NASA CONTRACTOR REPORT

NASA CR-2821

TRANDES: A FORTRAN PROGRAM FOR TRANSONIC AIRFOIL ANALYSIS OR DESIGN

Leland A. Carlson

Prepared by
TEXAS A&M UNIVERSITY
College Station, Tex. 77843
for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JUNE 1977

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-2821		

4. Title and Subtitle TRANDES: A Fortran Program for Transonic Airfoil Analysis or Design	5. Report Date June 1977
	6. Performing Organization Code
7. Author(s) Leland A. Carlson	8. Performing Organization Report No.
	10. Work Unit No. 505-06-33-08-00
9. Performing Organization Name and Address Texas A&M University College Station, Texas 77843	11. Contract or Grant No. NSG-1174
	13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

Final report

Langley Technical Monitor: Jerry C. South, Jr.

16. Abstract

A program called TRANDES is presented that can be used for the analysis of steady, irrotational, transonic flow over specified two-dimensional airfoils in free air or for the design of airfoils having a prescribed pressure distribution, including the effects of weak viscous interaction (i.e., massive separation is not considered). Instructions on program usage, listings of the program, and sample cases are given.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Supercritical airfoil design Viscous interaction Relaxation Cartesian coordinates	Unclassified - Unlimited Subject Category 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	109	$5.50

*For sale by the National Technical Information Service, Springfield, Virginia 22161

TRANDES: A FORTRAN PROGRAM FOR TRANSONIC AIRFOIL ANALYSIS OR DESIGN

By
Leland A. Carlson
Texas A&M University

SUMMARY

A program called TRANDES is presented that can be used for the analysis of steady, irrotational, transonic flow over specified two-dimensional airfoils in free air or for the design of airfoils having a prescribed pressure distribution, including the effects of weak (no massive separation) viscous interaction. Instructions on program usage, listings of the program, and sample cases are given.

INTRODUCTION

The program described in this report can be used for either the analysis of the flowfield about an airfoil in free air or for the design of an airfoil having a specified pressure distribution. In the direct or analysis mode the airfoil shape is prescribed and the flowfield and surface pressures are determined. In the direct-inverse or design mode an initial nose shape is given along with the pressure distribution on the remainder of the airfoil, and the flowfield and actual airfoil shape are computed. In either case, the effects of weak viscous interaction may be included at the option of the user.

The program solves the exact equation for the perturbation potential in a Cartesian coordinate system. Most of the background about the equations solved, formulation of the boundary conditions, and the difference scheme used is given in references 1-3. This report gives instructions on the use of the computer program and also some additional details concerning the inclusion of weak viscous interaction. It should be noted that in this report the term weak viscous interaction implies that there is no massive boundary layer separation on the airfoil. Nevertheless, for aft-loaded airfoils at transonic speeds, the effect of viscous interaction on airfoil performance may still be quite large.

The next section gives a general description of the problem and the method of solution. Then the instructions for using the computer program and a description of the inputs and outputs are given. The appendices contain additional details as well as listings of the program and the sample cases.

GENERAL DESCRIPTION

The program described in this paper obtains the inviscid flowfield by solving the full, inviscid, perturbation-potential flow equation in a cartesian grid system. This system which is usually aligned relative to the airfoil chord line, has been found to efficiently yield accurate solutions for biconvex, conventional, and aft-cambered airfoils. In the program, a stretching is applied to the coordinates such that the infinite physical plane is mapped to a finite computational space. Thus, the boundary conditions at infinity can be applied directly and there is no need for an asymptotic far-field solution. Details about the stretching functions are given in appendix A.

The method of solution is to replace the governing second-order partial differential equation with a non-conservative system of finite difference equations that includes, at supersonic points, a form of Jameson's "rotated" difference scheme (ref.4). The difference equations are then solved by column relaxation, which in order to obtain rapid convergence is usually done on several different grids. In the analysis case, the difference equations are first solved on a very coarse grid (typically 13x7). The solution is then interpolated and used as an initial condition for a coarse grid (typically 25x13). This procedure can be repeated twice more to obtain solutions on a medium grid (49x25) and on a fine grid (97X49). The latter has 130 points on the airfoil; however, excellent results are usually obtained on the medium grid, particularly considering the computer time involved. For typical examples see reference 1.

In the inverse case, which is normally used for airfoil modification or design, an initial airfoil shape must be assumed. However, this choice is not critical, and the final airfoil shape may be considerably different. Since experience indicates that the inverse scheme works best if the perturbation potentials have reasonable initial values, fifty relaxation cycles are first performed in the direct mode for the initial airfoil shape on a very coarse grid. The grid is then halved and the inverse procedure initiated using the input pressure distribution as the boundary condition in the inverse region. As in the direct case, the grid may be refined again to the medium grid (typically 49x25) where the results are usually adequate. Fine grid usage in the inverse case is not recommended due to slow convergence. On each grid the airfoil shape is recomputed every ten relaxation cycles after the first fifty.

After the solution has been obtained in the design case, the resultant shpae is treated as a displacement surface and the displacement thickness is automatically subtracted to obtain the actual airfoil surface. The boundary layer characteristics are determined by the Nash-Macdonald method (ref.5) with smoothing.

The effects of viscous interaction may also be included in analysis cases at the option of the user. To preserve numerical consistency with the inverse scheme, the Nash-Macdonald method is also used in such analysis cases, starting with the 50th cycle on the coarse grid. At that point the displacement thickness is computed at the same x coordinates as the inviscid grid and the displacement surface ordinates updated using under-relaxation. The slopes are then determined from cubic splines through the new ordinates, which are updated by a new boundary layer calculation every ten relaxation cycles thereafter. For those cases having extensive trailing edge separation an empirical boundary layer correction is available. However, it is not necessary for most cases.

It should be noted that while the program can include the effects of boundary layer interaction, no correction has been applied for the effects of wake curvature and an empirical approach has been used in the trailing edge region. Thus, the results should be viewed as pressure versus lift coefficient, moment vs. lift, etc. instead of angle of attack. However, the error in angle of attack is believed to be small.

Typical total computation times on an Amdahl 470/V6 or a CYBER 175-T are 60-70 seconds for medium grid results and less than 250 seconds for fine grid solutions.

PROGRAM USAGE

The program is written in FORTRAN IV programming language for use on IBM 360-370, Amdahl 470, CDC 6600, and CDC CYBER series computers. The program can be overlaid in order to reduce computer storage, if required. In nonoverlay mode it requires less than 200,000 bytes on an IBM type machine. Some modification to formats etc. may be required to run the program on different computer systems.

The input cards are summarized in the following table:

Read Order	Variables	Format
1	NTITLE	20A4
2	NAMELIST/FINP/M,W,X1,X2,ALP,EPS,EPSS, X4,S4,CONV,A1,A2,A3,RN,XIBDLY,CIR,CDCORR, RDEL,RDELFN,SP,XSEP,XLSEP,XPC	Namelist
3	NAMELIST/IINP/IMAX,JMAX,IKASE,INV,MITER, NHALF,ITACT,ISKP2,ISKP3,ISKP4,ITERP,IREAD, LP,ITEUPC,ITELWC	Namelist
4	P(I,J) I=1,IMAX; J=1,JMAX (Only if IREAD=1)	5E15.7
5	PB(I) I=1,IMAX (Only if IREAD=1)	5E15.7
6*	X1,X2	2F10.5
7	NI	I5
8	XI(I),YI(I), I=1,NI	8F10.4
9	DERIX,DERIY,DERFX,DERFY	8F10.4
10	NIB	I5
11	XIB(I),YIB(I), I=1, NIB	8F10.4
12	DERIXB,DERIYB,DERFXB,DERFYB	8F10.4
13*	X1,X2	2F10.5
14*	CPU(I), I=I1, ITE	8F10.3
15*	CPL(I), I=I1, ITE	8F10.3
16*	X1,X2	2F10.5
17*	CPU(I), I=I1,ITE	8F10.3
18*	CPL(I), I=I1, ITE	8F10.3
19*	X1,X2	2F10.5
20*	CPU(I), I=I1, ITE	8F10.3

21* CPL(I), I=I1, ITE 8F10.3

* Read only in the design mode when INV=1
NOTE: In the design mode steps 13-15 are for the coarse grid, 16-18 for the medium and 19-21 for the fine grid (if used).

The definitions of these input variables are as follows:

NTITLE - Description of case. Up to 80 alphanumeric characters. Appears on printed output, at the beginning of the results for each grid.

M - Freestream Mach number (real variable). Default 0.5

W - Relaxation factor for subsonic points. Should be in the range $0<W\leq 2.0$ Default 1.7

X1 - X location where direct calculation stops. In analysis mode it should be set to 0.5 (i.e. trailing edge). In design mode it is usually set to slightly less than the third point from the leading edge or larger. Default 0.5

X2 - End of the inverse region. For analysis case set to a large number. In inverse design case set to 0.5 (i.e. trailing edge). Default 10000.0

ALP - Angle of attack in degrees. Default 0.0

EPS - Subsonic damping factor to match difference equations at sonic line if needed. EPS has no effect on accuracy of solution, only on stability and convergence rate. Normally it is not needed. Default 0.0

EPSS - Supersonic damping factor for iterative stability. Note that EPSS has no effect on the accuracy of the converged solution, only on the stability and convergence rate. EPSS should typically be about $M^2_{max}-1$, where M_{max} is the maximum local Mach number. Default 0.4

X4 - The positive X location where the coordinate stretching changes. It should be near the airfoil trailing edge. Default 0.49.

S4 - The positive ξ value in the computational plane where the stretching changes. Default 2.0

CONV - Convergence criteria control value. Iterations stop when the maximum change in the perturbation potential (between relaxation cycles) is less than CONV. Default 1.E-05

A1 - Stretching constant for the Y direction. It can be used to control ΔY and $\Delta\eta$ near the horizontal axis. It is usually best to have $\Delta\xi = \Delta\eta$

near the leading edge of the airfoil. Default 0.246

A2 - First stretching constant for the X-direction. It is equivalent to $\frac{2}{\pi}\left(\frac{dx}{d\xi}\right)$ at $\xi = \xi_4$. The value of A_2 determines the horizontal step size near the leading and trailing edges, i.e.

$$\Delta X_{x=x_4} = \frac{\pi A_2}{2} \Delta\xi = \frac{\pi A_2}{2} \frac{(2(1+S4))}{(IMAX-1)}$$

See Appendix A. Default 0.15

A3 - Second stretching constant for the x-direction. It determines the physical location of the vertical grid line adjacent to grid side edge. Default 3.87.

RN - Freestream Reynolds number based on chord length. Used only when viscous interaction included. Default 20.E+06.

XIBDLY - The x-location at which transition is assumed to occur. The turbulent boundary layer calculation starts at the next grid point. The relationship to percent chord is

XIBDLY = (% chord-50.0)/100.0

Default -0.44.

CIR - Circulation about airfoil. If an initial solution is inputted, it must be the corresponding value of circulation. (CIR = C_L/2.0). Default 0.0

CDCORR - Correction to the computed wave drag coefficient for the finest grid used. Because of the lack of a large number of points in the leading and trailing edge regions, the wave drag coefficient has an error associated with grid size, spacing, and lift coefficient. The magnitude of CDCORR as a function of lift can be determined from a series of calculations at different angles of attack at subcritical speeds, where the wave drag should be zero. Note that the correction should be determined for each airfoil and grid combination. Default 0.0. See Appendix B.

RDEL - Relaxation parameter for the boundary layer displacement thickness. It is used only when viscous interaction is included and IMAX $\lesssim$ 55. Default 0. 25

RDELFN - Fine grid relaxation parameter for the boundary layer displacement thickness. It is used only when viscous interaction is included and

IMAX > 55. Default 0.125

SP - Maximum value allowed for the Nash-Macdonald separation parameter when x < XSEP. Used only in the viscous interaction case. Default 0.004.

XSEP - X location after which the Nash-Macdonald separation parameter can assume its calculated value. Used only in the viscous interaction case. Default 0.44

XLSEP - Location at which the trailing edge correction procedure begins. It should correspond to the point of separation, if used. Between XLSEP and the trailing edge the pressure distribution and the displacement surface is modified. Used only if ITEUPC and/or ITELWC equal 1. Default 0.50

XPC - Location after which the lower surface displacement thickness is required to continue decreasing once it has started to decrease. Upstream of XPC the displacement thickness is required to be monotonically increasing. For most aft-cambered airfoils it should be 0.1 and in conventional airfoils it should be 0.5. Default 0.1

IMAX - Number of vertical grid lines in the horizontal direction. I = 1 is upstream infinity and I = IMAX is downstream infinity. For each grid refinement IMAX is increased such that $IMAX_{new} = 2(IMAX_{old}) - 1$. The limit on IMAX is 99. Default for use on first grid is 13.

JMAX - Number of horizontal grid lines in the vertical direction. J = 1 corresponds to infinity below the airfoil and J = JMAX is infinity above the airfoil. The same formula and limit that apply to IMAX also apply to JMAX. Default 7.

IKASE - An integer number describing the case. It is limited to a maximum of six digits. Default 100.

INV - Parameter determining program mode. It should be zero for analysis cases and one for inverse design cases. Default 0.

MITER - Maximum number of iterations (complete relaxation cycles) allowed on first grid. MITER is halved for each grid refinement. However, on the fourth grid, MITER is reset to 400. Default 800.

NHALF - Number of grid refinements to be done. Default 2.

ITACT - Viscous interaction control parameter. It should be set to zero for analysis cases without interaction and for design cases. It should be one for analysis cases with interaction. Default 0.

ISKP2 - Airfoil update control parameter for grid two. It should be 0 if on grid two an update is desired every 10 iterations. It should be 1 if an update is not desired until the grid two solution is completed. Only used in the inverse design mode. Default 0.

ISKP3 - Same as ISKP2 but for grid 3 (medium grid).

ISKP4 - Same as ISKP2 but for grid 4 (fine grid).

ITERP - Interpolation parameter. If in the design mode the input C_p distribution for the grid 4 is to be read in, ITERP should be 0. If it is desired to linearly interpolate the C_p distribution of grid 3, it should be 1. Default 0.

IREAD - Starting solution control parameter. If IREAD is 0, the initial perturbation solution is assumed to everywhere be zero. If it is 1, an initial solution is read in from data cards. Default 0.

LP - Relaxation cycle interval at which boundary layer, surface ordinates, etc. details are printed. Useful for diagnostics. Default 1000. (No printout.)

ITEUPC - Upper surface trailing edge correction control parameter. If trailing edge correction desired, ITEUPC should be 1. If not it should be zero. Only used in the viscous interaction case. Normally the correction is not needed. Default 0.

ITELWC - Lower surface trailing edge correction control parameter. If correction desired, ITELWC should be 1. If not it should be 0. Only used in the viscous interaction case, and normally the correction is not needed. Default 0.

P(I,J) - Nondimensional perturbation potential, ϕ_{ij}, at point I,J.

PB(I) - Nondimensional perturbation potential at point I on the $y=0^-$ grid line.

X1, X2 - Same definition as above. However, in the inverse design case they must be read in prior to the solution of each grid. On the first grid (step 6 in above table) should use X1=0.5, X2=10000.0. On remaining grids (steps 13,16, and 19), X1 should be the location where the direct calculation stops and X2 should be 0.5.

NI - The number of coordinate pairs used to describe the upper surface of the airfoil. Presently limited to 99.

XI(I) - Input coordinates in the horizontal direction for the airfoil upper surface. The leading edge corresponds to XI=0.0 and the trailing edge is XI=1.0.

YI(I) - Input coordinates in the vertical direction for the airfoil upper surface.

DERIX - DX/DS of the airfoil upper surface at the leading edge (XI=0.0). It usually is 0.0.

DERIY - DY/DS of the airfoil upper surface at the leading edge (XI=0.0). It usually is 1.0.

DERFX - D^3X/DS^3 of the airfoil upper surface at the trailing edge (XI=1.0). It is usually sufficiently accurate to use 0.0.

DERFY - D^3Y/DS^3 of the airfoil upper surface at the trailing edge (XI=1.0). It is usually sufficiently accurate to use 0.0.

NIB - The number of coordinate pairs used to describe the lower surface of the airfoil. Presently limited to 99.

XIB(I) - Input coordinates in the horizontal direction for the airfoil lower surface. The leading edge corresponds to XIB=0.0 and the trailing edge is XIB=1.0.

YIB(I) - Input coordinates in the vertical direction for the airfoil lower surface. Since positive is up, the values of YIB are usually negative.

DERIXB - DX/DS of the airfoil lower surface at the leading edge. It is usually 0.0

DERIYB - DY/DS of the airfoil lower surface at the leading edge. It usually is -1.0

DERFXB - D^3X/DS^3 of the airfoil lower surface at the trailing edge. It is usually sufficiently accurate to use 0.0.

DERFYB - D^3Y/DS^3 of the airfoil lower surface at the trailing edge. It is usually sufficiently accurate to use 0.0.

