      ITE - IMIN SHOULD BE A MULTIPLE OF 4.
C      JMAXI - JUP + 1 SHOULD BE A MULTIPLE OF 4
C      JLOW - JMIN + 1 SHOULD BE A MULTIPLE OF 4
C      (WHERE JLOW IS LAST POINT BELOW SLIT AND
C      JUP IS FIRST POINT ABOVE SLIT.)
C
C      WHERE ITE = I FOR X = 1.0 (OR POINT ON BODY CLOSEST TO
C      X = 1.0).
C
C      SUBROUTINE KCMESH INSPECTS THE X AND Y MESHES TO SEE IF
C      THIS IS TRUE AND, IF NOT, WILL MODIFY INPUT MESH IN SOME
C      CASES.
C
C      XIN (F) ARRAY - X MESH POINTS. LIMIT 100 PTS.
C
C      YIN (F) ARRAY - Y MESH POINTS. LIMIT 100 PTS.
C      X AND Y MESH DEFAULT VALUES ARE KRUPP
C      BASIC GRID.
C
C      COMMON / COM1 / IMIN , IMAX , IUP , IDOWN , ILE ,
C      1 ITE , JMIN , JMAX , JUP , JLOW ,
C      2 JTOP , JBOT
C
C      COMMON / COM2 / AK , ALPHA , DUB , GAM1 , RTK
C
C      LOGICAL ABORT
C
C      COMMON / COM3 / IREF , ABORT , ICUT , KSTEP
C
C      LOGICAL AMESH
C
C      COMMON / COM4 / XIN(100) , YIN(100) , AMESH
C
C      COMMON / COM5 / CVERGE , DVERGE , IPRTER , MAXIT ,
C      1 WE(3) , EPS
C
C      INTEGER BCFOIL
C
C      COMMON / COM9 / BCFOIL , NL , NU , XL(100) , XU(100) ,
C      1 YL(100) , YU(100) , RIGF
C
C      LOGICAL XGRDIN , YGRDIN
C
C      COMMON / COM10 / YFREE(100) , YTUN(100) , XKRUPP(100) , GAM ,
C      1 JMKF , JXKT , XGRDIN , YGRDIN
C
C      INTEGER PSTART
C
C      LOGICAL PSAVE
C
C      READIN 88 C
C      READIN 89 C
C      READIN 90 C
C      READIN 91 C
C      READIN 92 C
C      READIN 93 C
C      READIN 94 C
C      READIN 95 C
C      READIN 96 C
C      READIN 97 C
C      READIN 98 C
C      READIN 99 C
C      READIN 100 C
C      READIN 101 C
C      READIN 102 C
C      READIN 103 C
C      READIN 104 C
C      READIN 105 C
C      READIN 106 C
C      READIN 107 C
C      READIN 108 C
C      READIN 109 C
C      READIN 110 C
C      READIN 111 C
C      READIN 112 C
C      READIN 113 C
C      READIN 114 C
C      READIN 115 C
C      READIN 116 C
C      READIN 117 C
C      READIN 118 C
C      READIN 119 C
C      READIN 120 C
C      READIN 121 C
C      READIN 122 C
C      READIN 123 C
C      READIN 124 C
C      READIN 125 C
C      READIN 126 C
C      READIN 127 C
C      READIN 128 C
C      READIN 129 C
C      READIN 130 C
C      READIN 131 C
C      READIN 132 C
C      READIN 133 C
C      READIN 134 C
C      READIN 135 C
C      READIN 136 C
C      READIN 137 C
C      READIN 138 C
C      READIN 139 C
C      READIN 140 C
C      READIN 141 C
C      READIN 142 C
C      READIN 143 C
C      READIN 144 C
C      READIN 145 C
C      READIN 146 C
C      READIN 147 C
C      READIN 148 C
C      READIN 149 C
C      READIN 150 C
C      READIN 151 C
C      READIN 152 C
C      READIN 153 C
C      BLANK 2
C      COM1 2
C      COM1 3
C      COM1 4
C      COM2 2
C      COM3 2
C      COM3 3
C      COM4 3
C      COM4 3
C      COM8 2
C      COM8 3
C      COM9 3
C      COM9 3
C      COM9 4
C      COM10 2
C      COM10 3
C      COM10 4
C      COM11 2
C      COM11 3

```

APPENDIX

```

COMMON /COM11/ ALPHA , CLOLO , DELTAO , DUBO , EMACHO , COM11 4
1 IMINO , IMAXO , IMAXI , JMINO , JMAXO , COM11 5
2 JMAXI , PSAVE , PSTART , TITLE(8), TITLEO(8), COM11 6
3 VOLO , XOLD(100),YOLD(100) , COM11 7
COMMON /COM12/ F , H , HALFFPI , PI , RTKPOR , COM12 2
1 TWHOPI COM12 3
LOGICAL FCR , KUTTA COM12 3
COMMON /COM14/ CLSET , FCR , KUTTA , WCIRC COM14 3
LOGICAL PHYS COM14 3
INTEGER PRTFLO , SIMDEF COM27 2
COMMON /COM27/ CL , DELTA , DELRT2 , EMACH , ENROOT , COM27 3
1 PHYS , PRTFLO , SIMDEF , SUNVEL , VFACT , COM27 4
2 YFACT COM27 5
INTEGER BCTYPE COM28 2
COMMON /COM28/ BCTYPE , CIRCFF , FHINV , POR , CIRCTE COM28 3
DATA DONE /8HFINISHED /
DATA IFIRST /0/
NAMelist /INP/ AK , ALPHA , AMESH , BCFOIL , BCTYPE , CLSET,
1 CVERGE , DELTA , DVERGE , EMACH , EPS , F , COM17 170
2 FCR , GAM , H , ICUT , IMAXI , IMIN , F , COM17 171
3 IPRTER , JMAXI , JMIN , KUTTA , MAXIT , NL , COM17 172
4 NU , PHYS , POR , PRTFLO , PSAVE , NL , COM17 173
5 PSTART , RIGF , SIMDEF , WCIRC , WE , COM17 174
6 XIN , YIN , XL , YL , XU , YU , COM17 175
7 XGRDIN , YGRDIN COM17 176
5 CONTINUE
IF (IFIRST .NE. 0) GO TO 2
IFIRST = 1
CALL SECONO (TIME2)
2 CONTINUE
TIME1 = TIME2
CALL SECONO (TIME2)
ELPTM = TIME2 - TIME1
IF (ELPTM .LT. .01) GO TO 3
WRITE (6,930) ELPTM
3 CONTINUE
READ (5,906) TITLE
WRITE(6,901) TITLE
IF (TITLE(1) .NE. DONE) GO TO 10
WRITE(6,904)
STOP
10 CONTINUE
READ (5,INP)
IF (PSTART .NE. 3) GO TO 13
IF (.NOT. ABORT) GO TO 13
WRITE (6,905)
GO TO 5
13 CONTINUE
IF ( PHYS ) AK = 0.0
IF (.NOT. AMESH ) GO TO 14
CALL AMESH
GO TO 22
14 CONTINUE
IF (YGRDIN) GO TO 18
IF YIN NOT READ IN NAMelist, FILL YIN BY
DEFAULT VALUE FOR TUNNEL OR FREE AIR CASE.
IF (BCTYPE .NE. 1) GO TO 16
JMAXI = JMAX
DO 15 J=JMIN,JMAXI
YIN(J) = YFREE(J)
15 CONTINUE
GO TO 18
16 CONTINUE
JMAXI = JMAX
DO 17 J=JMIN,JMAXI
YIN(J) = YTUN(J)
17 CONTINUE
18 CONTINUE
IF (XGRDIN) GO TO 22
DO 21 I=IMIN,IMAXI
21 XIN(I)=XKRUPP(I)
22 CONTINUE
WRITE (6,921)
WRITE (6,906) EMACH, POR, IMIN, BCTYPE, AMESH
WRITE (6,907) DELTA, CLSET, IMAXI, BCFOIL, PHYS
WRITE (6,908) ALPHA, EPS , JMIN , PSTART, PSAVE
WRITE (6,909) AK , RIGF , JMAXI , PRTFLO , KUTTA
WRITE (6,910) GAM, WCIRC, MAXIT, IPRTER, FCR
WRITE (6,911) F, CVERGE, NU, SIMDEF
WRITE (6,912) H, DVERGE, NL, ICUT
WRITE (6,913) WE
WRITE (6,913)
WRITE (6,919) (XIN(I),I=IMIN,IMAXI)
COM11 4
WRITE (6,914)
COM11 5
WRITE (6,919) (YIN(I),J=JMIN,JMAXI)
COM11 6
IF (BCFOIL .LE. 2) GO TO 19
COM11 7
WRITE (6,915)
COM11 7
WRITE (6,919) (XU(I),I=1,NU)
COM12 2
COM12 3
WRITE (6,916)
COM12 3
WRITE (6,919) (YU(I),I=1,NU)
COM14 2
COM14 3
WRITE (6,917)
COM27 2
WRITE (6,919) (XL(I),I=1,NL)
COM27 3
WRITE (6,918)
COM27 4
WRITE (6,919) (YL(I),I=1,NL)
COM27 5
19 CONTINUE
GAMI = GAM + 1.0
IMEF = 0
IMAX = IMAXI
JMAX = JMAXI
IM1 = IMAX - 1
JM1 = JMAX - 1
IF (IMAXI .GT. 100 .OR. JMAXI .GT. 100) CALL INPERR (1)
ANY CALL TO INPERR CAUSES A MESSAGE TO BE
PRINTED AND EXECUTION IS STOPPED.
CHECK INPUT MESH FOR MONOTONICALLY INCREASING
VALUES.
DO 20 I=IMIN,IM1
IF (XIN(I) .GE. XIN(I+1)) CALL INPERR (2)
20 CONTINUE
DO 30 J=JMIN,JM1
IF (YIN(J) .GE. YIN(J+1)) CALL INPERR (3)
30 CONTINUE
IF (EMACH .LT. .5 .OR. EMACH .GT. 2.0) CALL INPERR (4)
IF (ALPHA .LT. -9.0 .OR. ALPHA .GT. 9.0) CALL INPERR (5)
IF (DELTA .LT. 0.0 .OR. DELTA .GT. 1.0) CALL INPERR (6)
COMPUTE ILE AND ITE (LEADING AND TRAILING EDGE)
CALL ILSIT ( XIN )
COMPUTE JLOW AND JUP (LOCATION OF BODY SLIT)
CALL JLSIT ( YIN )
CHECK NUMBER OF MESH POINTS; IF NOT ODD ADD
POINTS TO APPROPRIATE AREAS TO MAKE ODD NO.
CALL CKMESH
CHECK BOUNDS OF YMESH FOR TUNNEL CALCULATIONS.
HTM = H + .00001
HTP = H + .00001
YS = ABS(YIN(JMIN))
YE = ABS(YIN(JMAX))
IF (YS .LT. HTM .OR. YE .GT. HTP) GO TO 40
IF (YE .GE. HTM .AND. YE .LE. HTP) GO TO 90
40 CONTINUE
RESCALE Y MESH TO -H,+H BOUNDS.
TERM = -H / YIN(JMIN)
DO 45 J=JMIN,JLOW
YIN(J) = TERM * YIN(J)
45 CONTINUE
TERM = H / YIN(JMAX)
DO 50 J=JUP,JMAX
YIN(J) = TERM * YIN(J)
50 CONTINUE
90 CONTINUE
IF (PSTART .NE. 2) GO TO 100
REWIND 7
READ (7,900) TITLED
READ (7,902) IMAXO, JMAXO, IMINO, JMINO
READ (7,903) CLOLO, EMACHO, ALPHA, DELTAO, VOLO, DUBO
READ (7,903) (XOLD(I),I=IMINO,IMAXO)
READ (7,903) (YOLD(I),J=JMINO,JMAXO)
DO 60 I=IMINO,IMAXO
READ (7,903) (PI,I), (J=JMINO,JMAXO)
60 CONTINUE
WRITE(6,1000) TITLED, IMINO,IMAXO,JMINO,JMAXO,CLOLO,EMACHO,
ALPHA,DELTAO,VOLO, DUBO
READIN 238
READIN 239
READIN 240
READIN 241
READIN 242
READIN 243
READIN 244
READIN 245
READIN 246
READIN 247
READIN 248
READIN 249
READIN 250
READIN 251
READIN 252
READIN 253
READIN 254
READIN 255
READIN 256
READIN 257
READIN 258
READIN 259
READIN 260
READIN 261
READIN 262
READIN 263
READIN 264
READIN 265
READIN 266
READIN 267
READIN 268
READIN 269
READIN 270
READIN 271
READIN 272
READIN 273
READIN 274
READIN 275
READIN 276
READIN 277
READIN 278
READIN 279
READIN 280
READIN 281
READIN 282
READIN 283
READIN 284
READIN 285
READIN 286
READIN 287
READIN 288
READIN 289
READIN 290
READIN 291
READIN 292
READIN 293
READIN 294
READIN 295
READIN 296
READIN 297
READIN 298
READIN 299
READIN 300
READIN 301
READIN 302
READIN 303
READIN 304
READIN 305
READIN 306
READIN 307
READIN 308
READIN 309
READIN 310
READIN 311
READIN 312
READIN 313
READIN 314
READIN 315
READIN 316
READIN 317
READIN 318
READIN 319
READIN 320
READIN 321
READIN 322

```