CPU(I) - Upper surface inverse region C_p values for design case. I1, which is computed internally, is the first grid point after X1. The distribution must be read in for each grid solved inversely (steps 14,17, and 20).

CPL(I) - Lower surface inverse C_p values for design case. They must be read in for each grid solved inversely (steps 15,18 and 21).

The Program Output for each Grid is:

1.) Heading
2.) Case Number
3.) Coordinate System. It is printed as I,X(I) followed by J,Y(J). Also, the Mach Number, angle of attack, and the location where the direct calculation stops is printed (i.e. X1).
4.) Listing of input data in namelists FINP and IINP.
5.) Airfoil ordinates in direct region
    X - horizontal ordinate, where -0.5 is leading edge and 0.5 is trailing edge
    YU - Upper surface ordinate
    YL - Lower surface ordinate
6.) Desired C_p distribution in inverse region. Only printed in the inverse design case.
7.) Iteration history at ten-cycle intervals.
    CIR - circulation
    DPM - maximum ϕ correction (absolute value) in the last relaxation cycle The (I,J) grid location of DPM is also printed.
    NSSP - Number of supersonic points.
    DELST - δ^* at last grid point on upper surface. In interaction cases it is an indicator of convergence of solution.
    DELTAY- Maximum absolute change in inverse region of the computational plane displacement surface ordinates during last surface update.
8.) Boundary layer details etc. - Every LP cycles details of the boundary layer calculation, current surface ordinates in computational coordinates, and surface slopes are printed. Useful if diagnostics needed.
9.) Final Boundary Layer Results (Viscous interaction case only)
    YUORIG - Airfoil upper surface ordinate.
    DU - Upper surface displacement thickness
    SLU - Slope of upper displacement surface
    YLORIG - Airfoil lower surface ordinate
    SLL - Slope of lower displacement surface
    DL - Lower surface displacement thickness

10.) Pressure Distribution on Airfoil

11.) Displacement Surface Ordinates and Slopes - In the inviscid case this will be the same as airfoil ordinates etc.

12.) Mach number chart of the flow field in the computational plane. Numbers printed are the Mach number multiplied by 100. I increases from top to bottom. J increases from left to right.

13.) Wave Drag Coefficient

14.) Plot of Results

U	- Upper surface C_p
L	- Lower surface C_p
T	- Upper displacement surface
B	- Lower displacement surface
CLCIR	- Lift coefficient from circulation
CL	- Lift coefficient from integration of C_p
CD	- CDWAVE + CDF
CMLE	- Moment coefficient about leading edge
CDF	- Skin friction drag coefficient
CMC4	- Moment coefficient about quarter-chord

In addition, in the inverse case the following is printed after the final grid results.

15.) Boundary layer details - Upper surface

16.) Airfoil ordinates of Upper surface

YOLD	- Displacement surface ordinate
YNEW	- Airfoil ordinate
DELSTAR	- Displacement thickness

17.) Boundary layer details - Lower surface

18.) Airfoil ordinates of lower surface.

APPENDIX A
COORDINATE STRETCHING FUNCTIONS

To facilitate the application of the infinity boundary condition, the coordinates are stretched from a physical x-y plane to the computational ξ-η plane. To do this, the x-axis is subdivided into three regions. The first is from $x = -\infty$ to $x = -x_4$. The second is from $x = -x_4$ to $x = x_4$, and the last is from $x = x_4$ to $x = +\infty$.

The stretching is symmetrical about the origin and is given by

$$x = x_4 + A_2 \tan [\frac{\pi}{2}(\xi-\xi_4)] + A_3 \tan [\frac{\pi}{2}(\xi-\xi_4)^3]$$

in the third region and by

$$x = \xi(a+b\xi^2)$$

in the second region. The constants a and b are automatically computed by the program to staisfy the requirements

$$x = x_4 \text{ at } \xi = \xi_4$$

and

$$\frac{dx}{d\xi} = \frac{\pi A_2}{2} \text{ at } \xi = \xi_4$$

The constant A_2 controls the grid spacing in the vicinity of x_4, near the leading and trailing edges of the airfoil, i.e. $\Delta x \approx \frac{\pi A_2}{2} \Delta\xi$ at $x=x_4$.

A_3 determines the physical location of the grid line adjacent to the grid edge.

In the y-direction the stretching relationship is given by

$$y = A_1 \tan (\frac{\pi}{2} \eta)$$

and thus, A_1 controls the grid size near the airfoil via

$$\frac{dy}{d\eta} = \frac{\pi A_1}{2} \sec^2(\frac{\pi}{2} \eta)$$

and

$$\Delta y = \frac{\pi A_1}{2} \Delta\eta \text{ at } \eta = 0$$

Notice that these stretchings map the infinite x,y plane into the finite computational plane

$$-(1 + \xi_4) \leq \xi \leq (1 + \xi_4)$$

$$-1 \leq \eta \leq 1$$

where ξ_4 determines the amount of the computational plane confined to the vicinity of the airfoil. Also

$$\Delta\xi = \frac{2(1+\xi_4)}{IMAX-1}$$

and

$$\Delta\eta = \frac{2.0}{JMAX-1}$$

Finally, it should be noted that JMAX needs to be sufficiently large so that all points on the J=2 and J=JMAX-1 grid lines are subsonic. Otherwise, the rotated difference scheme may attempt to use points outside the computational space.

APPENDIX B

VISCOUS BOUNDARY LAYER AND WAVE DRAG CORRECTION

Experimental evidence indicates that viscous boundary layer effects are very important in transonic flow. For example, an aft-cambered airfoil inviscidly designed to have a lift coefficient of 0.6 may actually develop 25-50% less lift. This loss in lift is due not only to the existence of a boundary layer displacement surface but also to such factors as wake curvature and vertical pressure gradients in the trailing edge region. To prevent such discrepancies, the effect of the viscous boundary layer should be included in both the analysis and design portion of any numerical method.

In the present program, the approach is to assume that the inviscid streamlines follow a displacement surface having ordinates and slopes different from the actual airfoil. The effect of the fact that the streamlines do not follow a displacement surface in the vicinity of the trailing edge and that they are influenced by wake curvature is assumed to be either secondary or capable of being handled empirically. In the design case, the approach is to treat the airfoil determined by the inverse method as the displacement surface and to subtract from it the displacement thickness determined by a boundary layer computation. The result is considered to be the actual airfoil ordinates. For the analysis case, the approach is to calculate a boundary layer displacement surface (i.e. airfoil ordinate plus δ^*) using under-relaxation. The invisicid flowfield is then solved, and the displacement surface is updated every ten iterative cycles.

Obviously, the boundary layer scheme must be reliable, reasonably accurate, and computationally very efficient. After extensive investigation, the Nash-Macdonald method (ref.5) together with certain smoothing operations, was selected for incorporation into the present program. In addition, the displacement thickness at the trailing edge grid point was always determined by linear extrapolation from the previous two upstream grid point values. As a result, the basic appraoch used in the present program is similar to that of reference 6.

To update the displacement surface, the momentum integral equation

$$\frac{d\theta}{ds} + (H + 2-M^2)\,\frac{\theta}{q}\,\frac{dq}{ds} = \tau$$

is solved for the momentum thickness θ using the formulas of Nash and Macdonald for skin friction, τ, and the shape factor $H = \delta^*/\theta$. This computation is performed on the same grid spacing as the corresponding invisicid solution. The resultant displacement thickness is then smoothed everywhere and extrapolated to obtain the thickness at the trailing edge point. The smoothing is performed twice on grids having IMAX less than 55 and four times on grids with IMAX greater than 55.

This smoothing and extrapolation process appears to have two consequences. First, it reduces the rapid variations in the solution which sometimes occur in regions with high pressure gradients. Second, based on comparisons with experiments, the Nash-Macdonald method with smoothing and extrapolation seems to yield a trailing edge behavior that is correct with respect to the effect of the boundary layer on pressure distribution and lift. Admittedly, this behavior is fortuitious and some sensitivity to grid spacing has been detected. However, it should serve as a reasonable engineering model until a more complete, rational, trailing-edge theory is available. At that time such a theory could be easily incorporated into the present program.

If a case with extensive upper surface separation is encountered, the user may need to incorporate the optional trailing edge correction feature of the program. In this correction the boundary layer is solved using a modified pressure distribution that is linear from a point XLSEP, corresponding to separation, to the trailing edge. The base pressure, which determines the pressure gradient in this region, is determined semi-empirically. For aft-cambered airfoils, it is automatically selected to be the same as the maximum value of the pressure coefficient encountered on the lower surface of the airfoil. For conventional airfoils, it should be selected by the user based upon experience. (A typical value is 0.6). For aft-cambered airfoils the resultant modified boundary layer computation is normally applied only to the upper surface and is only used to determine the ordinate and slope of the displacement surface at XLSEP. The slope is then assumed to be constant from XLSEP to the trailing edge and the resultant displacement surface shape determined. Based upon comparisons with experiment, this approach yields reasonably good results and eliminates oscillations in the pressure distribution which can occur due to very small changes in the displacement surface slopes. Thus the method is a combination of the approaches used in references 6 and 7.

For conventional airfoils, the modified boundary layer computation is used to determine when the slope of the displacement surface becomes zero. From that point to the trailing edge the slope is then held constant. This approach is based on the concept that the streamlines from both the upper and lower surfaces should enter the wake almost parallel. Thus, for conventional airfoils, if the trailing edge correction is used, it should be applied to both surfaces. Since in most cases, this correction is not needed on conventional airfoils, its extensive use is not recommended until it has been verified by experiment.

In both the normal and corrected uses, separation is assumed to occur when $(-\theta/q\ dq/ds)$ is greater than 0.004.

Now one of the difficulties associated with using a cartesian grid is that such a grid does not place a large number of computational points near the leading and trailing edges. Thus, the wave drag coefficient, which is determined by integration of the pressure distribution, has an inherent error associated with grid size, grid spacing, and the magnitude of the lift coefficient. Extensive comparisons with experimental data has indicated, however, that accurate estimates of the wave drag can be obtained by applying a suitable correction factor, CDCORR. This correction factor, which is different for each airfoil and computational grid, can be determined as a function lift from a series of calculations at different angles of attack at subcritical speeds, where the wave drag should be zero.

For each subcritical calculation, using a CDCORR of zero, determine the axial and normal coefficients using

$$CN = CL\ \cos\alpha + CDWAVE\ \sin\alpha$$

$$CA = -CL\ \sin\alpha + CDWAVE\ \cos\alpha$$

where CL is the lift coefficient determined from integration of the C_p distribution. Then the true CDCORR corresponding to CL can be computed from

$$CDCORR = CA + CN\ \sin\alpha/\cos\alpha$$

This value can then be used in supercritical runs having the same CL.

In some cases, it may be more convenient to compute the supercritical flows using a CDCORR of zero and to apply the correction later. In those cases, the following procedure can be used to determine the drag. First, compute the axial and normal coefficients using

$$CN = CL \cos \alpha + CDWAVE \sin \alpha$$

$$CA = -CL \sin \alpha + CDWAVE \cos \alpha$$

Then correct the axial coefficient for the appropriate lift and grid by

$$CA = CA - CDCORR$$

and recompute CDWAVE by

$$CDWAVE = CN \sin \alpha + CA \cos \alpha$$

The total drag coefficient is then given by

$$CD = CDWAVE + CDF$$

In all cases the drag due to skin friction and to changes resulting from the displacement surface shape is computed using the Squire-Young formula. While this formula is not exactly theoretically correct, it has been found to yield very accurate predictions.