```

100 CONTINUE
RETURN

900 FORMAT(FA10)
901 FORMAT(1H1,23X,8A10)
902 FORMAT(4I5)
903 FORMAT(8F10.6)
904 FORMAT(1H1)
905 FORMAT(21H0 CALCULATION ABORTED//
1 43H OUTPUT OF PREVIOUS SOLUTION NOT AVAILABLE.)
906 FORMAT(1H0,12X,7HEMACH =,F9.5,8X,5HPDR =,F9.5,7X,6HIMIN =,I5,
1 5X,8HBCTYPE =,I3,9X,8HAMESH =,L1)
907 FORMAT(1H0,12X,7HDELTA =,F9.5,6X,7HCLSET =,F9.5,6X,7HIMAXI =,I5,
1 5X,8HBCFOIL =,I3,10X,7HPHYS =,L1)
908 FORMAT(1H0,12X,7HALPHA =,F9.5,8X,5HSPS =,F9.5,7X,6HJMIN =,I5,
1 5X,8HSPSTART =,I3,9X,8HSPVE =,L1)
909 FORMAT(1H0,15X,6HAK =,F9.5,7X,6HRIIGF =,F9.5,6X,7HJMAXI =,I5,
1 5X,6HPRFLO =,I3,9X,8HKUTTA =,L1)
910 FORMAT(1H0,14X,5HGAM =,F9.5,6X,7HWCIRC =,F9.5,6X,7HMAXIT =,I5,
1 5X,8HIPRTER =,I3,11X,6HFCR =,L1)
911 FORMAT(1H0,16X,3HF =,F9.5,5X,8HCVERGE =,F9.5,9X,4HNU =,I5,
1 5X,6HSIMDEF =,I3)
912 FORMAT(1H0,16X,3HM =,F9.5,5X,8HDSVERGE =,F9.5,9X,4HNL =,I5,
1 7X,8HICUT =,I3)
913 FORMAT(1H0,14X,3HYIN)
914 FORMAT(1H0,14X,3HYIN)
915 FORMAT(1H0,15X,2HXU)
916 FORMAT(1H0,15X,2HYU)
917 FORMAT(1H0,15X,2HXL)
918 FORMAT(1H0,15X,2HYL)
919 FORMAT(1H0,15X,2HXL)
920 FORMAT(1H0,15X,2HME =,F4.2,2(IH,F4.2))
921 FORMAT(////58X,16HINPUT PARAMETERS/58X,16(IH*1//)
930 FORMAT(22H1TIME TO RUN CASE WAS ,F6.2,9H SECONDS/)
1000 FORMAT(39H1P INITIALIZED FROM PREVIOUS RUN TITLED/
1 1X,8A10/31H WHICH HAD THE FOLLOWING VALUES/
1 8H IMIN =,I4/8H IMAX =,I4/8H JMIN =,I4/8H JMAX =,I4/
1 8H CL =,F12.8/8H EMACH =,F12.8/8H ALPHA =,F12.8/
1 8H DELTA =,F12.8/8H VOL =,F12.8/8H DUB =,F12.8/)

END

SUBROUTINE RECIRC
SUBROUTINE RECIRC COMPUTES THE FOLLOWING
1.) JUMP IN P AT TRAILING EDGE = CIRCTE
2.) CIRCULATION FOR FARFIELD BOUNDARY = CIRCFF
3.) JUMP IN P ALONG SLIT Y=0, X =GT.1 BY LINEAR
INTERPOLATION BETWEEN CIRCTE AND CIRCFF
4.) P(JO,ITE) AND P(JO,ITE-1)
CALLED BY - SOLVE.

COMMON /COM1/ P(102,101),X(130) , Y(100)
COMMON /COM1/ IMIN , IMAX , IUP , IODWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT
COMMON /COM2/ AK , ALPHA , DUB , GAM1 , RTK
COMMON /COM7/ CJUP , CJUP1 , CJLOW , CJLOW1
COMMON /COM13/ CDFACT , CLFACT , CNFACT , CPFACT , CPSTAR
LOGICAL FCR , KUTTA
COMMON /COM14/ CLSET , FCR , KUTTA , WCIRC
LOGICAL OUTERR
COMMON /COM18/ ERROR , I1 , I2 , IERROR , JERROR ,
1 OUTER , EMU(100,2) , VC(100) ,
2 MI , DCIRC , POLD(103,2)
COMMON /COM22/ CX(100) , CXL(100) , CXR(100) , CXXC(100) , CXXL(100) ,
1 CXXR(100) , C11(100)
COMMON /COM26/ PJUMP(100)
INTEGER BCTYPE
COMMON /COM28/ BCTYPE , CIRCFF , FHINV , POR , CIRCTE

COMPUTE JUMP IN POTENTIAL AT TRAILING EDGE

CTEOLD=CIRCTE
PUP = CJUP*(JUP,ITE) - CJUP1*(JUP+1,ITE)
PLDW = CJLOW*(JLOW,ITE) - CJLOW1*(JLOW-1,ITE)
CIRCTE = PUP - PLDW
COMPUTE FAR FIELD CIRCULATION
CIRC0 = CIRCFF
IF(KUTTA) CIRCFF = (1.-WCIRC)*CIRC0 + CIRCTE*WCIRC

```

```

READIN 323
READIN 324
READIN 325
READIN 326
READIN 327
READIN 328
READIN 329
READIN 330
READIN 331
READIN 332
READIN 333
READIN 334
READIN 335
READIN 336
READIN 337
READIN 338
READIN 339
READIN 340
READIN 341
READIN 342
READIN 343
READIN 344
READIN 345
READIN 346
READIN 347
READIN 348
READIN 349
READIN 350
READIN 351
READIN 352
READIN 353
READIN 354
READIN 355
READIN 356
READIN 357
READIN 358
READIN 359
READIN 360
READIN 361
READIN 362
READIN 363

```

```

IF(.NOT. KUTTA) CIRCFF = .5*CLSET/CLFACT
FIX JUMP IN P AT AIRFOIL TRAILING EDGE IF KUTTA=.F.
AND LIFT OF AIRFOIL EXCEEDS CLSET. THIS TREATMENT
ASSUMES THAT CL IS INCREASING FROM BELOW CLSET OR
IS NOT TOO MUCH ABOVE CLSET TO START CALCULATION.

IF (.NOT. KUTTA) CIRCTE = CIRCFF
DCIRC=CIRCTE-CTEOLD

C SET JUMP IN P ALONG Y = 0, X =GT. 1
FACTOR = (CIRCFF - CIRCTE)/(X(IMAX) - 1.)
DO 35 I = ITE,IMAX
PJUMP(I) = CIRCTE + (X(I) - 1.) * FACTOR
CONTINUE

RETURN
END

SUBROUTINE REDUB
SUBROUTINE REDUB COMPUTES DOUBLET STRENGTH
FOR LIFTING FREE AIR FLOWS; DOUBLET STRENGTH IS SET
EQUAL TO MODEL VOLUME, FOR OTHER FLOWS, THE NON
LINEAR CONTRIBUTION IS ADDED.
CALLED BY - SOLVE.

COMMON /COM1/ P(102,101),X(130) , Y(100)
COMMON /COM1/ IMIN , IMAX , IUP , IODWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT
COMMON /COM2/ AK , ALPHA , DUB , GAM1 , RTK
COMMON /COM5/ XDIFF(100),YDIFF(100)
COMMON /COM6/ FL(100) , FXL(100) , FU(100) , FXU(100) ,
1 CAMBER(100) , THICK(100),VOL , XFOIL(100) , IFOIL
INTEGER BCTYPE
COMMON /COM28/ BCTYPE , CIRCFF , FHINV , POR , CIRCTE
COMMON /COM30/ XI(100) , ARG(100) , REST(204)

IF(BCTYPE .NE. 1) GO TO 10
IF(BCTYPE .EQ. 1 .AND. ABS(CIRCFF) .LT. .0001) GO TO 10
DUB = VOL
RETURN
CONTINUE

COMPUTE DOUBLE INTEGRAL OF U*U OVER MESH DOMAIN FOR
DOUBLET STRENGTH
U = PX IS CENTERED MIDWAY BETWEEN X MESH POINTS.
FIRST THE INTEGRAL (PX**2)DY IS CALCULATED FOR X
HELD CONSTANT. THUS 1./(X(I+1)-X(I))**2 MAY BE
PULLED OUT OF THE INTEGRAL WHICH IS CALCULATED BY
THE TRAPEZOIDAL RULE. THE X INTEGRATION CORRESPONDS
TO SUMMING THESE INTEGRALS, WHICH LIE MIDWAY BETWEEN
X MESH POINTS, USING A MODIFIED TRAPEZOIDAL RULE.

IEND = IMAX - 1
DBLSUM = 0.
DO 50 I=IMIN,IEND
NARG = 0
DO 30 J = JMIN, JMAX
NARG = NARG + 1
TEMP = P(J,I+1) - P(J,I)
ARG(NARG) = TEMP * TEMP
XI(NARG) = Y(J)
CONTINUE
CALL TRAP(XI,ARG,NARG,SUM)
DBLSUM = DBLSUM + SUM * XDIFF(I+1)
CONTINUE
DBLSUM = GAM1*.25*DBLSUM
DUB = VOL + DBLSUM
RETURN
END

```

```

RECIRC 31
RECIRC 32
RECIRC 33
RECIRC 34
RECIRC 35
RECIRC 36
RECIRC 37
RECIRC 38
RECIRC 39
RECIRC 40
RECIRC 41
RECIRC 42
RECIRC 43
RECIRC 44
RECIRC 45
RECIRC 46
RECIRC 47

REDUB 2
REDUB 3
REDUB 4
REDUB 5
REDUB 6
REDUB 7
REDUB 8
BLANK 2
COM1 2
COM1 3
COM1 4
COM2 2
COM2 2
COM2 2
COM2 2
COM2 3
COM28 2
COM28 3
REDUB 15
REDUB 16
REDUB 17
REDUB 18
REDUB 19
REDUB 20
REDUB 21
REDUB 22
REDUB 23
REDUB 24
REDUB 25
REDUB 26
REDUB 27
REDUB 28
REDUB 29
REDUB 30
REDUB 31
REDUB 32
REDUB 33
REDUB 34
REDUB 35
REDUB 36
REDUB 37
REDUB 38
REDUB 39
REDUB 40
REDUB 41
REDUB 42
REDUB 43
REDUB 44
REDUB 45
REDUB 46
REDUB 47
REDUB 48
REDUB 49

```

APPENDIX

170

```

SUBROUTINE REFINE
ROUTINE TO EXPAND THE X-MESH AND Y-MESH TO
DOUBLE THE NUMBER OF POINTS IN EACH. WILL HAVE
TO BE CALLED TWICE IF MESH WAS HALVED TWICE.
THE P(J,I) MESH IS ALSO FILLED BY
INTERPOLATION.
CALLED BY - TSFOIL.

COMMON / COM1/ P(102,101),X(100) , Y(100)
COMMON / COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT
LOGICAL
COMMON / COM3/ IREF , ABORT , ICUT , KSTEP
LOGICAL
COMMON / COM4/ XIN(100) , YIN(100) , AMESH
COMMON / COM20/ XMID(100) , YMID(100)
INTEGER
COMMON / COM28/ BCTYPE , CIRCFF , FFINV , PDR , CIRCCTE
COMMON / COM30/ PT(100) , REST(304)

JMAX0 = JMAX
IMAX = 2 * (IMAX - IMIN) + IMIN
JMAX = 2 * (JMAX - JMIN) + JMIN + 1
IM2 = IMAX - 2
JM2 = JMAX - 2
IF (IREF .GT. 1) GO TO 30
DO 10 I=IMIN,IMAX
X(I) = XIN(I)
10 CONTINUE
DO 20 J=JMIN,JMAX
Y(J) = YIN(J)
20 CONTINUE
IREF = 0
GO TO 60
30 CONTINUE
DO 40 I=IMIN,IMAX
X(I) = XMID(I)
40 CONTINUE
DO 50 J=JMIN,JMAX
Y(J) = YMID(J)
50 CONTINUE
IREF = 1
60 CONTINUE
CALL ISLIT ( X )
CALL JSLIT ( Y )

SPREAD P(J,I) TO ALTERNATE I(X-MESH) POINTS.
DO 90 J=JMIN,JMAX0
K = IMIN - 1
DO 10 I=IMIN,IMAX,2
K = K + 1
PT(I) = P(J,K)
70 CONTINUE
DO 80 I=IMIN,IMAX,2
P(J,I) = PT(I)
80 CONTINUE
90 CONTINUE

SPREAD P(J,I) TO ALTERNATE J (Y-MESH) POINTS.
DO 130 I=IMIN,IMAX,2
K = JMIN - 1
JE = JLOW - 1
JL = JLOW - 2
DO 95 J=JMIN,JE, 2
K = K + 1
PT(J) = P(K,I)
95 CONTINUE
JST = JUP + 1
DO 100 J=JST,JMAX,2
K = K + 1
PT(J) = P(K,I)
100 CONTINUE
DO 110 J=JMIN,JE,2
P(J,I) = PT(J)
110 CONTINUE
DO 120 J=JST,JMAX,2
P(J,I) = PT(J)
120 CONTINUE
130 CONTINUE

INTERPOLATE TO FILL IN THE MISSING P VALUES.
DO 140 I=IMIN,IM2
PT(I) = (X(I+1)-X(I)) / (X(I+2)-X(I))
140 CONTINUE
DO 150 J=JMIN,JE,2
DO 145 I=IMIN,IM2,2
P(J,I+1) = P(J,I) + PT(I) * (P(J,I+2) - P(J,I))
145 CONTINUE

```

```

REFINE 2
REFINE 3
REFINE 4
REFINE 5
REFINE 6
REFINE 7
REFINE 8
REFINE 9
BLANK 2
COM1 2
COM1 3
COM1 4
COM2 2
COM3 3
COM4 2
COM4 3
COM20 2
COM28 2
COM28 3
REFINE 16
REFINE 17
REFINE 18
REFINE 19
REFINE 20
REFINE 21
REFINE 22
REFINE 23
REFINE 24
REFINE 25
REFINE 26
REFINE 27
REFINE 28
REFINE 29
REFINE 30
REFINE 31
REFINE 32
REFINE 33
REFINE 34
REFINE 35
REFINE 36
REFINE 37
REFINE 38
REFINE 39
REFINE 40
REFINE 41
REFINE 42
REFINE 43
REFINE 44
REFINE 45
REFINE 46
REFINE 47
REFINE 48
REFINE 49
REFINE 50
REFINE 51
REFINE 52
REFINE 53
REFINE 54
REFINE 55
REFINE 56
REFINE 57
REFINE 58
REFINE 59
REFINE 60
REFINE 61
REFINE 62
REFINE 63
REFINE 64
REFINE 65
REFINE 66
REFINE 67
REFINE 68
REFINE 69
REFINE 70
REFINE 71
REFINE 72
REFINE 73
REFINE 74
REFINE 75
REFINE 76
REFINE 77
REFINE 78
REFINE 79
REFINE 80
REFINE 81
REFINE 82

```

```

150 CONTINUE
DO 160 J=JST,JMAX,2
DO 155 I=IMIN,IM2,2
P(J,I+1) = P(J,I) + PT(I) + (P(J,I+2) - P(J,I))
155 CONTINUE
160 CONTINUE
DO 170 J=JMIN,JM2
PT(J) = (Y(J+1)-Y(J)) / (Y(J+2)-Y(J))
170 CONTINUE
DO 190 I=IMIN,IMAX
DO 175 J=JMIN, JL,2
P(J+1,I) = P(J,I) + PT(J) * (P(J+2,I) - P(J,I))
175 CONTINUE
DO 180 J=JST,JM2,2
P(J+1,I) = P(J,I) + PT(J) * (P(J+2,I) - P(J,I))
180 CONTINUE
190 CONTINUE
C USE EXTRAPOLATION FOR JLOW, JUP
D1 = Y(JLOW) - Y(JLOW-1)
D2 = Y(JLOW-1) - Y(JLOW-2)
CL1 = (D1 + D2) / D2
CL2 = D1/D2
D1 = Y(JUP+1) - Y(JUP)
D2 = Y(JUP+2) - Y(JUP+1)
CU1 = (D1 + D2) / D2
CU2 = D1 / D2
DO 200 I = IMIN,IMAX
P(JUP,I) = CU1*P(JUP+1,I) - CU2*P(JUP+2,I)
P(JLOW,I) = CL1*P(JLOW-1,I) - CL2*P(JLOW-2,I)
200 CONTINUE
RETURN
END

```

```

REFINE 83
REFINE 84
REFINE 85
REFINE 86
REFINE 87
REFINE 88
REFINE 89
REFINE 90
REFINE 91
REFINE 92
REFINE 93
REFINE 94
REFINE 95
REFINE 96
REFINE 97
REFINE 98
REFINE 99
REFINE 100
REFINE 101
REFINE 102
REFINE 103
REFINE 104
REFINE 105
REFINE 106
REFINE 107
REFINE 108
REFINE 109
REFINE 110
REFINE 111
REFINE 112
REFINE 113
REFINE 114

RESET 2
RESET 3
RESET 4
RESET 5
RESET 6
BLANK 2
COM1 3
COM1 4
COM2 2
COM3 2
COM24 2
COM24 2
COM24 2
COM28 2
COM28 3
RESET 13
RESET 14
RESET 15
RESET 16
RESET 17
RESET 18
RESET 19
RESET 20
RESET 21
RESET 22
RESET 23
RESET 24
RESET 25
RESET 26
RESET 27
RESET 28
RESET 29
RESET 30
RESET 31
RESET 32

```

APPENDIX

```

SUBROUTINE RESET
SUBROUTINE RESET UPDATES FAR FIELD BOUNDARY
CONDITIONS FOR SUBSONIC FREESTREAM FLOWS.
CALLED BY - SOLVE.