APPENDIX C

PROGRAM LISTING

```
C  ********** TRANDES--  TRANSONIC ANALYSIS AND DESIGN PROGRAM  *******
C  *********  LELAND A. CARLSON, TEXAS A&M UNIVERSITY, 713-845-7541*****
C **********  JULY 1976*********************************************
C
      REAL M
      DIMENSION NTITLE(20),AA(500),IONIC(99)
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)
     1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
     1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99
     2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
     3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,
     4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1
     5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,
     1JMAX,JCON,JMAX1,NSSP,IW
      COMMON/JS/GG(99),GGP12(99),GGM12(99),GGM32(99),GGP32(99),A3
      COMMON/FIX/MHALF
      COMMON/DELTA/ITER
      COMMON/TAMU/DELTAY
      COMMON/RED/ITERP
      COMMON/FIPUT/IREAD
      COMMON/NASH/RN,IBDLY,ITACT,YUORIG(99),YLORIG(99),SUPPER(99),SLOWER
     1(99),DEL(99),DUPOLD(99),DLWOLD(99),CDF
      COMMON/IPT1/XIBDLY,RDEL,RDELFN,RCPB,SP,XSEP,CONV,CPB,XMON,XLSEP,
     1 MITER,LP,ITEUPC,ITELWC,XPC
      NAMELIST/FINP/M,W,X1,X2,ALP,EPS,EPSS,X4,S4,CONV,A1,A2,A3,RN,
     1 XIBDLY,CIR,CDCORR,RDEL,RDELFN,SP,XSEP,RCPB,CPB,XMON,XLSEP,XPC
      NAMELIST/IINP/IMAX,JMAX,IKASE,INV,MITER,NHALF,ITACT,ISKP2,ISKP3,
     1 ISKP4,ITERP,IREAD,LP,ITEUPC,ITELWC
      DELTAY=0.0
      CDF=0.
      CDCORR=0.0
      DPM=0.0
      ITER=0
      MHALF=1
      IW=0
      CIR=0.0
      DPMSUM=0.0
      IDPM=1
```

```
DPOLD=0.0
M=0.5
W=1.7
X1=0.5
X2=10000.0
ALP=0.0
EPS=0.0
EPSS=0.4
X4=0.49
S4=2.0
A1=0.246
A2=0.15
A3=3.87
RN=20.0E+06
XIBDLY=-.44
CPB=0.4
RDEL=0.25
RDELFN=0.125
SP=0.004
 XSEP=0.44
 XLSEP=0.50
 RCPB=0.2
 XMON=0.47
 CONV=1.E-05
 IMAX=13
 JMAX=7
 IKASE=100
 INV=0
 MITER=800
 NHALF=2
 ITACT=0
 ISKP2=0
 ISKP3=0
 ISKP4=0
 ITERP=0
 IREAD=0
 LP=1000
 ITEUPC=0
 ITELWC=0
 XPC=0.10
```

```
      READ(5,1)(NTITLE(I),I=1,20)
    1 FORMAT(20A4)
      PRINT 2
    2 FORMAT(1H1)
      READ(5,FINP)
      EPSSO=EPSS
      EPSO=EPS
      BETA=SQRT(1.-M*M)
      READ(5,IINP)
      ICASE=IKASE
 8001 CONTINUE
      ALPDEG=ALP
      CPSTAR=1.428/(M*M)*(((1.+0.2*M*M)/1.2)**3.5-1.)
      PI=4.*ATAN(1.0)
      PI2=0.5*PI
      ALP=ALP*PI/180.
      A22=2./(PI*A2)
      A11=2./(PI*A1)
      CALL VALUE
  101 PRINT 3,(NTITLE(I),I=1,20)
    3 FORMAT(20A4)
      CALL COORD
      PRINT 6,M,ALPDEG,X1,ICASE
    6 FORMAT(1H0,3X,'MACH NO. IS ',F5.3,' ANGLE OF ATTACK IS ',F5.3,' DE
     1GREES',/,10X,' DIRECT SOLUTION TO ',F8.2,                        /,
     225X,'CASE NUMBER',I6)
      IF(INV.EQ.0)PRINT 6001
 6001 FORMAT(1H0,3X,'INVISCID ANALYSIS CASE')
      IF(ITACT.EQ.1)PRINT 6002
 6002 FORMAT(1H ,3X,'WITH VISCOUS INTERACTION')
      IF(INV.EQ.1)PRINT 6003
 6003 FORMAT(1H0,3X,'INVERSE DESIGN CASE')
      WRITE(6,FINP)
      WRITE(6,IINP)
      IF(MHALF.EQ.1)GO TO 102
      JB=JMAX/2+1
      DO 104 I=ILE,IMAX
```

```
 104 P(I,JB-1)=0.5*(P(I,JB-2)+PB(I))
     P(IMAX,JMAX-1)=P(IMAX,JMAX-2)
     P(IMAX-1,JMAX)=P(IMAX-2,JMAX)
     P(2,JMAX)=P(3,JMAX)
     P(1,JMAX-1)=P(1,JMAX-2)
     P(1,2)=P(1,3)
     P(2,1)=P(3,1)
     P(IMAX-1,1)=P(IMAX-2,1)
     P(IMAX,2)=P(IMAX,3)
 102 CONTINUE
     ILE1=ILE-1
     I11=I1-1
     ITE=IMAX-ILE1
     ITE1=ITE+1
     CALL FOIL
     IF(IREAD.EQ.1.AND.MHALF.EQ.1)MHALF=MHALF+1
   7 DO 8 J=1,JMAX
   8 P1(J)=P(1,J)
     CALL FLOW1
     CALL FLOW2
     IF(INV.EQ.0)GO TO 9
     IF(MHALF.EQ.1)GO TO 9
     CALL FLOW3
     IF(X2.GT.1000.0)GO TO 10
   9 CALL WAKE
  10 STE=S4+2./PI*ATAN((0.5-X4)/A2)
     CIR=-(STE-S(ITE+1))/DS*(P(ITE,JB)-PB(ITE))+(STE-S(ITE))/DS*
    1(P(ITE+1,JB)-PB(ITE+1))
     QUAN1=-.5*CIR/PI
     QUAN2=ATAN(BETA*A1/A2*DS/DE)
     QUAN3=ATAN(BETA*TAN(ALP))
     IF(M.GT.1.)GO TO 11
     IF(ALP.GT.0.0)GO TO 108
     IF(ALP.LT.0.0)GO TO 1081
     P(IMAX,JMAX)=QUAN1*QUAN2
     DO 12 I=2,IMAX1
     P(I,JMAX)=- CIR/4.0
  12 P(I,1)=-0.75*CIR
     P(1,JMAX)=QUAN1*(PI-QUAN2)
```

```
      P(1,1)=QUAN1*(PI+QUAN2)
      P(IMAX,1)=QUAN1*(2.*PI-QUAN2)
      DO 13 J=2,JMAX1
   13 P(1,J)=-0.5*CIR
      JBM1=JB-1
      DO 14 J=2,JBM1
   14 P(IMAX,J)=-CIR
      GO TO 109
 1081 DO 1083 I=2,IMAX1
      P(I,JMAX)=QUAN1*(PI+ATAN(BETA/TAN(ALP)))
 1083 P(I,1)=QUAN1*PI+P(I,JMAX)
      GO TO 1082
  108 DO 110 I=2,IMAX1
      P(I,JMAX)=QUAN1*ATAN(BETA/TAN(ALP))
  110 P(I,1)=QUAN1*PI+P(I,JMAX)
 1082 CONTINUE
      DO 111 J=2,JMAX1
  111 P(1,J)=QUAN1*(PI-QUAN3)
      JBM1=JB-1
      DO 112 J=2,JBM1
  112 P(IMAX,J)=QUAN1*(2.*PI-QUAN3)
      DO 113 J=JB,JMAX1
  113 P(IMAX,J)=QUAN1*(-QUAN3)
      P(IMAX,JMAX)=QUAN1*QUAN2
      P(1,JMAX)=QUAN1*(PI-QUAN2)
      P(1,1)=QUAN1*(PI+QUAN2)
      P(IMAX,1)=QUAN1*(2.*PI-QUAN2)
  109 CONTINUE
      PB(IMAX)=-CIR+P(IMAX,JB)
   11 ITER=ITER+1
      DPMSUM=DPMSUM+DPM
      IDPM=IDPM+1
      IF(IDPM.LE.10)GO TO 512
      DPOLD=DPMSUM
      DPMSUM=0.0
      IDPM=1
  512 CONTINUE
      IF(ITACT.EQ.1)DELTAY=DUPOLD(ITE)
      IF(ITER/10*10.EQ.ITER)
```

```
     1PRINT 15,ITER,CIR,DPM,ICON,JCON,NSSP,DELTAY
   15 FORMAT(1H ,'ITERATION',I4,' CIR = ',F8.5,' DPM = ',F11.8,' AT',
     12I3,' NSSP = ',I4,' DELTAY OR DELSTAR = ',F7.4)
      IF(M.LE.1.0)GO TO 16
C  ADD P(IMAX,J) CARD HERE FOR M GT 1.0 CASE
   16 IF(INV.EQ.0.AND.ITACT.EQ.0)GO TO 24
      IF(MHALF.EQ.1)GO TO 24
      IF(IREAD.EQ.1.AND.ITACT.EQ.1)GO TO 9005
      IF(ITER.LT.50)GO TO 24
      IF(ITACT.EQ.1)GO TO 9005
      IF(MHALF.EQ.2.AND.ISKP2.EQ.1)GO TO 24
      IF(MHALF.EQ.3.AND.ISKP3.EQ.1)GO TO 24
      IF(MHALF.EQ.4.AND.ISKP4.EQ.1)GO TO 24
      IF(ITER/10*10.EQ.ITER)CALL SHAPE
      GO TO 9006
 9005 IF(ITER/10*10.EQ.ITER)CALL VISACT
 9006 IF(ITER/LP*LP.EQ.ITER)PRINT22,(X(I),YU(I),YL(I),I=ILE,IMAX1)
      IF(ITER/LP*LP.EQ.ITER)PRINT 22,(X(I),SLU(I),SLL(I),I=ILE,IMAX1)
   24 CONTINUE
      IF(ITER.GE.MITER)GO TO 17
      IF(INV.EQ.0.AND.ITACT.EQ.0)GO TO 106
      IF(MHALF.GT.1)GO TO 106
      IF(MHALF.EQ.1.AND.ITER.LT.50)GO TO 106
      DPM=0.
  106 CONTINUE
      IF(DPM.LT.CONV)GO TO 17
   21 DPM=0.0
      GO TO 7
   17 CONTINUE
C     *** THE FOLLOWING CAN BE USED TO PRINT OUT THE *******
C     ******PERTURBATION POTENTIAL FLOWFIELD SOLUTION IF DESIRED*******
C     DO 18 JJ=1,JMAX
C     J=JMAX+1-JJ
C     PRINT 19,J
C  19 FORMAT(1H ,'ROW  ',I5)
C     PRINT 20,(P(I,J),I=1,IMAX)
C  20 FORMAT(1H ,10E11.3)
C  18 CONTINUE
```

```
C      PRINT 19,JB
C      PRINT 20,(PB(I),I=1,IMAX)
       IF(MHALF.LE.NHALF)GO TO 8007
C      ***** THE FOLLOWING CA BE USED TO PUNCH OUTPUT IF DESIRED******
C      DO 8005 JJ=1,JMAX
C8006 FORMAT(5E15.7)
C      J=JMAX-JJ+1
C8005 CONTINUE
 8007 CONTINUE
C    PRINTE SHAPE HERE IF REQUIRED
       IF(MHALF.EQ.1)GO TO 26
       IF(INV.EQ.1)CALL SHAPE
       IF(ITACT.EQ.0)GO TO 26
       IF(ITER.GE.MITER)GO TO 7501
       CALL VISACT
 7501 PRINT 9008,RN
 9008 FORMAT('0','BOUNDARY LAYER ANALYSIS FOR REYNOLDS NUMBER OF',E12.3,
     *//,5X,'X',9X,'YUORIG',4X,'DU',8X,'SLU',7X,'YLORIG',4X,'DL',8X,'SLL
     *')
       PRINT 9009,(X(I),YUORIG(I),DUPOLD(I),SLU(I),YLORIG(I),DLWOLD(I),
     *SLL(I),I=ILE,ITE)
 9009 FORMAT(5X,7F10.5)
       GO TO 9007
   26 CALL PRESS
 9007 DO 25 I=ILE,IMAX1
       YU(I)=A1*TAN(PI/2.*YU(I))
   25 YL(I)=A1*TAN(PI/2.*YL(I))
       PRINT 6004
 6004 FORMAT(1H ,' CP BY CENTRAL DIFFERENCES')
       PRINT 9010
 9010 FORMAT(1H ,'X',10X,'CPU',10X,'CPL')
       PRINT 9011,(X(I),CPU(I),CPL(I),I=ILE,IMAX1)
 9011 FORMAT(1H ,3F10.3)
       IMAX2=IMAX-2
       PRINT 221
  221 FORMAT(1H ,'X',10X,'YU',10X,'YL',10X,'SLU',8X,'SLL')
   22 FORMAT(3(' X= ',F7.4,' YU= ',F7.4,' YL= ',F7.4))
       PRINT 220,(X(I),YU(I),YL(I),SLU(I),SLL(I),I=ILE,ITE)
  220 FORMAT(1H ,5F10.5)
       IF(MHALF.LE.NHALF)GO TO 8014
```

```
8014 CONTINUE
     DO 9002 I=ILE,ITE
     YU(I)=ATAN(YU(I)/A1)/PI2
9002 YL(I)=ATAN(YL(I)/A1)/PI2
     DO 500 I=2,ILE1
     DO 501 J=2,JMAX1
     U=QI*(COS(ALP)+FF(I)*(P(I+1,J)-P(I-1,J))/(2.*DS))
     V=QI*(SIN(ALP)+GG(J)*(P(I,J+1)-P(I,J-1))/(2.*DE))
     UU=U*U
     VV=V*V
     AD=AI2-0.2*(UU+VV-QI2)
 501 IONIC(J)=100.0*SQRT((UU+VV)/AD)
     PRINT 28,(IONIC(J),J=2,JMAX1)
 500 CONTINUE
     DO 502 I=ILE,ITE
     DO 503 J=2,JMAX1
 503 IONIC(J)=0
     JB2=JB-2
     DO 504 J=JB2,JMAX1
     IF(YU(I).GT.E(J).AND.YU(I).LE.E(J+1))GO TO 505
 504 CONTINUE
 505 JA=J+1
     IF(JA.LE.JB)JA=JB+1
     DO 506 J=JA,JMAX1
     U=QI*(COS(ALP)+FF(I)*(P(I+1,J)-P(I-1,J))/(2.*DS))
     V=QI*(SIN(ALP)+GG(J)*(P(I,J+1)-P(I,J-1))/(2.*DE))
     UU=U*U
     VV=V*V
     AD=AI2-0.2*(UU+VV-QI2)
 506 IONIC(J)=100.0*SQRT((UU+VV)/AD)
     JB2=JB+2
     DO 507 JJ=1,JMAX1
     J=JB2-JJ
     IF(YL(I).GE.E(J).AND.YL(I).LT.E(J+1))GO TO 508
 507 CONTINUE
 508 JA=J
     IF(JA.GE.JB)JA=JB-1
     DO 509 J=2,JA
     U=QI*(COS(ALP)+FF(I)*(P(I+1,J)-P(I-1,J))/(2.*DS))
     V=QI*(SIN(ALP)+GG(J)*(P(I,J+1)-P(I,J-1))/(2.*DE))
```

```
      IF(J.EQ.(JB-1))V=QI*(SIN(ALP)+GG(J)*(PB(I)-P(I,J-1))/(2.*DE))
      UU=U*U
      VV=V*V
      AD=AI2-0.2*(UU+VV-QI2)
  509 IONIC(J)=100.0*SQRT((UU+VV)/AD)
      PRINT 28,(IONIC(J),J=2,JMAX1)
  502 CONTINUE
      DO 510 I=ITE1,IMAX1
      DO 511 J=2,JMAX1
      U=QI*(COS(ALP)+FF(I)*(P(I+1,J)-P(I-1,J))/(2.*DS))
      V=QI*(SIN(ALP)+GG(J)*(P(I,J+1)-P(I,J-1))/(2.*DE))
      IF(J.EQ.JB)V=V-QI*(GG(J)*(CIR/(2.*DE)))
      IF(J.EQ.JB-1)V=V-QI*(GG(J)*(CIR/(2.*DE)))
      UU=U*U
      VV=V*V
      AD=AI2-0.2*(UU+VV-QI2)
  511 IONIC(J)=100.0*SQRT((UU+VV)/AD)
      PRINT 28,(IONIC(J),J=2,JMAX1)
  510 CONTINUE
   28 FORMAT(1H ,40I3)
      DO 9003 I=ILE,ITE
      YU(I)=A1*TAN(PI/2.*YU(I))
 9003 YL(I)=A1*TAN(PI/2.*YL(I))
      CL=0.5*(CPL(ILE)-CPU(ILE))*(X(ILE)+0.5)
      CPSTAG=2./(1.4*M*M)*((1.+0.2*M*M)**3.5-1.)
      CD=(CPSTAG+CPU(ILE))*(YU(ILE))*0.5-(CPSTAG+CPL(ILE))*(YL(ILE))*0.5
      CMLE=0.5*(CPU(ILE)-CPL(ILE))*(X(ILE)+0.5)**2
      IEND=ITE-1
      DO 9000 I=ILE,IEND
      T1=CPL(I)-CPU(I)
      T2=CPL(I+1)-CPU(I+1)
      T3=(X(I+1)-X(I))*0.5
      CL=CL+(T1+T2)*T3
      IF(ITACT.EQ.1)GO TO 8010
      CD=CD+(CPU(I)+CPU(I+1))*0.5*(YU(I+1)-YU(I))-(CPL(I)+CPL(I+1))*0.5
     1*(YL(I+1)-YL(I))
```

```
      GO TO 8011
 8010 CD=CD+((CPU(I)+CPU(I+1))*.5*(YUORIG(I+1)-YUORIG(I))-(CPL(I)+CPL(I+1
     *))*.5*(YLORIG(I+1)-YLORIG(I))
 8011 CONTINUE
      T6=-T1*(X(I)+0.5)
      T7=-T2*(X(I+1)+0.5)
 9000 CMLE=CMLE+(T6+T7)*T3
      CL=CL+0.5*(CPL(ITE)-CPU(ITE))*(0.5-X(ITE))
      CMLE=CMLE+0.5*(CPU(ITE)-CPL(ITE))*(X(ITE)+0.5)*(0.5-X(ITE))
      IF(ITACT.EQ.1)GO TO 8012
      CD=CD-(CPU(ITE)+CPL(ITE))*0.5*(YU(ITE)-YL(ITE))
      GO TO 8013
 8012 CD=CD-CDCORR