COMMON / COM1/ P(102,101),X(100) , Y(100)
COMMON / COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT
LOGICAL
COMMON / COM2/ ABORT , ALPHA , DUB , GAM1 , RTK
COMMON / COM3/ IREF , ABORT , ICUT , KSTEP
COMMON / COM24/ OTOP(100), DBOT(100),DUP(100), DDOWN(100),
1 VTOP(100), VBOT(100),VUP(100), VDOWN(100)
INTEGER
COMMON / COM28/ BCTYPE , CIRCFF , FFINV , PDR , CIRCCTE

SET BOUNDARY CONDITIONS AT UPSTREAM AND DOWNSTREAM
ENDS.
K = JMIN - KSTEP
DO 10 J = JMIN,JMAX
K = K + KSTEP
IF (J .EQ. JUP) K = K + KSTEP - 1
P(J,IMIN) = CIRCFF*VUP(K) + DUB*DBOT(K)
P(J,IMAX) = CIRCFF*VDOWN(K) + DUB*DDOWN(K)
10 CONTINUE
IF(BCTYPE .NE. 1) GO TO 25
C UPDATE BOUNDARY CONDITIONS ON TOP AND BOTTOM
K = IMIN - KSTEP
DO 20 I = IMIN,IMAX
K = K + KSTEP
P(JMIN,I) = CIRCFF*VBOT(K) + DUB*DBOT(K)
P(JMAX,I) = CIRCFF*VTOP(K) + DUB*DTOP(K)
20 CONTINUE
25 RETURN
END

```



```

SUBROUTINE SAVEP
C
C
C
SAVEP MOVES DATA INTO OLD DATA LOCATIONS AND
WRITES IT ON TAPE3 IF REQUESTED.
CALLED BY - TSFOIL.

COMMON / COM1/ P(102,101),X(100), Y(100)
COMMON / COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
ITE , JMIN , JMAX , JUP , JLOW ,
1 JTOP
COMMON / COM2/ AK , ALPHA , DUB , GAM1 , RTK
LOGICAL ABORT
COMMON / COM3/ IREF , ABORT , ICUT , KSTEP
LOGICAL AMESH
COMMON / COM4/ XIN(100) , YIN(100) , AMESH
COMMON / COM5/ FXL(100) , FYL(100) , FXU(100) ,
1 FYU(100)
COMMON / COM6/ CAMBER(100) , THICK(100) , VOL , XFJL(100) , IFOIL
INTEGER PSTART
LOGICAL PSAVE
COMMON / COM11/ ALPHAD , CLOLD , DELTAD , DUBO , EMACHD ,
1 IMINO , IMAXO , IMAXI , JMINO , JMAXO ,
2 JMAXI , PSAVE , PSTART , TITLE(9) , TITLE(8) ,
3 VOLD , XOLD(100) , YOLD(100)
COMMON / COM12/ F , H , HALFPI , PI , RTKPOR ,
1 TWOP1
LOGICAL PHYS
INTEGER PRTFLO , SIMDEF
COMMON / COM27/ CL , DELTA , DELRT2 , EMACH , EMROOT ,
1 PHYS , PRTFLO , SIMDEF , SONVEL , VFACT ,
2 YFACT
INTEGER BCTYPE
COMMON / COM28/ BCTYPE , CIRCF , FHINV , POR , CIRCTE

RESET PARAMETERS SCALED IN SUBROUTINE SCALE.
ALPHA = ALPHA * VFACT
H = H * YFACT
POR = POR / YFACT
DO 5 J=JMIN,JMAX
YINI(J) = YIN(J) + YFACT
6 CONTINUE
IF (ABORT) RETURN

MOVE RESTART DATA TO OLD BLOCK.
DO 10 I=1,8
TITLEO(I) = TITLE(I)
10 CONTINUE
IMINO = IMIN
JMINO = JMIN
IMAXO = IMAX
JMAXO = JMAX
CLOLD = CL
EMACHO = EMACH
ALPHAO = ALPHA
DELTAD = DELTA
VOLD = VOL
DUBO = DUB

DO 20 I=IMINO,IMAXO
XOLD(I) = X(I)
20 CONTINUE

DO 30 J=JMINO,JMAXO
YOLD(J) = YIN(J)
30 CONTINUE

CHECK TO SEE IF RESTART IS TO BE WRITTEN ON
TAPE3.
IF (.NOT. PSAVE) GO TO 100
40 CONTINUE
WRITE (3,900) TITLED
WRITE (3,901) IMAXO, JMAXO, IMINO, JMINO
WRITE (3,902) CLOLD, EMACHO, ALPHAO, DELTAD, VOLD, DUBO
WRITE (3,902) XOLD(I), I=IMINO,IMAXO
WRITE (3,902) YOLD(J), J=JMINO,JMAXO
DO 50 I=IMINO,IMAXO
WRITE (3,902) (P(J,I), J=JMINO, JMAXO)
50 CONTINUE
100 CONTINUE
RETURN
900 FORMAT(2A10)
901 FORMAT(4L5)
902 FORMAT(8F10.6)

END

```

```

SAVEP 2
SAVEP 3
SAVEP 4
SAVEP 5
SAVEP 6
SAVEP 7
BLANK 2
COM1 2
COM1 3
COM1 4
COM2 2
COM3 2
COM3 3
COM4 2
COM5 3
COM6 2
COM11 2
COM11 3
COM11 4
COM11 5
COM11 6
COM11 7
COM12 2
COM12 3
COM27 2
COM27 3
COM27 4
COM27 5
COM27 6
COM28 2
COM28 3
SAVEP 18
SAVEP 19
SAVEP 20
SAVEP 21
SAVEP 22
SAVEP 23
SAVEP 24
SAVEP 25
SAVEP 26
SAVEP 27
SAVEP 28
SAVEP 29
SAVEP 30
SAVEP 31
SAVEP 32
SAVEP 33
SAVEP 34
SAVEP 35
SAVEP 36
SAVEP 37
SAVEP 38
SAVEP 39
SAVEP 40
SAVEP 41
SAVEP 42
SAVEP 43
SAVEP 44
SAVEP 45
SAVEP 46
SAVEP 47
SAVEP 48
SAVEP 49
SAVEP 50
SAVEP 51
SAVEP 52
SAVEP 53
SAVEP 54
SAVEP 55
SAVEP 56
SAVEP 57
SAVEP 58
SAVEP 59
SAVEP 60
SAVEP 61
SAVEP 62
SAVEP 63
SAVEP 64
SAVEP 65
SAVEP 66
SAVEP 67

```

```

SUBROUTINE SCALE
C
C
C
SUBROUTINE SCALES PHYSICAL VARIABLES TO TRANSONIC
VARIABLES.
IF PHYS = .TRUE., ALL INPUT/OUTPUT QUANTITIES ARE IN
PHYSICAL UNITS NORMALIZED BY FREESTREAM VALUES AND
AIRFOIL CHORD. THIS SUBROUTINE THEN SCALES THE
QUANTITIES TO TRANSONIC VARIABLES BY THE FOLLOWING
CONVENTION
SIMDEF = 1 COLE SCALING
SIMDEF = 2 SPREITER SCALING
SIMDEF = 3 KRUPP SCALING
SIMDEF = 4 USER CHOICE
IF PHYS = .FALSE., INPUT IS ALREADY IN SCALED
VARIABLES AND NO FURTHER SCALING IS DONE.
CALLED BY - TSFOIL.

COMMON / COM1/ P(102,101),X(100), Y(100)
COMMON / COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP
COMMON / COM2/ AK , ALPHA , DUB , GAM1 , RTK
LOGICAL ABORT
COMMON / COM4/ XIN(100) , YIN(100) , AMESH
LOGICAL PSAVE
COMMON / COM11/ ALPHAD , CLOLD , DELTAD , DUBO , EMACHO ,
1 IMINO , IMAXO , IMAXI , JMINO , JMAXO ,
2 JMAXI , PSAVE , PSTART , TITLE(8) , TITLE(9) ,
3 VOLD , XOLD(100) , YOLD(100)
COMMON / COM12/ F , H , HALFPI , PI , RTKPOR ,
1 TWOP1
COMMON / COM13/ CDFACT , CLFACT , CMFACT , CPFAC , CPSTAR
LOGICAL PRTFLO , PHYS
INTEGER PRTFLO , SIMDEF
COMMON / COM27/ CL , DELTA , DELRT2 , EMACH , EMROOT ,
1 PHYS , PRTFLO , SIMDEF , SONVEL , VFACT ,
2 YFACT
INTEGER BCTYPE
COMMON / COM28/ BCTYPE , CIRCF , FHINV , POR , CIRCTE

IF (PHYS) GO TO 50
PHYS = .FALSE. NO SCALING

CDFACT = 1.
CLFACT = 1.
CMFACT = 1.
VFACT = 1.
GO TO 600

PHYS = .TRUE. COMPUTE CONSTANTS
CONTINUE
EMACH2 = EMACH*EMACH
BETA = 1. - EMACH2
DELRT1 = DELTA*(1./3.)
DELRT2 = DELTA*(2./3.)

BRANCH TO APPROPRIATE SCALING
GO TO (100,200,300,400), SIMDEF

SIMDEF = 1
COLE SCALING
CONTINUE
AK = BETA/DELRT2
VFACT = 1./DELRT1
CPFAC = DELRT2
CLFACT = DELRT2
CDFACT = DELRT2*DELTA
CMFACT = DELRT2
VFACT = DELTA*57.295779
GO TO 500

SIMDEF = 2
SPREITER SCALING
CONTINUE
EMROOT = EMACH*(2./3.)
AK = BETA/(DELRT2*EMROOT*EMROOT)
YFACT = 1./(DELRT1*EMROOT)
CPFAC = DELRT2/EMROOT
CLFACT = CPFAC
CMFACT = CPFAC
CDFACT = CPFAC*DELTA
VFACT = DELTA*57.295779
GO TO 500

SIMDEF = 3
KRUPP SCALING

```

```

SCALE 2
SCALE 3
SCALE 4
SCALE 5
SCALE 6
SCALE 7
SCALE 8
SCALE 9
SCALE 10
SCALE 11
SCALE 12
SCALE 13
SCALE 14
SCALE 15
SCALE 16
SCALE 17
SCALE 18
BLANK 2
COM1 2
COM1 3
COM1 4
COM2 2
COM4 2
COM11 2
COM11 3
COM11 4
COM11 5
COM11 6
COM11 7
COM12 2
COM12 3
COM13 2
COM27 2
COM27 3
COM27 4
COM27 5
COM27 6
COM28 2
COM28 3
SCALE 28
SCALE 29
SCALE 30
SCALE 31
SCALE 32
SCALE 33
SCALE 34
SCALE 35
SCALE 36
SCALE 37
SCALE 38
SCALE 39
SCALE 40
SCALE 41
SCALE 42
SCALE 43
SCALE 44
SCALE 45
SCALE 46
SCALE 47
SCALE 48
SCALE 49
SCALE 50
SCALE 51
SCALE 52
SCALE 53
SCALE 54
SCALE 55
SCALE 56
SCALE 57
SCALE 58
SCALE 59
SCALE 60
SCALE 61
SCALE 62
SCALE 63
SCALE 64
SCALE 65
SCALE 66
SCALE 67
SCALE 68
SCALE 69
SCALE 70
SCALE 71
SCALE 72

```

APPENDIX

```

300 CONTINUE
AK = BETA/(DELRT2*EMACH)
YFACT = 1./((DELRT1*EMACH**5)
CPFACT = DELRT2/(EMACH**75)
CLFACT = CPFACT
CMFACT = CPFACT
CDFACT = CPFACT*DELTA
VFACT = DELTA*57.295779
GO TO 500
400 CONTINUE
C SIMDEF = 4
C THIS ADDRESS IS INACTIVE
C USER MAY INSERT SCALING OF OWN CHOICE
C DEFINITION FOR LOCAL MACH NUMBER MUST BE ADJUSTED
C IN EMACH1.
WRITE(6,1000)
1000 FORMAT(34H1ABNORMAL STUP IN SUBROUTINE SCALE/
1 24H SIMDEF=4 IS NOT USEABLE)
500 CONTINUE
SCALE Y MESH
YFACIV = 1.0 / YFACT
DO 502 J=JMIN,JMAX
YIN(J) = YIN(J) * YFACIV
502 CONTINUE
IF (PSTART.EQ. 1) GO TO 505
DO 504 J=JMIN,JMAX
YOLD(J) = YOLD(J) * YFACIV
504 CONTINUE
505 CONTINUE
SCALE TUNNEL PARAMETERS
H = H/YFACT
POR = POR*YFACT
C SCALE ANGLE OF ATTACK
ALPHA = ALPHA/YFACT
600 CONTINUE
CHECK VALUE OF AK FOR DEFAULT.
IF (AK.EQ. 0.) CALL INPERR (7)
C COMPUTE SQUARE ROOT OF AK
RTK = SORT(ABS(AK))
C COMPUTE SONIC VELOCITY
IF (ABS(GAM1).GT..0001) GO TO 999
SONVEL=1.
CPSTAR=C.
RETURN
999 CONTINUE
SONVEL = AK / GAM1
CPSTAR = -2.* SONVEL * CPFACT
RETURN
END
SUBROUTINE SETBC
SUBROUTINE SETBC SETS THE LIMITS ON RANGE OF I AND J
FOR SOLUTION OF THE DIFFERENCE EQUATIONS.
THE BODY SLOPE BOUNDARY CONDITION AT THE CURRENT
X MESH POINTS ON THE BODY ARE MULTIPLIED BY MESH
SPACING CONSTANTS AND ENTERED INTO ARRAYS FXBC AND
FXLBC FOR USE IN SUBROUTINE SYOR.
CALLED BY - TSFOIL.
COMMON / COM1 / IMIN , IMAX , IUP , IDOWN , ILE ,
1 ITE , JMIN , JMAX , JUP , JLOW ,
2 JTOP , JBOT
COMMON / COM2 / AK , ALPHA , DUB , GAM1 , RTK
LOGICAL ABORT
COMMON / COM3 / IREF , ABORT , ICUT , KSTEP
COMMON / COM4 / FXL(100) , FXL(100) , FXU(100) ,
COMMON / COM5 / CAMBER(100) , THICK(100) , WGL , XFOIL(100) , IFOIL
COMMON / COM17 / CYYBLC , CYYBLD , CYYBLU , CYYBUC , CYYBDU ,
1 CYYBUU , FXLBC(100) , FXUBC(100)
INTEGER BCTYPE
COMMON / COM28 / BCTYPE , CIRCFF , PHINV , POR , CIRCFE
INT = 0
IF(AK.LT. 0.) INT = 1

```

SCALE	73
SCALE	74
SCALE	75
SCALE	76
SCALE	77
SCALE	78
SCALE	79
SCALE	80
SCALE	81
SCALE	82
SCALE	83
SCALE	84
SCALE	85
SCALE	86
SCALE	87
SCALE	88
SCALE	89
SCALE	90
SCALE	91
SCALE	92
SCALE	93
SCALE	94
SCALE	95
SCALE	96
SCALE	97
SCALE	98
SCALE	99
SCALE	100
SCALE	101
SCALE	102
SCALE	103
SCALE	104
SCALE	105
SCALE	106
SCALE	107
SCALE	108
SCALE	109
SCALE	110
SCALE	111
SCALE	112
SCALE	113
SCALE	114
SCALE	115
SCALE	116
SCALE	117
SCALE	118
SCALE	119
SCALE	120
SCALE	121
SCALE	122
SCALE	123
SCALE	124
SETBC	2
SETBC	3
SETBC	4
SETBC	5
SETBC	6
SETBC	7
SETBC	8
SETBC	9
SETBC	10
BLANK	2
COM1	2
COM1	3
COM1	4
COM1	5
COM2	2
COM2	3
COM2	4
COM2	5
COM2	6
COM2	7
COM2	8
COM2	9
COM2	10
COM2	11
COM2	12
COM2	13
COM2	14
COM2	15
COM2	16
COM2	17
COM2	18
COM2	19
COM2	20
COM2	21
COM2	22
COM2	23
COM2	24
COM2	25
COM2	26
COM2	27
COM2	28
COM2	29
COM2	30
COM2	31
COM2	32
COM2	33
COM2	34
COM2	35
COM2	36
COM2	37
COM2	38
COM2	39
COM2	40
COM2	41
COM2	42
COM2	43
COM2	44
COM2	45
COM2	46
COM2	47
COM2	48
COM2	49
COM2	50
COM2	51
COM2	52

```

IUP = IMIN + 1 + INT
IDOWN = IMAX - 1 + INT
JINT = 0
IF(BCTYPE.EQ. 1 .AND. AK.GT. 0.) JINT = 1
IF(BCTYPE.EQ. 3) JINT = 1
IF(BCTYPE.EQ. 5 .AND. POR.GT. 1.5) JINT = 1
JBOT = JMIN + JINT
JTOP = JMAX - JINT
C AIRFOIL BODY BOUNDARY CONDITION
C ZERO ELEMENTS IN ARRAYS FOR UPPER AND LOWER BODY
C BOUNDARY CONDITIONS
DO 30 I = IMIN,IMAX
FXLBC(I) = 0.
FXUBC(I) = 0.
30 CONTINUE
ENTER BODY SLOPES AT MESH POINTS ON AIRFOIL
INTO ARRAYS FOR BODY BOUNDARY CONDITIONS
IF(IREF.LC. 0) KSTEP = 1
IF(IREF.EQ. 1) KSTEP = 2
IF(IREF.EQ. 2) KSTEP = 4
NFOIL = ITE - ILE + 1
IF = IFOIL + KSTEP
I = ITE + 1
DO 50 N = 1,NFOIL
I = I-1
IF = IF - KSTEP
FXLBC(I) = CYYBLU*(FXL(IF) - ALPHA)
FXUBC(I) = CYYBUD*(FXU(IF) - ALPHA)
50 CONTINUE
RETURN
END
SUBROUTINE SIMP(R,X,Y,N,ISR)
SUBROUTINE TO INTEGRATE BY SIMPSONS RULE.
CALLED BY BODY.
DIMENSION X(N),Y(N)
R=0.0
IF(N.GT.1) GO TO 1
IER=2
RETURN
1 IF(X(1).EQ.X(2)) GO TO 12
NM1=N-1
IF(N.EQ.2) GO TO 13
IF(X(1).LT.X(2)) GO TO 3
C TEST FOR X TO BE MONOTONICALLY DECREASING
DO 2 I=2,NM1
IF(X(I+1).GE.X(I)) GO TO 12
2 CONTINUE
GO TO 5
C TEST FOR X TO BE MONOTONICALLY INCREASING
3 DO 4 I=2,NM1
IF(X(I+1).LE.X(I)) GO TO 12
4 CONTINUE
5 NM2=N-2
IF(MOD(N,2).EQ.0) GO TO 14
P=0.0
N1=1
6 S1=X(N1+1)-X(N1)
S2=X(N1+2)-X(N1+1)
S3=X(NM1)-X(NM2)
S4=X(N)-X(NM1)
R=(2.*S1**2+S1*S2-S2**2)/S1*Y(N1)+(2.*S4**2+S3*S4-S3**2)/S4*Y(N)
N1=N1+1
DO 7 I=N1,NM2,2
S1=X(I)-X(I-1)
S2=X(I+1)-X(I)
S3=X(I+1)-X(I)
S4=X(I+2)-X(I+1)
R=R+(2.*S2**2+S1*S2-S1**2)/S2+(2.*S3**2+S3*S4-S4**2)/S3*Y(I)
7 R=R/6.*P
10 CONTINUE
IER=1
RETURN

```

SETBC	22
SETBC	23
SETBC	24
SETBC	25
SETBC	26
SETBC	27
SETBC	28
SETBC	29
SETBC	30
SETBC	31
SETBC	32
SETBC	33
SETBC	34
SETBC	35
SETBC	36
SETBC	37
SETBC	38
SETBC	39
SETBC	40
SETBC	41
SETBC	42
SETBC	43
SETBC	44
SETBC	45
SETBC	46
SETBC	47
SETBC	48
SETBC	49
SETBC	50
SETBC	51
SETBC	52
SIMP	2
SIMP	3
SIMP	4
SIMP	5
SIMP	6
SIMP	7
SIMP	8
SIMP	9
SIMP	10
SIMP	11
SIMP	12
SIMP	13
SIMP	14
SIMP	15
SIMP	16
SIMP	17
SIMP	18
SIMP	19
SIMP	20
SIMP	21
SIMP	22
SIMP	23
SIMP	24
SIMP	25
SIMP	26
SIMP	27
SIMP	28
SIMP	29
SIMP	30
SIMP	31
SIMP	32
SIMP	33
SIMP	34
SIMP	35
SIMP	36
SIMP	37
SIMP	38
SIMP	39
SIMP	40
SIMP	41
SIMP	42
SIMP	43
SIMP	44
SIMP	45
SIMP	46
SIMP	47
SIMP	48