 8013 CONTINUE
      FN=CL*COS(ALP)-CD*SIN(ALP)
      FT=CL*SIN(ALP)+CD*COS(ALP)
      CL=FN
      CD=FT
      CMC4=CMLE+CL/4.
      CDWAV=CD
      CD=CDWAV+CDF
      PRINT 9012,CDWAV
 9012 FORMAT(1H0,20X,'WAVE CD = ',F10.6)
      NOV=ITE-ILE+1
      DO 114 I=ILE,ITE
      J=I-ILE1
      J1=J+NOV
      J2=J+2*NOV
      J3=J+3*NOV
      J4=J+4*NOV
      AA(J)=X(I)
      AA(J1)=CPU(I)
      AA(J2)=CPL(I)
      AA(J3)=-YU(I)
  114 AA(J4)=-YL(I)
      NL=50
      IF(NOV.GT.45)NL=100
      CALL PLOT(ICASE,AA,NOV,5,NL,0)
      CLCIR=2.*CIR
      PRINT 8002,CPSTAR,CLCIR
```

```
 8002 FORMAT(1H ,40X,'PRESSURE COEFFICIENT',//,41X,'CPSTAR = ',F10.4,
     1 5X,'CLCIR = ',F10.4)
      PRINT 9001,CL,CD,CMLE,CDF,CMC4
 9001 FORMAT(1H0,20X,'CL = ',F10.4,' CD = ',F10.6,' CMLE = ',F10.4,
     *' CDF = ',F10.6,' CMC4 = ',F10.4)
      PRINT 8003
 8003 FORMAT(1H1)
      IF(MHALF.GT.NHALF)GO TO 100
      MHALF=MHALF+1
      MITER=MITER/2
      CALL HALVE
C
      IF(INV.EQ.1.AND.MHALF.EQ.3)MITER=400
      IF(MHALF.EQ.4)MITER=400
C
      DPM=0.0
      DPOLD=0.0
      IDPM=1
      DPMSUM=0.0
      EPSS=EPSSO
      EPS=EPSO
      ITER=0
      DELTAY=0.0
      GO TO 101
  100 CONTINUE
      IF(INV.EQ.1)CALL BDLY
      IF(ITACT.EQ.1)GO TO 9014
      WRITE(7,9015)(X(I),YU(I),YL(I),CPU(I),CPL(I),I=ILE,ITE)
 9015 FORMAT(5F10.5)
      STOP
 9014 WRITE(7,9015)(X(I),YUORIG(I),YLORIG(I),CPU(I),CPL(I),I=ILE,ITE)
      STOP
      END
```

```
      SUBROUTINE FOIL
CC  **********  READS IN INITIAL AIRFOIL SHAPE AND DETERMINES ORDINATES
C **********  AND SLOPES AT COMPUTATIONAL GRID POINTS ************
      REAL M
      DIMENSION XI(99),YI(99),XO(99),YO(99),SI(99),SO(99),XP(99),YP(99),
     1D1Y(99),D2Y(99),D3Y(99),XIB(99),YIB(99)
      DIMENSION XOR(99)
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)
     1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
     1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99
     2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
     3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,
     4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1
     5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,
     1JMAX,JCON,JMAX1,NSSP,IW
      COMMON/FIX/MHALF
      COMMON/RED/ITERP
      COMMON/FIPUT/IREAD
      COMMON/NASH/RN,IBDLY,ITACT,YUORIG(99),YLORIG(99),SUPPER(99),SLOWER
     1(99),DEL(99),DUPOLD(99),DLWOLD(99),CDF
      COMMON/IPT1/XIBDLY,RDEL,RDELFN,RCPB,SP,XSEP,CONV,CPB,XMON,XLSEP,
     1 MITER,LP,ITEUPC,ITELWC,XPC
C     THIS PROGRAM DEPENDS UPON AIRFOIL BEING BEING STUDIED
      PRINT 2
    2 FORMAT(1H0,20X,'AIRFOIL COORDINATES',/,5X,'X       YU          YL
     1   UPPER SLOPE LOWER SLOPE')
      IBDLY=ILE-1
  215 IBDLY=IBDLY+1
      IF(X(IBDLY).LT.XIBDLY)GO TO 215
      IF(ITACT.EQ.1)GO TO 35
      IF(INV.EQ.0)GO TO 7
   35 IF(MHALF.LE.2)GO TO 7
      I=IMAX1/2
      II=IMAX-2
      ISTOR=I11
      IF(ITACT.EQ.1)I11=IBDLY-1
    8 CPU(II)=CPU(I)
      CPL(II)=CPL(I)
      SLU(II)=SLU(I)
```

```
      SLL(II)=SLL(I)
      DUPOLD(II)=DUPOLD(I)
      DLWOLD(II)=DLWOLD(I)
      YU(II)=YU(I)
      YL(II)=YL(I)
      I=I-1
      II=II-2
      IF(II.GE.I11)GO TO 8
      IMAX2=IMAX-2
      IS=II+3
      DO 9 I=IS,IMAX2,2
      DUPOLD(I)=.5*(DUPOLD(I+1)+DUPOLD(I-1))
      DLWOLD(I)=.5*(DLWOLD(I+1)+DLWOLD(I-1))
      CPU(I)=0.5*(CPU(I+1)+CPU(I-1))
      CPL(I)=0.5*(CPL(I+1)+CPL(I-1))
      SLU(I)=0.5*(SLU(I+1)+SLU(I-1))
      SLL(I)=0.5*(SLL(I+1)+SLL(I-1))
      YU(I)=0.5*(YU(I+1)+YU(I-1))
    9 YL(I)=0.5*(YL(I+1)+YL(I-1))
      YU(IMAX1)=0.0001
      YL(IMAX1)=-0.0001
      SLU(IMAX1)=0.0
      SLL(IMAX1)=0.0
      DUPOLD(IMAX1)=0.
      DLWOLD(IMAX1)=0.
      I11=ISTOR
      GO TO 10
    7 CONTINUE
      DO 6 I=ITE1,IMAX
      YUORIG(I)=.0001
      YLORIG(I)=-0.0001
      DUPOLD(I)=0.
      DLWOLD(I)=0.
      YU(I)=0.0001
      YL(I)=-0.0001
      SLU(I)=0.0
    6 SLL(I)=0.0
   10 IF(INV.EQ.1)IEND=I11
      IF(INV.EQ.0)IEND=ITE
      IF(MHALF.LT.3)IEND=ITE
```

```
C       UPPER SURFACE,XI IN PERCENT CHORD
        IF(MHALF.GT.1)GO TO 21
        READ 14, NI
        READ 15,(XI(I),YI(I),I=1,NI)
        READ15,DERIX,DERIY,DERFX,DERFY
     14 FORMAT(I5)
        DO 18 I=1,NI
     18 XI(I)=XI(I)-0.5
     21 DO 16 I=ILE,IEND
        II=I-ILE+1
     15 FORMAT(8F10.4)
C    15 FORMAT(2F10.4)
     16 XO(II)=X(I)
        NO=IEND-ILE+1
        CALL ARC(XI,YI,XO,YO,SI,SO,XP,YP,D1Y,D2Y,D3Y,DERIX,DERFX,DERIY,
       1DERFY,NI,NO,1)
        IF(ITACT.EQ.1)GO TO 23
        DO 17 I=ILE,IEND
        II=I-ILE+1
        DUPOLD(I)=0.
        YU(I)=YO(II)
        YUORIG(I)=YO(II)
     17 SLU(I)=YP(II)/XP(II)
        GO TO 25
     23 DO 24 I=ILE,IEND
        II=I-ILE+1
        IF(I.LT.IBDLY)GO TO 36
        IF(MHALF.GT.2)GO TO 26
     36 YU(I)=YO(II)
        YUORIG(I)=YO(II)
        DUPOLD(I)=0.
        SUPPER(I)=SO(II)
        SLU(I)=YP(II)/XP(II)
        GO TO 24
     26 YUORIG(I)=YO(II)
        SUPPER(I)=SO(II)
     24 CONTINUE
C       LOWER SURFACE, XI IN PERCENT CHORD
     25 IF(MHALF.GT.1)GO TO 22
        READ14,NIB
```

```
      READ15,(XIB(I),YIB(I),I=1,NIB)
      READ15,DERIXB,DERIYB,DERFXB,DERFYB
      DO 19 I=1,NIB
   19 XIB(I)=XIB(I)-0.5
   22 CALL ARC(XIB,YIB,XO,YO,SI,SO,XP,YP,D1Y,D2Y,D3Y,DERIXB,DERFXB,
     1DERIYB,DERFYB,NIB,NO,1)
      IF(ITACT.EQ.1)GO TO 27
      DO 20 I=ILE,IEND
      II=I-ILE+1
      DLWOLD(I)=0.
      YL(I)=YO(II)
      YLORIG(I)=YL(II)
   20 SLL(I)=YP(II)/XP(II)
      GO TO 28
   27 DO 29 I=ILE,IEND
      II=I-ILE+1
      IF(I.LT.IBDLY)GO TO 37
      IF(MHALF.GT.2)GO TO 30
   37 YL(I)=YO(II)
      YLORIG(I)=YO(II)
      DLWOLD(I)=0.
      SLOWER(I)=SO(II)
      SLL(I)=YP(II)/XP(II)
      GO TO 29
   30 YLORIG(I)=YO(II)
      SLOWER(I)=SO(II)
   29 CONTINUE
      IF(IREAD.EQ.0)GO TO 28
      READ 217,(YU(I),YL(I),SLU(I),SLL(I),I=ILE,ITE)
      READ 217,(DUPOLD(I),DLWOLD(I),I=ILE,ITE)
  217 FORMAT(5E15.7)
   28 DO 1 I=ILE,IEND
      PRINT 3, X(I),YU(I),YL(I),SLU(I),SLL(I)
    3 FORMAT(5F10.5)
    1 CONTINUE
C     FINDING COORDS IN ETA-PSI SYSTEM
      DO 4 I=ILE,IMAX1
      YU(I)=ATAN(YU(I)/A1)/PI2
    4 YL(I)=ATAN(YL(I)/A1)/PI2
      IF(INV.EQ.0)RETURN
```

```
      IF(IREAD.EQ.1.AND.MHALF.EQ.1)GO TO 103
      IF(MHALF.EQ.1)RETURN
      IF(MHALF.GT.3.AND.ITERP.EQ.1)GO TO 100
  103 CONTINUE
      READ(5,5)(CPU(I),I=I1 ,ITE)
      CPU(I11)=0.0
      READ(5,5)(CPL(I),I=I1 ,ITE)
      CPL(I11)=0.0
      ITEP1=ITE+1
      DO 339 I=ITEP1,IMAX
      CPU(I)=0.
  339 CPL(I)=0.
    5 FORMAT(8F10.3)
  100 CONTINUE
      PRINT 101
  101 FORMAT(1H0,20X,'UPPER CP INPUT')
      PRINT 5,(CPU(I),I=I1 ,ITE)
      PRINT 102
  102 FORMAT(1H0,20X,'LOWER CP INPUT')
      PRINT 5,(CPL(I),I=I1 ,ITE)
      RETURN
      END
```

```
      SUBROUTINE VISACT
C
C  *********  COMPUTES BOUNDARY LAYER WHEN VISCOUS INTERACTION INCLUDED
C**********   IN THE ANALYSIS CASE     **************
C
      REAL M,NEW
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)
     1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
     1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99
     2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
     3A1,A2,A12,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,
     4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1
     5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,
     1JMAX,JCON,JMAX1,NSSP,IW
      COMMON/NASH/RN,IBDLY,ITACT,YUORIG(99),YLORIG(99),SUPPER(99),SLOWER
     1(99),DEL(99),DUPOLD(99),DLWOLD(99),CDF
      COMMON/DELTA/ITER
      COMMON/IPT1/XIBDLY,RDEL,RDELFN,RCPB,SP,XSEP,CONV,CPB,XMON,XLSEP,
     1 MITER,LP,ITEUPC,ITELWC,XPC
      DIMENSION UE(99),DSS(99),DUDS(99),YUN(99),YLN(99),EM(99)
      DIMENSION XI(99),YI(99),XO(99),YO(99),SI(99),SO(99),XP(99),YP(99),
     1D1Y(99),D2Y(99),D3Y(99),XIB(99),YIB(99)
      DIMENSION HS(99),XOR(99)
      DIMENSION CPUT(99),CPLT(99)
      SEPMK=0
      IF(IMAX.GT.55)RDEL=RDELFN
      ISIDE=0
      ICYCLE=1
      ICYBOT=1
      CALL PRESS
      IF(DPM.LE.CONV)GO TO 5009
      IF(ITER.GE.(MITER-1))GO TO 5009
      GO TO 5005
 5009 DO 5006 J=ILE,IMAX1
      CPUT(J)=CPU(J)
 5006 CPLT(J)=CPL(J)
 5005 CONTINUE
      DO 500 J=ILE,ITE
      YU(J)=A1*TAN(PI/2.*YU(J))
```

```
  500 YL(J)=A1*TAN(PI/2.*YL(J))
      TR=0.3424
      TE1=5.E-03
      TE2=5.E-05
      LMON=IMAX1/2+1
 4001 LMON=LMON+1
      IF(X(LMON).LT.XMON)GO TO 4001
      LSEP=IMAX1/2+1
 4002 LSEP=LSEP+1
      IF(X(LSEP).LT.XLSEP)GO TO 4002
      CM=1.+.2*M**2
      IF(ITER/LP*LP.EQ.ITER)PRINT 1,RN
    1 FORMAT(1H0,10X,'BOUNDARY LAYER ANALYSIS FOR REYNOLDS NO. OF ',E10.
     13,//,5X,'X',9X,                 'M',8X,'DELS',4X,'THETA',3X,'SEP',
     210X,'H',9X,'PI',5X,'TAU')
 1000 ISIDE=ISIDE+1
      SEPMK=0
      DO 2 J=ILE,ITE
      DEL(J)=0.
      IF(ISIDE.EQ.2)GO TO 3
      CP=CPU(J)
    4 TEST=(5.*(CM/(1.+.7*CP*M**2)**(.2857143)-1.))
      EM(J)=0.
      IF(TEST.GT.0.)EM(J)=SQRT(TEST)
      DD=1.+.2*EM(J)**2
      T=CM/DD
      UE(J)=EM(J)/M*SQRT(T)
      GO TO 2
    3 CP=CPL(J)
      GO TO 4
    2 CONTINUE
      IF(ISIDE.EQ.1)USTR=UE(ITE)
      ILEP1=ILE+1
      DO 5 J=ILEP1,ITE
      IF(ISIDE.EQ.2)GO TO 6
      DSS(J)=SUPPER(J)-SUPPER(J-1)
      GO TO 5
    6 DSS(J)=SLOWER(J)-SLOWER(J-1)
    5 DUDS(J)=(UE(J)-UE(J-1))/DSS(J)
      DT=1.
```

```
      SEPR=0.
      HH=0.
      IBDS=ITE-1
      DO 200 J=IBDLY,IBDS
      EMT=(EM(J+1)+EM(J))/2.
      UESA=(UE(J+1)+UE(J))/2.
      VM=1.+.2*EMT**2
      T=CM/VM
      RFT=UESA*(T+TR)*T/(1.+TR)*RN
      IF(J.NE.IBDLY)GO TO 30
      THET1=320./RFT
      THET2=THET1
      GE=6.5
   30 FC=1.+.066*EMT**2-.008*EMT**3
      FR=1.-.134*EMT**2+.027*EMT**3
      IND=0
   40 IND=IND+1
      IF(THET1.LT.1.E-06)THET1=1.E-06
      IF(FR.LT.0.)FR=ABS(FR)
      IF(RFT.LT.0.)RFT=ABS(RFT)
      TAU=    (FC*(2.4711*ALOG(FR*RFT*THET1)+4.75)+1.5*GE+1724./(GE**2+
     1200.)-16.87)
      IF(TAU.LT.0.)TAU=-TAU
      TAU=1./TAU**2
      HB=1./(1.-GE*SQRT(TAU))
      H=(HB+1.)*(1.+.178*EMT**2)-1.
      SEP=-THET1*DUDS(J+1)/UESA
      IF(SEP.LT.SP)GO TO 41
      IF(X(J+1).LT.XSEP)SEP=SP
   41 PII=H*SEP/TAU
      IF(PII.LT.-1.5)PII=-1.5
      IF(PII.GT.1.E4)PII=1.E4
   50 CONTINUE
      G=6.1*SQRT(PII+1.81)-1.7
      T2=ABS((G-GE)/GE)
      GE=G
      DT2=DT
      DT=(H+2.-EMT**2)*SEP+TAU
      IF(IND.GT.1)GO TO 100
      THT=THET2
```

```
      THET1=DT*DSS(J+1)+THT
      THET1=.5*(THET1+THT)
      GO TO 40
  100 DT=(DT2+DT)/2.
      TI=ABS((DT-DT2)/DT)
      IF(TI.LT.TE2)GO TO 120
  110 THET1=DT*DSS(J+1)+THT
      THET1=.5*(THET1+THT)
      IF(IND.LE.500)GO TO 40