```

SUBROUTINE SPLM1 (X,Y,N)
C      CALLED BY - BODY.
COMMON /SPLN/ A(200) , B(200) , DY1 , DY2 , K1
1      K2 , XP , YP , OYP
CONTINUOUS DERIVATIVE INTERPOLATION SUBROUTINES
CURFIT COMPUTES COEFFICIENTS OF CUBICS -- A(I)...I=1,2*N-2
C.....FOR THE WHOLE TABULATED TABLE
C      X(I) = INDEPENDENT VARIABLE.....I=1,N (GIVEN)
C      Y(I) = DEPENDENT VARIABLE.....I=1,N (GIVEN)
C      N = LENGTH OF Y-VS-X TABLE (GIVEN)
C      DY1 = 1ST OR 2ND DERIVATIVE AT LOWER END OF TABLE
C      DY2 = 1ST OR 2ND DERIVATIVE AT UPPER END OF TABLE
C      K1 = 1 .....DY1 = 1ST DERIVATIVE (GIVEN)
C      K1 = 2 .....DY1 = 2ND DERIVATIVE (GIVEN)
C      K2 = 1 .....DY2 = 1ST DERIVATIVE (GIVEN)
C      K2 = 2 .....DY2 = 2ND DERIVATIVE (GIVEN)
DIMENSION X(1), Y(1)
N1=N-2
C1=X(2)-X(1)
IF (K1.EQ. 2) GO TO 4
B(1) = 0.
A(1) = (DY1 - (Y(2) - Y(1))/C1)/C1
GO TO 5
4 B(1) = -C1
A(1) = -DY1/2.
5 J=1
IF(N.EQ.2)GO TO 42
DO 10 I=1,N1
J=J+1
C1=X(I+1)-X(I)
C2=X(I+2)-X(I+1)
C3=Y(I+1)-Y(I)
C4=Y(I+2)-Y(I+1)
C5=C3/C1-C4/C2
C6=C1/C2
C7=C1/C2
B(J)=1.0/(C6*(C1-B(J-1)))
A(J)=(C5/C2-C6*(J-1))*B(J)
J=J+1
B(J)=1.0/((-C1-C2)/C7-C6*B(J-1))
A(J) = (-C5/C7-C6*(J-1))*B(J)
10 CONTINUE
IF(K2.EQ.2)GO TO 30
A(J+1) = (DY2-C4/C2+C2*A(J))/(C2*(B(J)-C2))
GO TO 45
30 A(J+1)=(DY2/2.+A(J))/(-2.*C2+B(J))
GO TO 45
C STATEMENTS 42 TO 44 ARE FOR N=2 ONLY
42 C3=K1
C2=1.0/C3
IF(K2.EQ.2)GO TO 44
A(J+1) = (Y(2)-Y(1))/C1-A(J)*C1-DY2/(C1*C1)*C2
GO TO 45
44 A(J+1)=C3*(DY2+2.*A(J))/(4.0*C1)
45 J=2*(N-1)
5C J=J-1
IF(J.LE.0)RETURN
A(J) = A(J)-B(J)*A(J+1)
GO TO 50
C
C ENTRY POINT FOR INTERPOLATION
ENTRY SPLN1X
C
IF(XP.GT.X(1)) GO TO 11
SPECIAL CASE FOR EXTRAPOLATION BEYOND LOWER END OF X-TABLE
C=X(2)-X(1)
DYP=(Y(2)-Y(1))/C+A(1)*C
YP=Y(1)+DYP*(XP-X(1))
RETURN
11 IF(XP.LT.X(N)) GO TO 13
SPECIAL CASE FOR EXTRAPOLATION BEYOND UPPER END OF Y-TABLE
C=X(N)-X(N-1)
DYP=(Y(N)-Y(N-1))/C-A(2*N-3)*C-A(2*N-2)*C*C
YP=Y(N)+DYP*(XP-X(N))
RETURN
13 I=1
14 I=I+1
IF(X(I).LT.XP) GO TO 14
NOW XP HAS BEEN BRACKETED SO THAT X(I-1).LT.XP.LE.X(I)
C=XP-X(I-1)
D=X(I)-XP

```

SPLN1 2
SPLN1 3
SPLN1 4
SPLN1 5
SPLN1 6
SPLN1 7
SPLN1 8
SPLN1 9
SPLN1 10
SPLN1 11
SPLN1 12
SPLN1 13
SPLN1 14
SPLN1 15
SPLN1 16
SPLN1 17
SPLN1 18
SPLN1 19
SPLN1 20
SPLN1 21
SPLN1 22
SPLN1 23
SPLN1 24
SPLN1 25
SPLN1 26
SPLN1 27
SPLN1 28
SPLN1 29
SPLN1 30
SPLN1 31
SPLN1 32
SPLN1 33
SPLN1 34
SPLN1 35
SPLN1 36
SPLN1 37
SPLN1 38
SPLN1 39
SPLN1 40
SPLN1 41
SPLN1 42
SPLN1 43
SPLN1 44
SPLN1 45
SPLN1 46
SPLN1 47
SPLN1 48
SPLN1 49
SPLN1 50
SPLN1 51
SPLN1 52
SPLN1 53
SPLN1 54
SPLN1 55
SPLN1 56
SPLN1 57
SPLN1 58
SPLN1 59
SPLN1 60
SPLN1 61
SPLN1 62
SPLN1 63
SPLN1 64
SPLN1 65
SPLN1 66
SPLN1 67
SPLN1 68
SPLN1 69
SPLN1 70
SPLN1 71
SPLN1 72
SPLN1 73
SPLN1 74
SPLN1 75
SPLN1 76
SPLN1 77
SPLN1 78
SPLN1 79
SPLN1 80
SPLN1 81
SPLN1 82
SPLN1 83
SPLN1 84
SPLN1 85
SPLN1 86
SPLN1 87

```

K=2*I-3
SLOPE=(Y(I)-Y(I-1))/(X(I)-X(I-1))
YP=Y(I-1)+(SLOPE+(A(K)+A(K+1)*C1)*D)*C
DYP=SLOPE+A(K)*D-C1+A(K+1)*(2.*D-C)*C
RETURN
END

```

SPLN1 88
SPLN1 89
SPLN1 90
SPLN1 91
SPLN1 92
SPLN1 93

SYORNDX

```

SUBROUTINE SYOR
C      SYOR COMPUTES NEW P AT ALL MESH POINTS.
C      CALLED BY - SOLVE.
COMMON /COM1/ IMIN , IMAX , IUP , IDOWN , ILE ,
1      ITE , JMIN , JMAX , JUP , JLOW ,
2      JTOP , JBOT
COMMON /COM2/ AK , ALP+A , DUB , GAM1 , RTK
COMMON /COM3/ CVERGE , DVERGE , IPRTER , MAXIT ,
1      WE(3) , EPS
LOGICAL FCR , KUTTA
COMMON /COM4/ CLSET , FCR , KUTTA , WCIRC
COMMON /COM17/ CYBLC , CYBLD , CYBLU , CYBUC , CYBUO ,
1      CYBUU , FXLBC(100),FXUBC(100)
LOGICAL OUTERR , IZ , IERRR , JERRR ,
COMMON /COM18/ OUTERR , EMU(100,2) , VC(100) ,
2      WI , DCIRC , POLD(100,2)
COMMON /COM19/ DIAG(100),RHS(100),SUB(100),SUP(100)
COMMON /COM22/ CX(100),CXL(100),CXI(100),CXK(100),CXKX(100),
1      CXR(100),CL(100)
COMMON /COM23/ CYC(100),CYD(100),CYU(100),IVAL
COMMON /COM25/ PJUMP(100)
COMMON /COM32/ BIGRL , IRL , JRL
DIMENSION SAVE(100)
IM2=IUP-1
IF( AK .LT. 0.0 ) IM2=IUP-2
J1 = JBOT + 1
J2 = JTOP - JBOT
DO 200 I=IUP, IDOWN
EPSX = EPS/(X(I)-X(I-1))*2)
C COMPUTE VC = 1 - M**2
DO 1C J=JBOT,JTOP
VC(J) = C1(I) - (CXK(I)*POLD(J,I2) + CXK(I)*P(J,I)
1      + CXR(I)*P(J,I+1))
EMU(J,I) = 0.0
POLD(J,I) = P(J,I)
10 CONTINUE
DO 20 J=JBOT,JTOP
IF (VC(J) .LT. 0.0) EMU(J,I) = VC(J)
20 CONTINUE
IF ( FCR ) GO TO 22
DO 21 J=JBOT,JTOP
EMU(J,I2) = EMU(J,I1)
21 CONTINUE
22 CONTINUE
C COMPUTE ELEMENTS OF MATRIX
DO 30 J=JBOT,JTOP
DIAG(J) = (EMU(J,I1) - VC(J)) * CXK(I) * WI
1      + EMU(J,I2) * CXK(I-1) - CYC(J)
SUP(J)=CYD(J)
SUB(J)=CYU(J)
30 CONTINUE
C COMPUTE RESIDUAL
DO 40 J=JBOT,JTOP
RHS(J) = -(VC(J)-EMU(J,I1))*
1      (CXK(I)*P(J,I-1) - CXK(I)*P(J,I) + CXK(I)*P(J,I+1))
40 CONTINUE
DO 50 J=JBOT,JTOP
RHS(J) = RHS(J) - (EMU(J,I2) * (CXK(I-1)*P(J,IM2)

```

SYOR 2
SYOR 3
SYOR 4
SYOR 5
BLANK 2
COM1 2
COM1 3
COM1 4
COM2 2
COM3 2
COM4 2
COM4 3
COM17 2
COM17 3
COM18 2
COM18 3
COM18 4
COM18 5
COM19 2
COM22 2
COM22 3
COM23 2
COM25 2
COM26 2
COM26 2
SYOR 18
SYOR 19
SYOR 20
SYOR 21
SYOR 22
SYOR 23
SYOR 24
SYOR 25
SYOR 26
SYOR 27
SYOR 28
SYOR 29
SYOR 30
SYOR 31
SYOR 32
SYOR 33
SYOR 34
SYOR 35
SYOR 36
SYOR 37
SYOR 38
SYOR 39
SYOR 40
SYOR 41
SYOR 42
SYOR 43
SYOR 44
SYOR 45
SYOR 46
SYOR 47
SYOR 48
SYOR 49
SYOR 50
SYOR 51
SYOR 52
SYOR 53
SYOR 54
SYOR 55
SYOR 56
SYOR 57
SYOR 58
SYOR 59
SYOR 60
SYOR 61
SYOR 62
SYOR 63