      IF(PII.EQ.-1.5)GO TO 130
      GO TO 130
  120 IF(T2.GE.TE1)GO TO 110
  130 THET2=DT*DSS(J+1)+THT
      THET1=.5*(THET2+THT)
      SEP=-THET1*DUDS(J+1)/UESA
      SEPR=(SEPR*DSS(J+1)+SEP*DSS(J))/(DSS(J)+DSS(J+1))
      HH=(HH*DSS(J+1)+H*DSS(J))/(DSS(J)+DSS(J+1))
      DELS=HH*THT
      DEL(J)=DELS
      IF(DEL(J).GT.0.1)DEL(J)=0.1
      HS(J)=HH
      IF(ITER/LP*LP.EQ.ITER)PRINT 10,X(J),EM(J),DELS,THT,SEPR,HH,PII,TAU
   10 FORMAT(9F10.5,I10,F10.5)
  205 CONTINUE
    9 CONTINUE
      IF(J.EQ.IBDS)GO TO 200
      SEPR=SEP
      HH=H
  200 CONTINUE
      SEPR=-SEPR+2.*SEP
      HH=HS(ITE-1)+(DSS(ITE)/DSS(ITE-1))*(HS(ITE-1)-HS(ITE-2))
      HS(ITE)=HH
      DELS=HH*THET2
      DEL(ITE)=DELS
      IF(DEL(ITE).GT.0.1)DEL(ITE)=0.1
      IF(ITER/LP*LP.EQ.ITER)PRINT10,X(ITE),EM(ITE),DELS,THET2,SEPR,
     *HH,PII,TAU
  202 IF(ISIDE.EQ.2)GO TO 203
      EMSTR=EM(ITE)
      HSTR=HH
```

```
      TSTR=THET2
      DO 170 J=ILE,IBDS
  170 IF(DEL(J+1).LT.DEL(J))DEL(J+1)=DEL(J)
  203 CONTINUE
      IF(ISIDE.EQ.1)GO TO 2200
      J=ILE
 2180 J=J+1
      IF(DEL(J+1).LT.DEL(J))GO TO 2185
      IF(J.LT.IBDS)GO TO 2180
      GO TO 2200
 2185 IF(X(J).GT.XPC)GO TO 2190
      DEL(J+1)=DEL(J)
      GO TO 2180
 2190 J=J+1
      IF(J.GT.IBDS)GO TO 2200
      IF(DEL(J+1).GT.DEL(J))DEL(J+1)=DEL(J)
      IF(J.LT.IBDS)GO TO 2190
 2200 CONTINUE
      ISMOTH=2
      IF(IMAX.GT.55)ISMOTH=4
      DO 171  JJ=1,ISMOTH
      OLD=DEL(ILE)
      ILEP2=ILE+2
      DO 171  J=ILEP2,ITE
      NEW=DEL(J-1)
      DEL(J-1)=.25*(OLD+NEW+NEW+DEL(J))
  171 OLD=NEW
      FAC=-DSS(ITE)/DSS(ITE-1)
      DEL(ITE)=FAC*DEL(ITE-2)+(1.-FAC)*DEL(ITE-1)
      DO 172  J=ILEP1,IBDS
      SLOPE=SLU(J)
      IF(ISIDE.EQ.2)SLOPE=SLL(J)
      CO=ABS(ATAN(SLOPE))
      CO=COS(CO)
      IF(ISIDE.EQ.2)GO TO 173
      DY=DUPOLD(J)+RDEL*(DEL(J)-DUPOLD(J))
      YU(J)=YUORIG(J)+DY/CO
      DUPOLD(J)=DY
      GO TO 172
  173 DY=DLWOLD(J)+RDEL*(DEL(J)-DLWOLD(J))
```

```
      YL(J)=YLORIG(J)-DY/CO
      DLWOLD(J)=DY
      GO TO 172
  172 CONTINUE
       SLOPE=SLU(ITE)
      IF(ISIDE.EQ.2)SLOPE=SLL(ITE)
      CO=ABS(ATAN(SLOPE))
      CO=COS(CO)
      IF(ISIDE.EQ.2)GO TO 175
      DY=DUPOLD(ITE)+RDEL*(DEL(ITE)-DUPOLD(ITE))
      YU(ITE)=YUORIG(ITE)+DY/CO
      DUPOLD(ITE)=DY
      GO TO 204
  175 DY=DLWOLD(ITE)+RDEL*(DEL(ITE)-DLWOLD(ITE))
      YL(ITE)=YLORIG(ITE)-DY/CO
      DLWOLD(ITE)=DY
  204 CONTINUE
      IF(ITEUPC.EQ.0)GO TO 5003
C     **INSERT SEPERATED CORRECTION HERE IF DESIRED**
C     ** SEPERATED COORECTION**
      IF(ISIDE.EQ.2)GO TO 5003
      IF(ICYCLE.GT.1)GO TO 300
      LMON=IMAX/2+1
      CPB=0.
      DO 5001 J=LMON,IBDS
      CPN=CPL(J)
      CPB=AMAX1(CPB,CPN)
 5001 CONTINUE
      CPB=0.6
      PRINT 5002,CPB
 5002 FORMAT(' ','BASE PRESSURE COEFFICIENT = ',F10.3)
      IF(LSEP.EQ.ITE)LSEP=ITE-1
      LSEP1=LSEP+1
      SLOP=(CPB-CPU(LSEP))/(.5-X(LSEP))
      DO 501 J=LSEP1,ITE
  501 CPU(J)=SLOP*(X(J)-X(LSEP))+CPU(LSEP)
      ICYCLE=ICYCLE+1
      ISIDE=0
      GO TO 1000
 5003 CONTINUE
```

```
      IF(ITELWC.EQ.0)GO TO 300
      IF(ISIDE.EQ.1)GO TO 300
C     ** LOWER SURFACE CORRECTION, IF DESIRED**
      IF(ICYBOT.GT.1)GO TO 300
      IF(LSEP.EQ.ITE)LSEP=ITE-1
      SLOP=(CPB-CPL(LSEP))/(.5-X(LSEP))
      LSEP1=LSEP+1
      DO 5004 J=LSEP1,ITE
 5004 CPL(J)=SLOP*(X(J)-X(LSEP))+CPL(LSEP)
      ICYBOT=ICYBOT+1
      ISIDE=1
      GO TO 1000
C     ** END SEPERATED REGION CORRECTION **
  300 CONTINUE
      IF(ISIDE.LT.2)GO TO 1000
      XO(1)=-.5
      XI(1)=-.5
      YI(1)=0.
      NI=ITE-ILE+2
      DO 210 I=ILE,ITE
      II=I-ILE+2
      XI(II)=X(I)
      XO(II)=X(I)
  210 YI(II)=YU(I)
      NO=NI
      CALL ARC(XI,YI,XO,YO,SI,SO,XP,YP,D1Y,D2Y,D3Y,0.0,0.0,1.0,0.0,NI,
     1NO,1)
      DO 211 I=ILE,ITE
      II=I-ILE+2
      YI(II)=YL(I)
      IF(I.LT.IBDLY)GO TO 211
      SLU(I)=YP(II)/XP(II)
  211 CONTINUE
      IF(ITEUPC.EQ.0)GO TO 5025
      LSEP1=LSEP+1
      IF(XPC.LT.0.495)GO TO 5029
      DO 5030 J=LSEP1,ITE
      IF(SLU(J).GT.0.0)GO TO 5031
 5030 CONTINUE
      GO TO 5025
```

```
 5031 DO 5032 I=J,ITE
      YU(I)=YU(J-1)
 5032 SLU(I)=0.0
      GO TO 5025
 5029 CONTINUE
      DO 5026 I=LSEP1,ITE
      YU(I)=YU(LSEP)+SLU(LSEP)*(X(I)-X(LSEP))
 5026 SLU(I)=SLU(LSEP)
 5025 CONTINUE
      CALL ARC(XI,YI,XO,YO,SI,SO,XP,YP,D1Y,D2Y,D3Y,0.0,0.0,-1.0,0.0,
     1NI,NO,1)
      DO 212 I=ILE,ITE
      II=I-ILE+2
      YU(I)=ATAN(YU(I)/A1)/PI2
      YL(I)=ATAN(YL(I)/A1)/PI2
      IF(I.LT.IBDLY)GO TO 212
      SLL(I)=YP(II)/XP(II)
  212 CONTINUE
      IF(ITELWC.EQ.0)GO TO 5027
      LSEP1=LSEP+1
      IF(XPC.LT.0.495)GO TO 5036
      DO 5033 J=LSEP1,ITE
      IF(SLL(J).LT.0.0)GO TO 5034
 5033 CONTINUE
      GO TO 5027
 5034 DO 5035 I=J,ITE
      YL(I)=YL(J-1)
 5035 SLL(I)=0.0
      GO TO 5027
 5036 CONTINUE
      DO 5028 I=LSEP1,ITE
      YL(I)=YL(LSEP)+SLL(LSEP)*(X(I)-X(LSEP))
 5028 SLL(I)=SLL(LSEP)
 5027 CONTINUE
      HBT=(HSTR+1.)/(1.+.178*EMSTR**2)-1.
      HBB=(HH+1.)/(1.+.178*EM(ITE)**2)-1.
      CDF=TSTR*(USTR**(2.5+.5*HBT))+THET2*(UE(ITE)**(2.5+.5*HBB))
      CDF=2.*CDF
      IF(DPM.LE.CONV) GO TO 5010
      IF(ITER.GE.(MITER-1))GO TO 5010
```

```
      GO TO 5008
 5010 DO 5007 J=ILE,IMAX1
      CPU(J)=CPUT(J)
 5007 CPL(J)=CPLT(J)
 5008 CONTINUE
      RETURN
      END

      SUBROUTINE BDLY
C
C   ***********  COMPUTES BOUNDARY LAYER IN THE DESIGN CASE  *****
C
      REAL M,NEW
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)
     1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
     1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99
     2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
     3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,
     4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1
     5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,
     1JMAX,JCON,JMAX1,NSSP,IW
      COMMON/NASH/RN,IBDLY,ITACT,YUORIG(99),YLORIG(99),SUPPER(99),SLOWER
     1(99),DEL(99),DUPOLD(99),DLWOLD(99),CDF
      COMMON/DELTA/ITER
      COMMON/IPT1/XIBDLY,RDEL,RDELFN,RCPB,SP,XSEP,CONV,CPB,XMON,XLSEP,
     1 MITER,LP,ITEUPC,ITELWC,XPC
      DIMENSION UE(99),DSS(99),DUDS(99),YUN(99),YLN(99),EM(99)
      DIMENSION HS(99)
      DIMENSION XI(99),YI(99),XO(99),YO(99),SI(99),XP(99),D1Y(99),
     1D2Y(99),D3Y(99),XIB(99),YIB(99),SO(99) ,YP(99)
      ISIDE=0
      LSEP=ITE
      SEPMK=0
      ICYCLE=1
      DO 500 J=ILE,ITE
      YUN(J)=YU(J)
  500 YLN(J)=YL(J)
      TR=0.3424
```

```
      TE1=5.E-03
      TE2=5.E-05
      CM=1.+.2*M**2
    1 FORMAT(1H1,10X,'BOUNDARY LAYER ANALYSIS FOR REYNOLDS NO. OF ',E10.
     13,//,5X,'X',9X,'Y',9X,'YNEW',8X,'M',8X,'DELS',4X,'THETA',3X,'SEP',
     210X,'H',9X,'PI',5X,'TAU')
 1000 ISIDE=ISIDE+1
      PRINT 1,RN
      DO 2 J=ILE,ITE
      DEL(J)=0.
      IF(ISIDE.EQ.2)GO TO 3
      CP=CPU(J)
    4 TEST=(5.*(CM/(1.+.7*CP*M**2)**(.2857143)-1.))
      EM(J)=0.
      IF(TEST.GT.0.)EM(J)=SQRT(TEST)
      DD=1.+.2*EM(J)**2
      T=CM/DD
      UE(J)=EM(J)/M*SQRT(T)
      GO TO 2
    3 CP=CPL(J)
      GO TO 4
    2 CONTINUE
      IF(ISIDE.EQ.1)USTR=UE(ITE)
      ILEP1=ILE+1
      XO(1)=-.5
      XI(1)=-.5
      YI(1)=0.
      NI=ITE-ILE+2
      DO 210 I=ILE,ITE
      II=I-ILE+2
      XI(II)=X(I)
      XO(II)=X(I)
  210 YI(II)=YU(I)
      NO=NI
      CALL ARC(XI,YI,XO,YO,SI,SO,XP,YP,D1Y,D2Y,D3Y,0.0,0.0,1.0,0.0,NI,
     1NO,1)
      DO 211 I=ILE,ITE
      II=I-ILE+2
      YI(II)=YL(I)
      SLU(I)=YP(II)/XP(II)
```

```
211 SUPPER(I)=SO(II)
    CALL ARC(XI,YI,XO,YO,SI,SO,XP,YP,D1Y,D2Y,D3Y,0.0,0.0,-1.0,0.0,
   1NI,NO,1)
    DO 212 I=ILE,ITE
    II=I-ILE+2
    SLL(I)=YP(II)/XP(II)
212 SLOWER(I)=SO(II)
    DO 5 J=ILEP1,ITE
    IF(ISIDE.EQ.2)GO TO 6
    DSS(J)=SUPPER(J)-SUPPER(J-1)
    GO TO 5
  6 DSS(J)=SLOWER(J)-SLOWER(J-1)
  5 DUDS(J)=(UE(J)-UE(J-1))/DSS(J)
    DT=1.
    SEPR=0.
    HH=0.
    IBDS=ITE-1
    DO 200 J=IBDLY,IBDS
    EMT=(EM(J+1)+EM(J))/2.
    UESA=(UE(J+1)+UE(J))/2.
    VM=1.+.2*EMT**2
    T=CM/VM
    RFT=UESA*(T+TR)*T/(1.+TR)*RN
    IF(J.NE.IBDLY)GO TO 30
    THET1=320./RFT
    THET2=THET1
    GE=6.5
 30 FC=1.+.066*EMT**2-.008*EMT**3
    FR=1.-.134*EMT**2+.027*EMT**3
    IND=0
 40 IND=IND+1
    TAU=1./(FC*(2.4711*ALOG(FR*RFT*THET1)+4.75)+1.5*GE+1724./(GE**2+
   1200.)-16.87)**2
    HB=1./(1.-GE*SQRT(TAU))
    H=(HB+1.)*(1.+.178*EMT**2)-1.
    SEP=-THET1*DUDS(J+1)/UESA
    IF(SEP.LT.SP)GO TO 41
    IF(X(J+1).LT.XSEP)SEP=SP
 41 PII=H*SEP/TAU
    IF(PII.LT.-1.5)PII=-1.5
```

```
      IF(PII.GT.1.E4)PII=1.E4
   50 CONTINUE
      G=6.1*SQRT(PII+1.81)-1.7
      T2=ABS((G-GE)/GE)
      GE=G
      DT2=DT
      DT=(H+2.-EMT**2)*SEP+TAU
      IF(IND.GT.1)GO TO 100
      THT=THET2
      THET1=DT*DSS(J+1)+THT
      THET1=.5*(THET1+THT)
      GO TO 40
  100 DT=(DT2+DT)/2.
      TI=ABS((DT-DT2)/DT)
      IF(TI.LT.TE2)GO TO 120
  110 THET1=DT*DSS(J+1)+THT
      THET1=.5*(THET1+THT)
      IF(IND.LE.500)GO TO 40
      IF(PII.EQ.-1.5)GO TO 130
      PRINT 160
  160 FORMAT(' PROBLEMS')
      GO TO 130
  120 IF(T2.GE.TE1)GO TO 110
  130 THET2=DT*DSS(J+1)+THT
      THET1=.5*(THET2+THT)
      SEP=-THET1*DUDS(J+1)/UESA
      SEPR=(SEPR*DSS(J+1)+SEP*DSS(J))/(DSS(J)+DSS(J+1))
      HH=(HH*DSS(J+1)+H*DSS(J))/(DSS(J)+DSS(J+1))
      DELS=HH*THT
      DEL(J)=DELS
      IF(DEL(J).GT.0.1)DEL(J)=0.1
      HS(J)=HH
      IF(ISIDE.EQ.2)GO TO 8
      SLOPE=SLU(J)
      CO=ABS(ATAN(SLOPE))
      CO=COS(CO)
      YUN(J)=YU(J)-DELS/CO
      PRINT 10,X(J),YU(J),YUN(J),EM(J),DELS,THT,SEPR,HH,PII,
     1IND,TAU
   10 FORMAT(9F10.5,I10,F10.5)
```

```
      IF(SEPMK.EQ.1)GO TO 205
      IF(SEPR.GT.SP)LSEP=J
      IF(LSEP.NE.ITE)SEPMK=1
  205 CONTINUE
      GO TO 9
    8 SLOPE=SLL(J)
      CO=ABS(ATAN(SLOPE))
      CO=COS(CO)
      YLN(J)=YL(J)+DELS/CO
      PRINT 10,X(J),YL(J),YLN(J),EM(J),DELS,THT,SEPR,HH,PII,
     1IND,TAU
    9 CONTINUE
      IF(J.EQ.IBDS)GO TO 200
      SEPR=SEP
      HH=H
  200 CONTINUE
      SEPR=-SEPR+2.*SEP
      HH=HS(ITE-1)+(DSS(ITE)/DSS(ITE-1))*(HS(ITE-1)-HS(ITE-2))
      HS(ITE)=HH
      DELS=HH*THET2
      DEL(ITE)=DELS
      IF(DEL(ITE).GT.0.1)DEL(ITE)=0.1
      IF(ISIDE.EQ.2)GO TO 201
      SLOPE=SLU(ITE)
      CO=ABS(ATAN(SLOPE))
      CO=COS(CO)
      YUN(ITE)=YU(ITE)-DELS/CO
      PRINT10,X(ITE),YU(ITE),YUN(ITE),EM(ITE),DELS,THET2,SEPR,HH,PII,
     1IND,TAU
      GO TO 202
  201 SLOPE=SLL(ITE)
      CO=ABS(ATAN(SLOPE))
      CO=COS(CO)
      YLN(ITE)=YL(ITE)+DELS/CO
      PRINT 10,X(ITE),YL(ITE),YLN(ITE),EM(ITE),DELS,THET2,SEPR,HH,PII,
     1IND,TAU
  202 IF(ISIDE.EQ.2)GO TO 203
      EMSTR=EM(ITE)
      HSTR=HH
      TSTR=THET2
```

```
      DO 170 J=ILE,IBDS
  170 IF(DEL(J+1).LT.DEL(J))DEL(J+1)=DEL(J)
  203 CONTINUE
      IF(ISIDE.EQ.1)GO TO 2200
      J=ILE
 2180 J=J+1
      IF(DEL(J+1).LT.DEL(J))GO TO 2185
      IF(J.LT.IBDS)GO TO 2180
      GO TO 2200
 2185 IF(X(J).GT.XPC)GO TO 2190
      DEL(J+1)=DEL(J)
      GO TO 2180
 2190 J=J+1
      IF(DEL(J+1).GT.DEL(J))DEL(J+1)=DEL(J)
      IF(J.LT.IBDS)GO TO 2190
 2200 CONTINUE
      ISMOTH=2
      IF(IMAX.GT.55)ISMOTH=4
      DO 171  JJ=1,ISMOTH
      OLD=DEL(ILE)
      ILEP2=ILE+2
      DO 171  J=ILEP2,ITE
      NEW=DEL(J-1)
      DEL(J-1)=.25*(OLD+NEW+NEW+DEL(J))
  171 OLD=NEW
      FAC=-DSS(ITE)/DSS(ITE-1)
      DEL(ITE)=FAC*DEL(ITE-2)+(1.-FAC)*DEL(ITE-1)
      PRINT 180
  180 FORMAT('  X           YOLD        YNEW        DELSTAR')
      DO 172 J=ILEP1,IBDS
      SLOPE=SLU(J)
      IF(ISIDE.EQ.2)SLOPE=SLL(J)
      CO=ABS(ATAN(SLOPE))
      CO=COS(CO)
      IF(ISIDE.EQ.2)GO TO 173
      YUN(J)=YU(J)-DEL(J)/CO
      PRINT 174,X(J),YU(J),YUN(J),DEL(J)
      GO TO 172
  173 YLN(J)=YL(J)+DEL(J)/CO
      PRINT 174,X(J),YL(J),YLN(J),DEL(J)
```

```
 174 FORMAT(4F10.5)
 172 CONTINUE
      SLOPE=SLU(ITE)
     IF(ISIDE.EQ.2)SLOPE=SLL(ITE)
     CO=ABS(ATAN(SLOPE))
     CO=COS(CO)
     IF(ISIDE.EQ.2)GO TO 175
     YUN(ITE)=YU(ITE)-DEL(ITE)/CO
     PRINT 174,X(ITE),YU(ITE),YUN(ITE),DEL(ITE)
     GO TO 204
 175 YLN(ITE)=YL(ITE)+DEL(ITE)/CO
     PRINT 174,X(ITE),YL(ITE),YLN(ITE),DEL(ITE)
 204 IF(ISIDE.LT.2)GO TO 1000
     HBT=(HSTR+1.)/(1.+.178*EMSTR**2)-1.
     HBB=(HH+1.)/(1.+.178*EM(ITE)**2)-1.
     CDF=TSTR*(USTR**(2.5+.5*HBT))+THET2*(UE(ITE)**(2.5+.5*HBB))
     CDF=2.*CDF
     PRINT 3010,CDF
3010 FORMAT(1H0,' CDF = ',F10.6)
     IF(X(LSEP).GE.XSEP)GO TO 3011
     PRINT 3012,X(LSEP)
3012 FORMAT(1H ,'UPPER SURFACE SEPARATION DETECTED AT ',F10.5)
     RETURN
3011 PRINT 3013,XSEP
3013 FORMAT(1H ,'NO UPPER SURFACE SEPARATION BEFORE',F10.5)
     RETURN
     END
```

```
      SUBROUTINE VALUE
C
C   ********   THIS DETERMINES THE INITIAL SOLUTION *****
C
      REAL M
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)
     1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
     1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99
     2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
     3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,
     4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1
     5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
      CCMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,
     1JMAX,JCON,JMAX1,NSSP,IW
      CCMMON/FIPUT/IREAD
C      INITIALIZES VALUES
      IF(IREAD.EQ.0)GO TO 3
      DO 4 JJ=1,JMAX
      J=JMAX-JJ+1
      READ 5,(P(I,J),I=1,IMAX)
    5 FORMAT(5E15.7)
    4 CCNTINUE
      READ5,(PB(I),I=1,IMAX)
      RETURN
    3 CONTINUE
      DO 1 I=1,IMAX
      PB(I)=0.0
      DO 1 J=1,JMAX
    1 P(I,J)=0.0
      RETURN
      END
```

```
      SUBROUTINE SOLVE(JL,JU)
C
C  *********  THIS SUBROUTINE SETS UP THE COEFFICIENTS USED IN THE *****
C                RELAXATION SOLUTION    **********
C
      REAL M,ML
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)
     1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
     1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99
     2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
     3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,
     4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1
     5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,
     1JMAX,JOON,JMAX1,NSSP,IW
      COMMON/ADAM/VJB,VJB1,VJBP1
      COMMON/JS/GG(99),GGP12(99),GGM12(99),GGM32(99),GGP32(99),A3
      DST2=2.*DS
      DET2=2.*DE
      DEDE=DE*DE
      DSDS=DS*DS
      DSDE=DS*DE
      FDS=F/DS
      DO 3 J=JL,JU
      IF(I.NE.ILE1)GO TO 300
      IF(J.NE.JB-1)GO TO 300
      HLD=P(ILE,JB)
      P(ILE,JB)=PB(ILE)
  300 CONTINUE
      G=GG(J)
      GP12=GGP12(J)
      GM12=GGM12(J)
      GDE=G/DE
      U=QI*(COS(ALP)+F*(P(I+1,J)-P1(J))/(2.*DS))
      V=QI*(SIN(ALP)+G*(P(I,J+1)-P(I,J-1))/(2.*DE))
      IF(IW.EQ.0)GO TO 6
      IF(J.EQ.JB)V=V-QI*(G*(CIR/(2.*DE)))
      IF(J.EQ.JB-1)V=V-QI*(G*(CIR/(2.*DE)))
    6 CONTINUE
      UU=U*U
```

```
      VV=V*V
      Q=SQRT(UU+VV)
      QQ=Q*Q
      AA=AI2-0.2*(UU+VV-QI2)
      IF(AA.LT.0.0)PRINT 11,I,J
   11 FORMAT(' AA.LT.0.0    I=',I5,' J = ',I5)
      IF(AA.LT.0.0)GO TO 21
      IF(IW.EQ.0)GO TO 7
      IF(J.NE.JB.AND.J.NE.JB-1)GO TO 9
      IF(J.EQ.JB-1)GO TO 8
      VVJB=VV
      VJB=V
      AAJB=AA
      QQJB=QQ
      UUJB=UU
      GO TO 7
    8 VVJB1=VV
      VJB1=V
      AAJB1=AA
      QQJB1=QQ
      UUJB1=UU
      GO TO 7
   10 VVJBP1=VV
      VJBP1=V
      QQJBP1=QQ
      AAJBP1=AA
      GO TO 7
    9 IF(J.EQ.JB+1)GO TO 10
    7 CONTINUE
      ML=(UU+VV)/AA
      IF(ML.GT.1.0)GO TO 4
C     SUBSONIC POINT
      FAV=1.0*(FP12+FM12)
      FPF=0.5*F*FAV
      SUB(J-1)=(1.-VV/AA)*GDE*GM12/DE
      D(J)=-2.*(1.-UU/AA)*FPF/DSDS/W-(1.-VV/AA)*G*(GP12+GM12)/DEDE
     1-EPS*FDS*U/Q*FM12/DS
      SUP(J)=(1.-VV/AA)*GDE*GP12/DE
      RS(J)=(1.-UU/AA)*F*(-FP12/DSDS*P(I+1,J)+FAV/DSDS*(1.-1./W)
     1*P(I,J)-FM12/DSDS*P(I-1,J))+U*V/AA*FDS*GDE*0.5*(P(I-1,J-1)
```

```
     2-P(I-1,J+1)-P(I+1,J-1)+P(I+1,J+1))
      RS(J)=RS(J)+EPS*FDS*(U/Q*FM12*(-P(I,J)-P(I-1,J)+P1(J))/DS)
      IF(V.LE.0.0)GO TO 200
      SUB(J-1)=SUB(J-1)+EPS*FDS*V/Q*GM12/DE
      D(J)=D(J)-EPS*FDS*V/Q*GM12/DE
      RS(J)=RS(J)+EPS*FDS*V/Q*GM12/DE*(P(I,J-1)-P(I,J))
      GO TO 3
  200 SUP(J)=SUP(J)-EPS*FDS*V/Q*GP12/DE
      D(J)=D(J)+EPS*FDS*V/Q*GP12/DE
      RS(J)=RS(J)+EPS*FDS*V/Q*GP12/DE*(P(I,J)-P(I,J+1))
      GO TO 3
C     DAMPING COEFF IS EPSS AT SUPERSONIC POINTS
C     SUPERSONIC CASE, V GT 0
    4 GM32=GGM32(J)
      NSSP=NSSP+1
      GP32=GGP32(J)
      IF(V.LT.0.0)GO TO 5
      SUB(J-1)=UU/QQ*GDE*GM12/DE+EPSS*FDS*V/Q*GM12/DE
      D(J)=-VV/QQ*FDS*FM12/DS-UU/QQ*GDE*(GP12+GM12)/DE-EPSS*FDS*
     1(U/Q*FM12/DS+V/Q*GM12/DE)
      SUP(J)=UU/QQ*GDE*GP12/DE
      RS(J)=-(1.-QQ/AA)*(UU/QQ*F DS*(FM12*(P(I,J)-P1(J))-FM32*(P1(J)-
     1P2(J)))/DS+2.*U*V/QQ*F*G*(P(I,J)-P1(J)-P(I,J-1)+P1(J-1))/DSDE
     2+VV/QQ*G*(GM12*(P(I,J)-P(I,J-1))-GM32*(P(I,J-1)-P(I,J-2)))/
     3DEDE)
      RS(J)=RS(J)-VV/QQ*F*(FP12*(P(I+1,J)-P(I,J))+FM12*P(I-1,J))/DSDS
     1+U*V/QQ*F*G*(P(I-1,J-1)-P(I-1,J+1)-P(I+1,J-1)+P(I+1,J+1))/DSDE
     2*0.5
     3+EPSS*F*(U/Q*FM12*(-P(I,J)-P(I-1,J)+P1(J))/DSDS+V/Q*GM12*(P(I,J-1)
     4-P(I,J))/DSDE)
      GO TO 3
C     SUPERSONIC CASE V  LT  0
    5 SUB(J-1)=UU/QQ*G DE*GM12/DE
      D(J)=-VV/QQ*F DS*FM12/DS-UU/QQ*G DE*(GM12+GP12)/DE-EPSS*F DS*
     1(U/Q*FM12/DS-V/Q*GP12/DE)
      SUP(J)=UU/QQ*GDE*GP12/DE-EPSS*FDS*V/Q*GP12/DE
      RS(J)=-(1.-QQ/AA)*(UU/QQ*F DS*(FM12*(P(I,J)-P1(J))-FM32*(P1(J)-
     1P2(J)))/DS+2.*U*V/QQ*F*G*(-P(I,J)+P1(J)+P(I,J+1)-P1(J+1))/DSDE
     2+VV/QQ*G*(GP12*(P(I,J)-P(I,J+1))-GP32*(P(I,J+1)-P(I,J+2)))/
     3DEDE)
```

```
      RS(J)=RS(J)-VV/QQ*F*(FP12*(P(I+1,J)-P(I,J))+FM12*P(I-1,J))/DSDS
     1+U*V/QQ*F*G*(P(I-1,J-1)-P(I-1,J+1)-P(I+1,J-1)+P(I+1,J+1))/DSDE
     1*0.5
     2+EPSS*F*(U/Q*FM12*(-P(I,J)-P(I-1,J)+P1(J))/DSDS+V/Q*GP12*(P(I,J)-
     3P(I,J+1))/DSDE)
      IF(I.NE.ILE1)GO TO 3
      IF(J.NE.JB-1)GO TO 3
      P(ILE,JB)=HLD
    3 CONTINUE
      RETURN
   21 DO 18 JJ=1,JMAX
      J=JMAX+1-JJ
      PRINT 19,J
   19 FORMAT(1H ,'ROW ',I5)
      PRINT 20,(P(I,J),I=1,IMAX)
   20 FORMAT(1H ,10E11.3)
   18 CONTINUE
      PRINT 19,JB
      PRINT 20,(PB(I),I=1,IMAX)
      STOP
      END
```

```
      SUBROUTINE PRESS
C
C  ********     THIS COMPUTES THE CP DISTRIBUTION ON THE AIRFOIL *******
C
      REAL M
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)
     1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
     1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99
     2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
     3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,
     4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1
     5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,
     1JMAX,JCON,JMAX1,NSSP,IW
      COMMON/FIX/MHALF
```

```
      COMMON/NASH/RN,IBDLY,ITACT,YUORIG(99),YLORIG(99),SUPPER(99),SLOWER
     1(99),DEL(99),DUPOLD(99),DLWOLD(99),CDF
C     COMPUTES CP ON AIRFOIL
      IPRM=0
      IF(ITACT.EQ.1)IPRM=1
   19 IPRM=IPRM+1
      IEND=IMAX1
      JB2=JB-2
      DO 1 I=ILE,IEND
      TEMP2=YU(I)
      IF(I.GT.ITE)YU(I)=0.0001
      DO 2 J=JB2,JMAX1
      IF(YU(I).GT.E(J).AND.YU(I).LE.E(J+1))GO TO 3
    2 CONTINUE
    3 JA=J+1
      IF(JA.LE.JB)JA=JB+1
      F=FF(I)
      IF(IPRM.EQ.2)GO TO 20
      IF(I.GT.(ILE+1))GO TO 15
   20 U=0.0
      U=QI*(COS(ALP)+F*((P(I+1,JA-1)-P(I-1,JA-1))/(2.*DS)+(YU(I)-E(JA-1)
     1)*(P(I+1,JA)-P(I+1,JA-1)-P(I-1,JA)+P(I-1,JA-1))/(2.*DS*DE)))
      GO TO 16
   15 CONTINUE
C     U USING BACKWARD DIFFERENCE ON PHIX
      U=QI*(COS(ALP)+F*((3.*P(I,JA-1)-4.*P(I-1,JA-1)+P(I-2,JA-1))/(2.*DS
     1)+(YU(I)-E(JA-1)
     1)*(P(I+1,JA)-P(I+1,JA-1)-P(I-1,JA)+P(I-1,JA-1))/(2.*DS*DE)))
   16 CONTINUE
      UU=U*U
      GB=A11/(1.+TAN(PI2*YU(I))**2)
  100 V=QI*(SIN(ALP)+GB*((-3.*P(I,JA-1)+4.*P(I,JA)-P(I,JA+1))/(2.*DE)
     1+(YU(I)-E(JA-1))*(P(I,JA-1)-2.*P(I,JA)+P(I,JA+1))/(DE**2)))
  101 VV=V*V
      YU(I)=TEMP2
    1 CPU(I)=(1./(0.7*M*M))*((1.+0.2*M*M*(1.-(UU+VV)/QI2))**3.5-1.)
      ITE1=ITE+1
      DO 4 I=ILE1,IMAX
      TEMP1=PB(I)
      PB(I)=P(I,JB)
```

```
    4 P(I,JB)=TEMP1
      JB2=JB+2
      DO 5 I=ILE,IEND
      TEMP2=YL(I)
      IF(I.GT.ITE)YL(I)=-0.0001
      DO 6 JJ=1,JB
      J=JB2-JJ
      IF(YL(I).GE.E(J).AND.YL(I).LT.E(J+1))GO TO 7
    6 CONTINUE
    7 JA=J
      IF(JA.GE.JB)JA=JB-1
      F=FF(I)
      IF(IPRM.EQ.2)GO TO 21
      IF(I.GT.(ILE+1))GO TO 17
   21 U=0.0
      U=QI*(COS(ALP)+F*((P(I+1,JA+1)-P(I-1,JA+1))/(2.*DS)+(YL(I)-E(JA+1)
     1)*(P(I+1,JA+1)-P(I+1,JA)-P(I-1,JA+1)+P(I-1,JA))/(2.*DS*DE)))
      GO TO 18
   17 CONTINUE
C     U USING BACKWARDS DIFFERNCE SCHEME ON PHIX
      U=QI*(COS(ALP)+F*((3.*P(I,JA+1)-4.*P(I-1,JA+1)+P(I-2,JA+1))/(2.*DS
     1)+(YL(I)-E(JA+1)
     1)*(P(I+1,JA+1)-P(I+1,JA)-P(I-1,JA+1)+P(I-1,JA))/(2.*DS*DE)))
   18 CONTINUE
      UU=U*U
      GB=A11/(1.+TAN(YL(I)*PI2)**2)
  102 V=QI*(SIN(ALP)+GB*((3.*P(I,JA+1)-4.*P(I,JA)+P(I,JA-1))/(2.*DE)+
     1(YL(I)-E(JA+1))*(P(I,JA+1)-2.*P(I,JA)+P(I,JA-1))/(DE**2)))
  103 VV=V*V
      YL(I)=TEMP2
    5 CPL(I)=(1./(0.7*M*M))*((1.+0.2*M*M*(1.-(UU+VV)/QI2))**3.5-1.)
      DO 8 I=ILE1,IMAX
      TEMP1=PB(I)
      PB(I)=P(I,JB)
    8 P(I,JB)=TEMP1
      IF(ITACT.EQ.1)RETURN
   11 IF(IPRM.EQ.1)PRINT 200
  200 FORMAT(1H ,'CP BY BACKWARD DIFFERENCES')