```

1      - CXXC(I-1)*P(J,I-1) + CXXR(I-1)*P(J,I))
50 CONTINUE
JA = JBOT + 1
JB = JTOP - 1
DO 60 J=JA,JB
RHS(J) = RHS(J) - (CYD(J)*P(J-1,I) - CYC(J)*P(J,I)
+ CYU(J)*P(J+1,I))
1
60 CONTINUE
RHS(JBOT) = RHS(JBOT) - (-CYC(JBOT)*P(JBOT,I)
+ CYU(JBOT)*P(JBOT+1,I))
1
IF (JBOT.EQ. JMIN) GO TO 61
RHS(JBOT) = RHS(JBOT) - CYD(JBOT)*P(JBOT-1,I)
61 CONTINUE
RHS(JTOP) = RHS(JTOP) - (CYD(JTOP)*P(JTOP-1,I)
- CYC(JTOP)*P(JTOP,I))
1
IF (JTOP.EQ. JMAX) GO TO 62
RHS(JTOP) = RHS(JTOP) - CYU(JTOP)*P(JTOP+1,I)
62 CONTINUE

C          CHECK FOR AIRFOIL B.C. AND KUTTA SLICE.
IF (I.LT. ILE) GO TO 80
IF (I.GT. ITE) GO TO 70

C          AIRFOIL B. C.
J=JUP
DIAG(J)=DIAG(J)+CYC(J)-CYBUC
SUP(J)=0.0
SUB(J)=CYBUU
RHS(J)=RHS(J)+CYD(J)*P(J-1,I)-CYC(J)*P(J,I)+CYU(J)*P(J+1,I)
-(-CYBUC*P(J,I)+CYBUU*P(J+1,I)+FXUBC(I))
1
J=JLW
DIAG(J)=DIAG(J)+CYC(J)-CYBLC
SUP(J)=CYBLD
SUB(J)=0.0
RHS(J)=RHS(J)+CYD(J)*P(J-1,I)-CYC(J)*P(J,I)+CYU(J)*P(J+1,I)
-(-CYBLC*P(J,I)+CYBLD*P(J+1,I)+FXLBC(I))
GO TO 80

C          KUTTA SLICE CHANGE.
70 CONTINUE
RHS(JLW) = RHS(JLW) + CYU(JLW)*PJUMP(I)
RHS(JUP) = RHS(JUP) - CYD(JUP)*PJUMP(I)
80 CONTINUE

C          INSERT #ALL B. C.
IVAL = 1
CALL BCEND

C          COMPUTE MAX RESIDUAL.
IF (.NOT. OUTERR) GO TO 110
DO 100 J=JBOT,JTOP
ARHS = ABS(RHS(J))
IF (ARHS.GT. BIGRL) GO TO 90
GO TO 100
90 CONTINUE
BIGRL = ARHS
IRL = I
JRL = J
100 CONTINUE
110 CONTINUE

C***** ADD PXT
C
DO 300 J = JBOT,JTOP
DIAG(J) = DIAG(J) - EPSX
300 RHS(J) = RHS(J) - EPSX*(P(J,I-1)-POLD(J,I2))

C          SOLVE TRIDIAGONAL MATRIX EQUATION.
DNOM = 1.0 / DIAG(JBOT)
SAVE(JBOT)=SUB(JBOT)*DNOM
RHS(JBOT) = RHS(JBOT) * DNOM
DO 120 J=JL,JTOP
DNOM=1.0/(DIAG(J)-SUP(J)*SAVE(J-1))
SAVE(J)=SUB(J)*DNOM
RHS(J)=(RHS(J)-SUP(J)*RHS(J-1))*DNOM
120 CONTINUE
DO 130 K=1,J2
J = JTOP - K
RHS(J) = RHS(J) - SAVE(J) * RHS(J+1)
130 CONTINUE

```

```

SYOR 64
SYOR 65
SYOR 66
SYOR 67
SYOR 68
SYOR 69
SYOR 70
SYOR 71
SYOR 72
SYOR 73
SYOR 74
SYOR 75
SYOR 76
SYOR 77
SYOR 78
SYOR 79
SYOR 80
SYOR 81
SYOR 82
SYOR 83
SYOR 84
SYOR 85
SYOR 86
SYOR 87
SYOR 88
SYOR 89
SYOR 90
SYOR 91
SYOR 92
SYOR 93
SYOR 94
SYOR 95
SYOR 96
SYOR 97
SYOR 98
SYOR 99
SYOR 100
SYOR 101
SYOR 102
SYOR 103
SYOR 104
SYOR 105
SYOR 106
SYOR 107
SYOR 108
SYOR 109
SYOR 110
SYOR 111
SYOR 112
SYOR 113
SYOR 114
SYOR 115
SYOR 116
SYOR 117
SYOR 118
SYOR 119
SYOR 120
SYOR 121
SYOR 122
SYOR 123
SYOR 124
SYOR 125
SYOR 126
SYOR 127
SYOR 128
SYOR 129
SYOR 130
SYOR 131
SYOR 132
SYOR 133
SYOR 134
SYOR 135
SYOR 136
SYOR 137
SYOR 138
SYOR 139
SYOR 140
SYOR 141
SYOR 142
SYOR 143
SYOR 144
SYOR 145
SYOR 146
SYOR 147
SYOR 148
SYOR 149

```

```

C          COMPUTE NEW P.
DO 140 J=JBOT,JTOP
P(J,I) = P(J,I) + RHS(J)
140 CONTINUE

C          COMPUTE MAX ERROR
IF (.NOT. OUTERR) GO TO 180
DO 170 J = JBOT,JTOP
ARHS = ABS(RHS(J))
IF (ARHS.GT. ERROR) GO TO 163
GO TO 170
160 CONTINUE
ERROR = ARHS
IERROR = I
JERROR = J
170 CONTINUE
180 CONTINUE
IF (AK.GT. 0.0) GO TO 195
IF (I.NE. IDOWN-1) GO TO 195

C          SET P(IDOWN+1) = P(IDOWN-1) TO OBTAIN
C          CENTERED VELOCITY AT IDOWN FOR SUPERSINIC
C          FREESTREAM FLOW.
DO 190 J=JMIN,JMAX
P(J,IDOWN+1) = P(J,IDOWN-1)
190 CONTINUE
195 CONTINUE
ISAVE = I2
I2 = I1
I1 = ISAVE
200 CONTINUE
RETURN
END

SUBROUTINE TRAP(X,Y,N,SUM)
C          INTEGRATE Y DX BY TRAPEZODIAL RULE
C          N IS THE NUMBER OF (X,Y) POINTS AND SUM IS THE
C          RESULTING INTEGRAL.
C          CALLED BY - COCDLE, DRAG, PITCH, REDUB.
C          INTEGRAL
C          DIMENSION X(1),Y(1)
SUM = 0.
NM1 = N-1
DO 10 I=1,NM1
Z = X(I+1) - X(I)
W = Y(I+1) + Y(I)
SUM = SUM + Z*W
10 CONTINUE
SUM = .5*SUM
RETURN
END

SUBROUTINE VROOTS
C          COMPUTE CONSTANTS BETA0,BETA1,BETA2,PSIO,PSI1,PSI2,
C          USED IN FORMULA FOR VORTEX IN SLOTTED WIND TUNNEL
C          WITH SUBSONIC FREESTREAM
C          CALLED BY - FARFLD.
COMMON /COM12/ F , H , HALFP1 , PI , RTKPOR ,
1 COMMON /COM15/ B , BETA0 , BETA1 , BETA2 , PSIO ,
1 PSI1 , PSI2
ERROR = .00001
C          CALCULATE BETA0
BETA0 = 0.
DO 10 I = 1,100

```

```

SYOR 150
SYOR 151
SYOR 152
SYOR 153
SYOR 154
SYOR 155
SYOR 156
SYOR 157
SYOR 158
SYOR 159
SYOR 160
SYOR 161
SYOR 162
SYOR 163
SYOR 164
SYOR 165
SYOR 166
SYOR 167
SYOR 168
SYOR 169
SYOR 170
SYOR 171
SYOR 172
SYOR 173
SYOR 174
SYOR 175
SYOR 176
SYOR 177
SYOR 178
SYOR 179
SYOR 180
SYOR 181
SYOR 182
SYOR 183
SYOR 184
SYOR 185
SYOR 186

TRAP 2
TRAP 3
TRAP 4
TRAP 5
TRAP 6
TRAP 7
TRAP 8
TRAP 9
TRAP 10
TRAP 11
TRAP 12
TRAP 13
TRAP 14
TRAP 15
TRAP 16
TRAP 17
TRAP 18

VROOTS 2
VROOTS 3
VROOTS 4
VROOTS 5
VROOTS 6
VROOTS 7
COM12 2
COM12 3
COM12 3
COM15 3
VROOTS 10
VROOTS 11
VROOTS 12
VROOTS 13
VROOTS 14

```