      IF(IPRM.EQ.2)PRINT 201
  201 FORMAT(1H ,'CP BY CENTRAL DIFFERENCES')
```

```
      PRINT 12
   12 FORMAT(1H ,'X',10X,'CPU',10X,'CPL')
      PRINT 13,(X(I),CPU(I),CPL(I),I=ILE,IMAX1)
   13 FORMAT(1H ,3F10.3)
      IF(IPRM.LT.2)GO TO 19
      RETURN
      END


      SUBROUTINE FLOW1
C
C  *********   SOLVES FLOW IN FRONT OF THE AIRFOIL   *****
C
      REAL M
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)
     1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
     1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99
     2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
     3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,
     4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1
     5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,
     1JMAX,JCON,JMAX1,NSSP,IW
C     RELAXES FLOW IN FRONT OF AIRFOIL
      NSSP=0
      ISTAR=2
      IF(M.LT.1.0)GO TO 1
      ISTAR=3
      DO 2 J=1,JMAX
      P1(J)=P(2,J)
    2 P2(J)=P(1,J)
    1 CONTINUE
      AI=1117.0
      AI2=AI**2
      QI=M*AI
      QI2=QI**2
      DO 1000 I=ISTAR,ILE1
      F=FF(I)
      FP12=FFP12(I)
```

```
      FM12=FFM12(I)
      FM32=FFM32(I)
      CALL SOLVE(2,JMAX1)
      RS(2)=RS(2)-SUB(1)*P(I,1)
      RS(JMAX1)=RS(JMAX1)-SUP(JMAX1)*P(I,JMAX)
      CALL TRID(2,JMAX1)
      DO 6 J=2,JMAX1
      DP=ABS(P(I,J)-RS(J))
      IF(DP.GT.DPM)ICON=I
      IF(DP.GT.DPM)JCON=J
      IF(DP.GT.DPM)DPM=DP
      P2(J)=P1(J)
      P1(J)=P(I,J)
    6 P(I,J)=RS(J)
      P2(1)=P1(1)
      P2(JMAX)=P1(JMAX)
      P1(1)=P(I,1)
      P1(JMAX)=P(I,JMAX)
 1000 CONTINUE
      RETURN
      END


      SUBROUTINE COORD
C
C  ********     SETS UP COORDINATES IN COMPUTATIONAL AND PHYSICAL GRIDS *
C
      REAL M
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)
     1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
     1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99
     2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
     3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,
     4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1
     5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,
     1JMAX,JCON,JMAX1,NSSP,IW
      COMMON/FIX/MHALF
      COMMON/JS/GG(99),GGP12(99),GGM12(99),GGM32(99),GGP32(99),A3
```

```
      DE=2.0/(JMAX-1)
      IF(INV.EQ.0)GO TO 999
      READ 997,X1,X2
  997 FORMAT(2F10.5)
  999 CONTINUE
      DS=2.*(1.+S4)/(IMAX-1)
C     THIS PROGRAM DEPENDS UPON TRANSFORMATION USED
      S(1)=-1.0-S4
      E(1)=-1.0
      S(IMAX)=1.0+S4
      E(JMAX)=1.0
      IMAX1=IMAX-1
      JMAX1=JMAX-1
      DO 2 I=2,IMAX1
    2 S(I)=S(I-1)+DS
      S3=-S4+0.5*DS-0.01
      DO 11 I=1,IMAX1
      IF(S3.GE.S(I).AND.S3.LT.S(I+1))GO TO 12
   11 CONTINUE
   12 I3=I+1
      I31=I3-1
      IM=IMAX/2+1
      I4=IMAX-I31
      I41=I4+1
      DO 13 I=2,I31
   13 X(I)=-X4+A2*TAN(PI2*(S(I)+S4))+A3*TAN(PI2*(S(I)+S4)**3)
      TERM1=1.5*X4/S4-.25*PI*A2
      TERM2=(.5*PI*A2*S4-X4)/(2.*S4**3)
      DO 14 I=I3,I4
   14 X(I)=S(I)*(TERM1+TERM2*S(I)**2)
      DO 16 I=I41,IMAX1
   16 X(I)=X4+A2*TAN(PI2*(S(I)-S4))+A3*TAN(PI2*(S(I)-S4)**3)
      DO 3 J=2,JMAX1
      E(J)=E(J-1)+DE
    3 Y(J)=A1*TAN(PI2*E(J))
      PRINT 4
    4 FORMAT(//,25X,'X-Y GRID SYSTEM',//)
      PRINT 5,(I,X(I),I=2,IMAX1)
    5 FORMAT(    6(I5,E12.4))
      PRINT 5,(J,Y(J),J=2,JMAX1)
```

```
      DO 7 I=2,IMAX1
      IF(X1.GE.X(I).AND.X1.LT.X(I+1))GO TO 8
    7 CONTINUE
    8 I1=I+1
      SLE=-0.5
      DO 9 I=2,IMAX1
      IF(SLE.GE.X(I).AND.SLE.LE.X(I+1))GO TO 10
    9 CONTINUE
   10 ILE=I+1
      QUAN1=S(2)+S4
      FF(2)=    PI2*A2*(1.+TAN(PI2*QUAN1)**2)+1.5*A3*(1.+TAN(PI2*QUAN1**3
     1)**2)*(QUAN1**2)*PI
      FF(2)=1./FF(2)
      QUAN1=S(2)+0.5*DS+S4
      FFP12(2)=PI2*A2*(1.+TAN(PI2*QUAN1)**2)+1.5*A3*(1.+TAN(PI2*QUAN1**3
     1)**2)*(QUAN1**2)*PI
      FFP12(2)=1./FFP12(2)
      QUAN1=S(2)-0.5*DS+S4
      FFM12(2)=PI2*A2*(1.+TAN(PI2*QUAN1)**2)+1.5*A3*(1.+TAN(PI2*QUAN1**3
     1)**2)*(QUAN1**2)*PI
      FFM12(2)=1./FFM12(2)
      FFM1(2)=0.0
      DO 18 I=3,I31
      QUAN1=S(I)+S4
      FF(I)=    PI2*A2*(1.+TAN(PI2*QUAN1)**2)+1.5*A3*(1.+TAN(PI2*QUAN1**3
     1)**2)*(QUAN1**2)*PI
      FF(I)=1./FF(I)
      QUAN1=S(I)+0.5*DS+S4
      FFP12(I)=PI2*A2*(1.+TAN(PI2*QUAN1)**2)+1.5*A3*(1.+TAN(PI2*QUAN1**3
     1)**2)*(QUAN1**2)*PI
      FFP12(I)=1./FFP12(I)
      FFM12(I)=FFP12(I-1)
      FFM1(I)=FF(I-1)
   18 FFM32(I)=FFM12(I-1)
      FFP12(I31)=1./(TERM1+3.*TERM2*(S(I31)+0.5*DS)**2)
      IM1=IM-1
      DO 19 I=I3,IM
      FF(I)=1./(TERM1+3.*TERM2*S(I)**2)
      FFP12(I)=1./(TERM1+3.*TERM2*(S(I)+0.5*DS) **2)
      FFM12(I)=FFP12(I-1)
```

```
      FFM1(I)=FF(I-1)
   19 FFM32(I)=FFM12(I-1)
      FFP12(IM)=FFM12(IM)
      DO 800 II=2,IM1
      I=IMAX-II+1
      FF(I)=FF(II)
      FFP12(I)=FFM12(II)
      FFM12(I)=FFP12(II)
      FFM1(I)=FF(II+1)
  800 FFM32(I)=FFP12(II+1)
      GGP32(2)=A11*COS(PI2*(E(3)+0.5*DE))**2
      GG(2)=A11*COS(PI2*E(2))**2
      GGP12(2)=A11*COS(PI2*(E(2)+0.5*DE))**2
      GGM12(2)=A11*COS(PI2*(E(2)-0.5*DE))**2
      GGM32(2)=0.0
      DO 801 J=3,JMAX1
      IF(J.EQ.JMAX1)GO TO 804
      GGP32(J)=A11*COS(PI2*(E(J)+1.5*DE))**2
  804 GG(J)=A11*COS(PI2*E(J))**2
      GGP12(J)=A11*COS(PI2*(E(J)+0.5*DE))**2
      GGM12(J)=GGP12(J-1)
  801 GGM32(J)=GGM12(J-1)
      GGP32(JMAX1)=0.0
      RETURN
      END


      SUBROUTINE FLOW3
C
C     *********   SOLVES FOW IN THE INVERSE REGION   *****
C
      REAL M
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)
     1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
     1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99
     2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
     3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,
     4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1
     5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
```

```
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,
     1JMAX,JCON,JMAX1,NSSP,IW
      COMMON/BAKER/TEMP3,TEMP4
      COMMON/DELTA/ITER
      COMMON/FIX/MHALF
C     RELAXES FLOW IN INVERSE REGION
      JA=JA1
      IS=I1-2
      DO 14 I=IS,IMAX
   14 TEMP(I)=P(I,JB-1)
      DO 3000 I=I1,IMAX1
      IF(X(I).GE.X2)GO TO 3100
C     FLOW ABOVE AIRFOIL
      JB2=JB-2
      DO 3 J=JB2,JMAX1
      IF(YU(I).GT.E(J).AND.YU(I).LE.E(J+1))GO TO 4
    3 CONTINUE
    4 JA=J+1
      IF(JA.LE.JB)JA=JB+1
      F=FF(I)
      FP12=FFP12(I)
      FM12=FFM12(I)
      FM1=FFM1(I)
      FM32=FFM32(I)
      GB=A11/(1.+TAN(YU(I   )*PI2)**2)
      VV=SLU(I)**2
      UU=1.0
 4000 DU1(I)=-COS(ALP)+((1./(1.+VV/UU))*(1.-((1.+0.7*M*M*CPU(I   ))**0.28
     16-1.)*5./(M*M)))**0.5-F*((-4.*P1(JA-1)+P2(JA-1))/(2.*DS))
     2-F*(YU(I)-E(JA-1))*(P(I+1,JA)-P(I+1,JA-1)-P1(JA)+P1(JA-1))/(2.
     3*DS*DE)
      DU1(I)=DU1(I)/(F*(1.5/DS))
 4001 CONTINUE
      DU2(I)=-P(I,JA)+2.0*DU1(I)
      P(I,JA-1)=DU1(I)
      P(I,JA-2)=DU2(I)
      CALL SOLVE(JA,JMAX1)
      RS(JA)=RS(JA)-SUB(JA-1)*P(I,JA-1)
      RS(JMAX1)=RS(JMAX1)-SUP(JMAX1)*P(I,JMAX)
      CALL TRID(JA,JMAX1)
```

```
      DO 1 J=JA,JMAX1
      DP=ABS(P(I,J)-RS(J))
      IF(DP.GT.DPM)ICON=I
      IF(DP.GT.DPM)JCON=J
      IF(DP.GT.DPM)DPM=DP
      P2(J)=P1(J)
      P1(J)=P(I,J)
    1 P(I,J)=RS(J)
      JAM1=JA-1
      DO 2 J=JB,JAM1
      P2(J)=P1(J)
    2 P1(J)=P(I,J)
      VV=SLU(I)**2
      UU=1.0
 4002 DU1(I)=-COS(ALP)+((1./(1.+VV/UU))*(1.-((1.+0.7*M*M*CPU(I  ))**0.28
     16-1.)*5./(M*M)))**0.5-F*((-4.*P(I-1,JA-1)+P(I-2,JA-1))/(2.*DS))
     2-F*(YU(I)-E(JA-1))*(P(I+1,JA)-P(I+1,JA-1)-P(I-1,JA)+P(I-1,JA-1))/
     3(2.*DS*DE)
      DU1(I)=DU1(I)/(F*(1.5/DS))
 4003 CONTINUE
      DU2(I)=-P(I,JA)+2.0*DU1(I)
      P(I,JA-1)=DU1(I)
      P(I,JA-2)=DU2(I)
      P2(JMAX)=P1(JMAX)
      P1(JMAX)=P(I,JMAX)
      JA1=JA
 3000 CONTINUE
 3100 JA=JB1
      DO 5 I=IS,IMAX
      P(I,JB-1)=TEMP(I)
      TEMP(I)=P(I,JB+1)
      TEMP1=PB(I)
      PB(I)=P(I,JB)
    5 P(I,JB)=TEMP1
    6 TEMP1=TEMP4
      TEMP4=P1(JB)
      P1(JB)=TEMP1
      TEMP1=TEMP3
      TEMP3=P2(JB)
      P2(JB)=TEMP1
```

```
      DO 3500 I=I1,IMAX1
C     FLOW BELOW AIRFOIL
      IF(X(I).GE.X2)GO TO 3600
      JB2=JB+2
      DO 10 JJ=1,JB
      J=JB2-JJ
      IF(YL(I).GE.E(J).AND.YL(I).LT.E(J+1))GO TO 11
   10 CONTINUE
   11 JA=J
      IF(JA.GE.JB)JA=JB-1
      F=FF(I)
      FP12=FFP12(I)
      FM12=FFM12(I)
      FM1=FFM1(I)
      FM32=FFM32(I)
      GB=A11/(1.+TAN(YL(I   )*PI2)**2)
      VV=SLL(I)**2
      UU=1.0
 4004 DL1(I)=-COS(ALP)+((1./(1.+VV/UU))*(1.-((1.+0.7*M*M*CPL(I   ))**0.28
     16-1.)*5./(M*M)))**0.5-F*((-4.*P1(JA+1)+P2(JA+1))/(2.*DS))
     2-F*(YL(I)-E(JA+1))*(P(I+1,JA+1)-P(I+1,JA)-P1(JA+1)+P1(JA))/(2.
     3*DS*DE)
      DL1(I)=DL1(I)/(F*(1.5/DS))
 4005 CONTINUE
      DL2(I)=2.*DL1(I)-P(I,JA)
      P(I,JA+1)=DL1(I)
      P(I,JA+2)=DL2(I)
      CALL SOLVE(2,JA)
      RS(2)=RS(2)-SUB(1)*P(I,1)
      RS(JA)=RS(JA)-SUP(JA)*P(I,JA+1)
      CALL TRID(2,JA)
      DO 7 J=2,JA
      DP=ABS(P(I,J)-RS(J))
      IF(DP.LE.DPM)GO TO 8
      ICON=I
      JCON=J
      DPM=DP
    8 P2(J)=P1(J)
      P1(J)=P(I,J)
    7 P(I,J)=RS(J)
```

```
      JAM1=JA+1
      DO 9 J=JAM1,JB
      P2(J)=P1(J)
    9 P1(J)=P(I,J)
      VV=SLL(I)**2
      UU=1.0
 4006 DL1(I)=-COS(ALP)+((1./(1.+VV/UU))*(1.-((1.+0.7*M*M*CPL(I   ))**0.28
     16-1.)*5./(M*M)))**0.5-F*((-4.*P(I-1,JA+1)+P(I-2,JA+1))/(2.*DS))
     2-F*(YL(I)-E(JA+1))*(P(I+1,JA+1)-P(I+1,JA)-P(I-1,JA+1)+P(I-1,JA))/
     3(2.*DS*DE)
      DL1(I)=DL1(I)/(F*(1.5/DS))
 4007 CONTINUE
      DL2(I)=2.*DL1(I)-P(I,JA)
      P(I,JA+1)=DL1(I)
      P(I,JA+2)=DL2(I)
      P2(1)=P1(1)
      P1(1)=P(I,1)
      JB1=JA
 3500 CONTINUE
 3600 DO 12 I=IS,IMAX
      P(I,JB+1)=TEMP(I)
      TEMP1=PB(I)
      PB(I)=P(I,JB)
   12 P(I,JB)=TEMP1
      P1(JB)=TEMP4
      P2(JB)=TEMP3
      RETURN
      END
```

```
      SUBROUTINE FLOW2
C
C  **********   SOLVES FLOW IN THE DRIECT REGION  ******
C
      REAL M
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)
     1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
     1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99
     2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
     3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,
     4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1
     5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,
     1JMAX,JCON,JMAX1,NSSP,IW
      COMMON/BAKER/TEMP3,TEMP4
C     RELAXES FLOW ABOVE AND BELOW AIRFOIL
      JB=JMAX/2+1
      JB1=JB-1
      JA1=JB+1
      JB2=JB-2
      TEMP5=P1(JB)
      TEMP6=P2(JB)
      DO 12 I=ILE,I1
   12 TEMP(I)=P(I,JB-1)