```

TEMP = BETA0
Q = -F*TEMP + RTKPOR
BETA0 = ATAN(Q)
DBETA = ABS(TEMP - BETA0)
IF(DBETA .LT. ERROR) GO TO 19
10 CONTINUE
N = 0
GO TO 9999
15 CONTINUE
      CALCULATE BETA1
BETA1 = 0.
DO 20 I=1,100
TEMP = BETA1
Q = -F*(TEMP + PI) + RTKPOR
BETA1 = ATAN(Q)
DBETA = ABS(BETA1 - TEMP)
IF(DBETA .LT. ERROR) GO TO 25
20 CONTINUE
N = 1
GO TO 9999
25 CONTINUE
      CALCULATE BETA2
BETA2 = 0.
DO 30 I=1,100
TEMP = BETA2
Q = -F*(TEMP - PI) + RTKPOR
BETA2 = ATAN(Q)
DBETA = ABS(BETA2 - TEMP)
IF(DBETA .LT. ERROR) GO TO 35
30 CONTINUE
N = 2
GO TO 9999
35 CONTINUE
      COMPUTE PSIO,PSI1,PSI2
TEMP = TAN(BETA0)
PSIO = (1. + F/(1. + TEMP*TEMP))
PSIO = 1.0 / PSIO
TEMP = TAN(BETA1)
PSI1 = (1. + F/(1. + TEMP*TEMP))
PSI1 = 1.0 / PSI1
TEMP = TAN(BETA2)
PSI2 = (1. + F/(1. + TEMP*TEMP))
PSI2 = 1.0 / PSI2
RETURN
C
9999 CONTINUE
      ABNORMAL STOP IF ITERATIONS FOR BETAS DID NOT CONVERGE
WRITE(6,1000) N
1000 FORMAT(35H1ABNORMAL STOP IN SUBROUTINE VROOTS/
1 37H0NONCONVERGENCE OF ITERATION FOR BETA, I1)
STOP
END
VROOTS 15
VROOTS 16
VROOTS 17
VROOTS 18
VROOTS 19
VROOTS 20
VROOTS 21
VROOTS 22
VROOTS 23
VROOTS 24
VROOTS 25
VROOTS 26
VROOTS 27
VROOTS 28
VROOTS 29
VROOTS 30
VROOTS 31
VROOTS 32
VROOTS 33
VROOTS 34
VROOTS 35
VROOTS 36
VROOTS 37
VROOTS 38
VROOTS 39
VROOTS 40
VROOTS 41
VROOTS 42
VROOTS 43
VROOTS 44
VROOTS 45
VROOTS 46
VROOTS 47
VROOTS 48
VROOTS 49
VROOTS 50
VROOTS 51
VROOTS 52
VROOTS 53
VROOTS 54
VROOTS 55
VROOTS 56
VROOTS 57
VROOTS 58
VROOTS 59
VROOTS 60
VROOTS 61
VROOTS 62
VROOTS 63
VROOTS 64
VROOTS 65

```

REFERENCES

1. Murman, E. M., Bailey, F. R., and Johnson, M. H.: TSFOIL - A Computer Code for 2-D Transonic Calculations, Including Wind-Tunnel Wall Effects and Wave-Drag Evaluation. NASA SP-347, Part II, 1975, pp. 769-788.
2. Krupp, J. A.: The Numerical Calculation of Plane Steady Transonic Flows Past Thin Lifting Airfoils. Ph.D. Dissertation, University of Washington, 1971.
3. Krupp, J. A. and Murman, E. M.: Computation of Transonic Flows Past Lifting Airfoils and Slender Bodies. AIAA Journal, vol. 10, no. 7, July 1972, pp. 880-886.
4. Bailey, F. R. and Ballhaus, W. F.: Relaxation Methods for Transonic Flow About Wing-Cylinder Combinations and Lifting Swept Wings. Lecture Notes in Physics, vol. 19, Springer-Verlag, 1972, pp. 2-9.
5. Ballhaus, W. F. and Bailey, F. R.: Numerical Calculation of Transonic Flow About Swept Wings. AIAA Paper No. 72-677, June 1972.
6. Murman, E. M.: Analysis of Embedded Shock Waves Calculated by Relaxation Methods. AIAA Journal, vol. 12, no. 5, May 1974, pp. 626-632.
7. Murman, E. M. and Cole, J. D.: Inviscid Drag at Transonic Speeds. AIAA Paper No. 74-540, June 1974.
8. Murman, E. M.: Computation of Wall Effects in Ventilated Transonic Wind Tunnels. AIAA Paper No. 72-1007, Sept. 1972.
9. Murman, E. M.: A Relaxation Method for Calculating Transonic Flows with Detached Bow Shocks. Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics. Lecture Notes in Physics, vol. 19, July 1972, pp. 201-206.
10. Lock, R. C.: Research in the UK on Finite Difference Methods for Computing Steady Transonic Flows. RAE Tech. Memo. Aero. 1644, July 1975.
11. Albone, C. M., Catherall, D., Hall, M. G., and Joyce, G.: An Improved Numerical Method for Solving the Transonic Small-Perturbation Equation for the Flow Past a Lifting Airfoil, RAE Tech. Rept. 74056, August 1974.
12. Jameson, A.: Transonic Flow Calculation, Lecture Series 87 von Karman Institute for Fluid Dynamics, Computational Fluid Dynamics

14. Olsen, J.: Subsonic and Transonic Flow Over Sharp and Round Nosed Nonlifting Airfoils. Ohio State Univ. Ph.D. Thesis, 1976.
15. Kacprzyński, J. J., Ohman, L., Garabedian, P., and Korn, D.: Analysis of the Flow Past a Shockless Lifting Airfoil in Design and Off Design Conditions. Aeronautical Report LR-554, National Research Council of Canada, Nov. 1971.
16. Baldwin, B., Turner, J., and Knechtel, E.: Wall Interference in Wind Tunnels with Slotted and Porous Boundaries at Subsonic Speeds. NACA TN 3176, 1954.
17. Collins, D. J. and Krupp, J. A.: Experimental and Theoretical Investigations in Two-Dimensional Transonic Flow. AIAA Journal, vol. 12, no. 6, June 1974, pp. 771-778.
18. Bailey, F. R. and Ballhaus, W. F.: Comparisons of Computed and Experimental Pressures for Transonic Flows About Isolated Wings and Wing-Fuselage Configurations. NASA SP-347, Part I, 1975, pp. 1213-1232.
19. Murman, E. M.: Analysis of Embedded Shock Waves Calculated by Relaxation Methods. Proceedings of AIAA Computational Fluid Dynamics Conference, Palm Springs, CA, July 1973, pp. 27-40.
20. Murman, E. M. and Cole, J. D.: Calculation of Plane Steady Transonic Flows. AIAA Journal, vol. 9, no. 1, Jan. 1971, pp. 114-121.
21. Bailey, F. R.: On the Computation of Two- and Three-Dimensional Steady Transonic Flows by Relaxation Methods. VKI Lecture Series "Progress in Numerical Fluid Dynamics," Von Karman Institute for Fluid Dynamics, Rhode-St-Genese, Belgium, Feb. 11-15, 1974.
22. Lock, R. C.: Test Cases for Numerical Methods in Two-Dimensional Transonic Flow. AGARD Rept. No. 575, Nov. 1970.
23. Steger, J. L. and Lomax, H.: Transonic Flow About Two-Dimensional Airfoils by Relaxation Procedures. AIAA Journal, vol. 10, no. 1, 1972, pp. 49-54.
24. Garabedian, P. R. and Korn, D. G.: Analysis of Transonic Airfoils. Comm. Pure Applied Math., vol. XXIV, 1971, pp. 841-851.
25. Sells, C. C. L.: Plane Subcritical Flow Past a Lifting Airfoil. RAE TR 67146, June 1967.
26. Jameson, A.: Iterative Solution of Transonic Flows Over Airfoils and Wings, Including Flows at Mach 1. Comm, Pure Applied Math., vol. 27, 1974, pp. 283-309.
27. Yoshihara, H.: A Survey of Computational Methods for 2D and 3D Transonic Flows with Shocks. Advances in Fluid Dynamics, AGARD Lecture Series No. 64, 1973, pp. 6-1 to 6-35.
28. Magnus, R. M.: The Direct Comparison of the Relaxation Method and the Pseudo-Unsteady Finite Difference Method for Calculating Steady Planar Transonic Flow. TN-73-SP03, General Dynamics-Convair Aerospace Division, 1973.

29. 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.
30. Osborne, J.: A Selection of Measured Transonic Flow Pressure Distributions for the NACA 0012 Airfoil: Provisional Data from an NPL Tunnel. Aero. Dept., RAE, June 1971.
31. Hafez, M. M. and Cheng, H. K.: Convergence Acceleration and Shock Fitting for Transonic Aerodynamic Computations. AIAA Paper No. 75-51, Jan. 1975.
32. Lomax, H. and Martin, E. D.: Fast Direct Numerical Solution of the Nonhomogeneous Cauchy-Riemann Equations. Journal Comp. Phys., vol. 15, 1974, pp. 55-80.
33. Martin, E. D.: A Fast Semidirect Method for Computing Transonic Aerodynamic Flows. AIAA 2nd Computational Fluid Dynamics Conference (Proceedings), Hartford, CT, June 19-20, 1975, pp. 162-174.
34. Hafez, M. M. and Cheng, H. K.: Shock-Fitting Applied to Relaxation Solutions of Transonic Small-Disturbance Equations. AIAA Jour. vol. 15, no. 6, June 1977, pp. 786-793.

1. Report No. NASA CR-3064	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Operational Manual for Two-Dimensional Transonic Code TSFOIL		5. Report Date December 1978	6. Performing Organization Code
		8. Performing Organization Report No. NEAR TR 94	10. Work Unit No.
7. Author(s) Stephen S. Stahara		11. Contract or Grant No. NAS2-8648	13. Type of Report and Period Covered Contractor Report
9. Performing Organization Name and Address Nielsen Engineering & Research, Inc. 510 Clyde Avenue Mountain View, California 94043		14. Sponsoring Agency Code	
		12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D. C. 20546	
15. Supplementary Notes			
16. Abstract <p>A detailed operating manual is presented for the computer code TSFOIL recently developed by Murman, Bailey, and Johnson at NASA/Ames Research Center. This code solves the two-dimensional, transonic, small-disturbance equations for flow past lifting airfoils in both free air and various wind-tunnel environments by using a variant of the finite-difference method initially proposed by Murman and Cole. A description of the theoretical and numerical basis of the code is provided, together with complete operating instructions and sample cases for the general user. In addition, a programmer's manual is also presented to assist the user interested in modifying the code. Included in the programmer's manual are a dictionary of subroutine variables in common and a detailed description of each subroutine.</p>			
17. Key Words (Suggested by Author(s)) Transonic Theory Steady Flow Airfoils Wind Tunnel Flows Finite-Difference Relaxation		18. Distribution Statement UNCLASSIFIED-UNLIMITED Cat. 02	
19. Security Classif. (of this report) UNCLASSIFIED	20. Security Classif. (of this page) UNCLASSIFIED	21. No. of Pages 179	22. Price* \$9.00