      IF(ILE.GT.I11)GO TO 2001
      DO 2000 I=ILE,I11
C     FLOW ABOVE AIRFOIL
      F=FF(I)
      FP12=FFP12(I)
      FM12=FFM12(I)
      FM32=FFM32(I)
      GB=A11/(1.+TAN(PI2*YU(I))**2)
      DO 1 J=JB2,JMAX1
      IF(YU(I).GT.E(J).AND.YU(I).LE.E(J+1))GO TO 2
    1 CONTINUE
    2 JA=J+1
      IF(JA.LE.JB)JA=JB+1
      DU1(I)=SLU(I)*(COS(ALP)+F*((P(I+1,JA-1)-P1(JA-1))/(2.*DS)
     1+(YU(I)-E(JA-1))*(P(I+1,JA)-P(I+1,JA-1)-P1(JA)+P1(JA-1))/(2.*DS*DE
     2)))-SIN(ALP)-GB*((4.*P(I,JA)-P(I,JA+1))/(2.*DE)+(YU(I)-E(JA-1))*
```

```
     3(-2.*P(I,JA)+P(I,JA+1))/(DE**2))
      DU1(I)=DU1(I)/(-1.5*GB/DE     +(YU(I)-E(JA-1))/(DE**2)*GB)
      DU2(I)=-P(I,JA)+DU1(I)*2.0
      P(I,JA-1)=DU1(I)
      P(I,JA-2)=DU2(I)
      CALL SOLVE(JA,JMAX1)
      RS(JA)=RS(JA)-SUB(JA-1)*P(I,JA-1)
      RS(JMAX1)=RS(JMAX1)-SUP(JMAX1)*P(I,JMAX)
      CALL TRID(JA,JMAX1)
      DO 4 J=JA,JMAX1
      DP=ABS(P(I,J)-RS(J))
      IF(DP.GT.DPM)ICON=I
      IF(DP.GT.DPM)JCON=J
      IF(DP.GT.DPM)DPM=DP
      P2(J)=P1(J)
      P1(J)=P(I,J)
    4 P(I,J)=RS(J)
      JAM1=JA-1
      DO 5 J=JB,JAM1
      P2(J)=P1(J)
    5 P1(J)=P(I,J)
      P2(JMAX)=P1(JMAX)
      P1(JMAX)=P(I,JMAX)
      JA1=JA
      DU1(I)=SLU(I)*(COS(ALP)+F*((P(I+1,JA-1)-P(I-1,JA-1))/(2.*DS)
     1+(YU(I)-E(JA-1))*(P(I+1,JA)-P(I+1,JA-1)-P(I-1,JA)+P(I-1,JA-1))/
     2(2.*DS*DE)))-SIN(ALP)-GB*((4.*P(I,JA)-P(I,JA+1))/(2.*DE)+
     3(YU(I)-E(JA-1))*(-2.*P(I,JA)+P(I,JA+1))/(DE**2))
      DU1(I)=DU1(I)/(-1.5*GB/DE+(YU(I)-E(JA-1))/(DE**2)*GB)
      DU2(I)=-P(I,JA)+DU1(I)*2.0
      P(I,JA-1)=DU1(I)
      P(I,JA-2)=DU2(I)
 2000 CONTINUE
 2001 CONTINUE
      TEMP3=P2(JB)
      TEMP4=P1(JB)
C     FLOW BELOW AIRFOIL
      DO 8 I=ILE,I1
      P(I,JB-1)=TEMP(I)
      TEMP(I)=P(I,JB+1)
```

```
      TEMP1=PB(I)
      PB(I)=P(I,JB)
      P(I,JB)=TEMP1
    8 CONTINUE
      P1(JB)=TEMP5
      P2(JB)=TEMP6
      IF(ILE.GT.I11)GO TO 2501
      DO 2500 I=ILE,I11
      GB=A11/(1.+TAN(PI2*YL(I))**2)
      F=FF(I)
      FP12=FFP12(I)
      FM12=FFM12(I)
      FM32=FFM32(I)
      DO 6 JJ=1,JMAX1
      J=JB-JJ+2
      IF(YL(I).GE.E(J).AND.YL(I).LT.E(J+1))GO TO 7
    6 CONTINUE
    7 JA=J
      IF(JA.GE.JB)JA=JB-1
      DL1(I)=SLL(I)*(COS(ALP)+F*((P(I+1,JA+1)-P1(JA+1))/(2.*DS)+
     1(YL(I)-E(JA+1))*(P(I+1,JA+1)-P(I+1,JA)-P1(JA+1)+P1(JA))/(2.*DS*DE)
     2))-SIN(ALP)-GB*((-4.*P(I,JA)+P(I,JA-1))/(2.*DE)+(YL(I)-E(JA+1))*
     3(-2.*P(I,JA)+P(I,JA-1))/(DE**2))
      DL1(I)=DL1(I)/(1.5*GB/DE+(YL(I)-E(JA+1))/(DE**2)*GB)
      DL2(I)=2.*DL1(I)-P(I,JA)
      P(I,JA+1)=DL1(I)
      P(I,JA+2)=DL2(I)
      CALL SOLVE(2,JA)
      RS(2)=RS(2)-SUB(1)*P(I,1)
      RS(JA)=RS(JA)-SUP(JA)*P(I,JA+1)
      CALL TRID(2,JA)
      DO 9 J=2,JA
      DP=ABS(P(I,J)-RS(J))
      IF(DP.GT.DPM)ICON=I
      IF(DP.GT.DPM)JCON=J
      IF(DP.GT.DPM)DPM=DP
      P2(J)=P1(J)
      P1(J)=P(I,J)
    9 P(I,J)=RS(J)
      JAM1=JA+1
```

```
      DO 10 J=JAM1,JB
      P2(J)=P1(J)
   10 P1(J)=P(I,J)
      P2(1)=P1(1)
      P1(1)=P(I,1)
      JB1=JA
      DL1(I)=SLL(I)*(COS(ALP)+F*((P(I+1,JA+1)-P(I-1,JA+1))/(2.*DS)+
     1(YL(I)-E(JA+1))*(P(I+1,JA+1)-P(I+1,JA)-P(I-1,JA+1)+P(I-1,JA))/(2.
     2*DS*DE)))-SIN(ALP)-GB*((-4.*P(I,JA)+P(I,JA-1))/(2.*DE)+(YL(I)-E(
     3JA+1))*(-2.*P(I,JA)+P(I,JA-1))/(DE**2))
      DL1(I)=DL1(I)/(1.5*GB/DE+(YL(I)-E(JA+1))/(DE**2)*GB)
      DL2(I)=2.*DL1(I)-P(I,JA)
      P(I,JA+1)=DL1(I)
      P(I,JA+2)=DL2(I)
 2500 CONTINUE
 2501 CONTINUE
      DO 11 I=ILE,I1
      P(I,JB+1)=TEMP(I)
      TEMP1=PB(I)
      PB(I)=P(I,JB)
   11 P(I,JB)=TEMP1
      TEMP1=TEMP4
      TEMP4=P1(JB)
      P1(JB)=TEMP1
      TEMP1=TEMP3
      TEMP3=P2(JB)
      P2(JB)=TEMP1
      PB(ILE1)=P(ILE1,JB)
      PB(ILE-2)=P(ILE-2,JB)
      RETURN
      END
```

```
      SUBROUTINE WAKE
C
C  **********   SOLVES FLOW BEHIND THE AIRFOIL ******
C
      REAL M
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)
     1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
     1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99
     2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
     3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,
     4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1
     5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,
     1JMAX,JCON,JMAX1,NSSP,IW
      COMMON/ADAM/VJB,VJB1,VJBP1
C     RELAXES FLOW IN WAKE DIRECTLY
      IW=1
      DO 4000 I=ITE1,IMAX1
      F=FF(I)
      FP12=FFP12(I)
      FM12=FFM12(I)
      FM32=FFM32(I)
      CALL SOLVE(2,JMAX1)
      RS(2)=RS(2)-SUB(1)*P(I,1)
      RS(JMAX1)=RS(JMAX1)-SUP(JMAX1)*P(I,JMAX)
      IF(QQJBP1.LE.AAJBP1)GO TO 1
      IF(VJBP1.LT.0.)GO TO 1
      G=A11/(1.+TAN(PI2*E(JB+1))**2)
      GM32=A11/(1.+TAN(PI2*(E(JB+1)-1.5*DE))**2)
      RS(JB+1)=RS(JB+1)-(1.-QQJBP1/AAJBP1)*(VVJBP1/QQJBP1*G*GM32*CIR/
     1(DE**2))
    1 IF(QQJB.LE.AAJB)GO TO 2
      IF(VJB.LT.0.0)GO TO 3
      G=A11/(1.+TAN(PI2*E(JB))**2)
      GM12=A11/(1.+TAN(PI2*(E(JB)-0.5*DE))**2)
      RS(JB)=RS(JB)+(1.-QQJB/AAJB)*(VVJB/QQJB*G*GM12*CIR/(DE**2))
     1-UUJB/QQJB*G*GM12*CIR/(DE**2)
      GO TO 4
    3 G=A11/(1.+TAN(PI2*E(JB))**2)
      GM12=A11/(1.+TAN(PI2*(E(JB)-0.5*DE))**2)
```

```
      RS(JB)=RS(JB)-UUJB/QQJB*G*GM12*CIR/(DE**2)
      GO TO 4
    2 G=A11/(1.+TAN(PI2*E(JB))**2)
      C=(1.-VVJB/AAJB)*G
      GM12=A11/(1.+TAN(PI2*(E(JB)-0.5*DE))**2)
      RS(JB)=RS(JB)-C*GM12*CIR/(DE**2)
    4 IF(QQJB1.GT.AAJB1)GO TO 5
      G=A11/(1.+TAN(PI2*E(JE-1))**2)
      C=(1.-VVJB1/AAJB1)*G
      GP12=A11/(1.+TAN(PI2*(E(JB-1)+0.5*DE))**2)
      RS(JB-1)=RS(JB-1)+C*GP12*CIR/(DE**2)
      GO TO 6
    5 IF(VJB1.LT.0.0)GO TO 7
      G=A11/(1.+TAN(PI2*E(JB-1))**2)
      GP12=A11/(1.+TAN(PI2*(E(JB-1)+0.5*DE))**2)
      RS(JB-1)=RS(JB-1)+UUJB1/QQJB1*G*GP12*CIR/(DE**2)
      GO TO 6
    7 G=A11/(1.+TAN(PI2*E(JB-1))**2)
      GP12=A11/(1.+TAN(PI2*(E(JB-1)+0.5*DE))**2)
      RS(JB-1)=RS(JB-1)-(1.-QQJB1/AAJB1)*VVJB1/QQJB1*G*GP12*CIR/(DE**2)
     1+UUJB1/QQJB1*G*GP12*CIR/(DE**2)
    6 CALL TRID(2,JMAX1)
      DO 8 J=2,JMAX1
      DP=ABS(P(I,J)-RS(J))
      IF(DP.LT.DPM)GO TO 9
      ICON=I
      JCON=J
      DPM=DP
    9 P2(J)=P1(J)
      P1(J)=P(I,J)
    8 P(I,J)=RS(J)
      P2(1)=P1(1)
      P2(JMAX)=P1(JMAX)
      P1(1)=P(I,1)
      P1(JMAX)=P(I,JMAX)
 4000 CONTINUE
      DO 10 I=ITE1,IMAX1
      PE(I)=P(I,JB)-CIR
   10 CONTINUE
      PE(ILE1)=P(ILE1,JB)
```

```
      IW=0
      RETURN
      END



      SUBROUTINE SHAPE
C
C    *********    COMPUTES SHAPE OF AIRFOIL IN INVERSE DESIGN CASE********
C
      REAL M
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)
     1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
     1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99
     2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
     3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,
     4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1
     5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,
     1JMAX,JCON,JMAX1,NSSP,IW
      COMMON/TAMU/DELTAY
      DELTAY=0.0
      IF(INV.EQ.0)I1=ILE+2
      IP1=I1
      DO 1 I=IP1,ITE1
      YOLD=YU(I)
      JB2=JB-2
      DO 3 J=JB2,JMAX1
      IF(YU(I-1).GT.E(J).AND.YU(I-1).LE.E(J+1))GO TO 4
    3 CONTINUE
    4 JA=J+1
      L=I-1
      IF(JA.LE.JB)JA=JB+1
      F=FF(L)
      U=QI*(COS(ALP)+F*((P(I,JA-1)-P(I-2,JA-1))/(2.*DS)+(YU(L)-E(JA-1)
     1)*(P(I,JA)-P(I,JA-1)-P(I-2,JA)+P(I-2,JA-1))/(2.*DS*DE)))
      GB=A11/(1.+TAN(PI2*YU(I-1))**2)
      V=QI*(SIN(ALP)+GB*((-3.*P(L,JA-1)+4.*P(L,JA)-P(L,JA+1))/(2.*
```

```
     1DE)+(YU(L)-E(JA-1))*(P(L,JA-1)-2.*P(L,JA)+P(L,JA+1))/(DE**2)))
      FY=(GB/F)*V/U
      IF(I.EQ.I1)GO TO 14
      SLU(I-1)=V/U
   14 CONTINUE
      FK1=DS*FY
      YN=YU(L)+0.5*FK1
      F=FFM12(I)
      DO 20  J=JB2,JMAX1
      IF(YN.GT.E(J).AND.YN.LE.E(J+1))GO TO 50
   20 CONTINUE
   50 JA=J+1
      IF(JA.LE.JB)JA=JB+1
      U=QI*(COS(ALP)+F*((P(I,JA-1)-P(I-1,JA-1))/DS+(YN-E(JA-1))*
     1(P(I,JA)-P(I,JA-1)-P(L,JA)+P(L,JA-1))/(DS*DE)))
      GB=A11/(1.+TAN(PI2*YN)**2)
      V=QI*(SIN(ALP)+GB*((-3.*(P(I,JA-1)+P(L,JA-1))+4.*(P(I,JA)+P(L,
     1JA))-P(I,JA+1)-P(L,JA+1))/(4.*DE)+(YN-E(JA-1))*0.5*(P(I,JA-1)
     2+P(I-1,JA-1)-2.*(P(I,JA)+P(L,JA))+P( I.JA+1)+P(L,JA+1))/(DE**2)))
      FK2=GB/F*DS*V/U
      YN=YU(L)+0.5*FK2
      DO 21  J=JB2,JMAX1
      IF(YN.GT.E(J).AND.YN.LE.E(J+1))GO TO 22
   21 CONTINUE
   22 JA=J+1
      IF(JA.LE.JB)JA=JB+1
      U=QI*(COS(ALP)+F*((P(I,JA-1)-P(I-1,JA-1))/DS+(YN-E(JA-1))*
     1(P(I,JA)-P(I,JA-1)-P(L,JA)+P(L,JA-1))/(DS*DE)))
      GB=A11/(1.+TAN(PI2*YN)**2)
      V=QI*(SIN(ALP)+GE*((-3.*(P(I,JA-1)+P(L,JA-1))+4.*(P(I,JA)+P(L,
     1JA))-P(I,JA+1)-P(L,JA+1))/(4.*DE)+(YN-E(JA-1))*0.5*(P(I,JA-1)
     2+P(I-1,JA-1)-2.*(P(I,JA)+P(L,JA))+P( I,JA+1)+P(L,JA+1))/(DE**2)))
      FK3=GB/F*DS*V/U
      YN=YU(L)+FK3
      F=FF(I)
      DO 2  J=JB2,JMAX1
      IF(YN.GT.E(J).AND.YN.LE.E(J+1))GO TO 5
    2 CONTINUE
    5 JA=J+1
      IF(JA.LE.JB)JA=JB+1
```

```
      U=QI*(COS(ALP)+F*((P(I+1,JA-1)-P(I-1,JA-1))/(2.*DS)+(YN-E(JA-1))*
     1(P(I+1,JA)-P(I+1,JA-1)-P(I-1,JA)+P(I-1,JA-1))/(2.*DS*DE)))
      GB=A11/(1.+TAN(PI2*YN)**2)
      V=QI*(SIN(ALP)+GB*((-3.*P(I,JA-1)+4.*P(I,JA)-P(I,JA+1))/(2.*DE)
     1+(YN-E(JA-1))*(P(I,JA-1)-2.*P(I,JA)+P(I,JA+1))/(DE**2)))
      FK4=GB/F*DS*V/U
      YU(I)=YU(L)+(FK1+2.*FK2+2.*FK3+FK4)/6.
      IF(I.GT.ITE)GO TO 1
      CHANGE=ABS(YU(I)-YOLD)
      IF(CHANGE.GT.DELTAY)DELTAY=CHANGE
    1 CONTINUE
      DO 6 I=ILE1,IMAX
      TEMP1=P(I,JB)
      P(I,JB)=PB(I)
    6 PB(I)=TEMP1
      DO 7 I=IP1,ITE1
      YOLD=YL(I)
      JB2=JB+2
      DO 8 JJ=1,JB
      J=JB2-JJ
      IF(YL(I-1).GE.E(J).AND.YL(I-1).LT.E(J+1))GO TO 9
    8 CONTINUE
    9 JA=J
      IF(JA.GE.JB)JA=JB-1
      L=I-1
      F=FF(L)
      U=QI*(COS(ALP)+F*((P(I,JA+1)-P(I-2,JA+1))/(2.*DS)+(YL(L)-E(JA+1))*
     1(P(I,JA+1)-P(I,JA)-P(I-2,JA+1)+P(I-2,JA))/(2.*DS*DE)))
      GB=A11/(1.+TAN(PI2*YL(L))**2)
      V=QI*(SIN(ALP)+GB*((3.*P(L,JA+1)-4.*P(L,JA)+P(L,JA-1))/(2.*DE)+
     1(YL(L)-E(JA+1))*(P(L,JA+1)-2.*P(L,JA)+P(L,JA-1))/(DE**2)))
      FY=GB/F*V/U
      IF(I.EQ.I1)GO TO 15
      SLL(I-1)=V/U
   15 CONTINUE
      FK1=DS*FY
      YN=YL(L)+0.5*FK1
      F=FFM12(I)
      DO 25 JJ=1,JB
      J=JB2-JJ
```

```
      IF(YN.GE.E(J).AND.YN.LT.E(J+1))GO TO 26
   25 CONTINUE
   26 JA=J
      IF(JA.GE.JB)JA=JB-1
      U=QI*(COS(ALP)+F*((P(I,JA+1)-P(L,JA+1))/DS+(YN-E(JA+1))*(P(I,JA+1)
     1-P(I,JA)-P(L,JA+1)+P(L,JA))/(DS*DE)))
      GB=A11/(1.+TAN(PI2*YN)**2)
      V=QI*(SIN(ALP)+GB*((3.*(P(I,JA+1)+P(L,JA+1))-4.*(P(I,JA)+P(L,JA))
     1+P(I,JA-1)+P(L,JA-1))/(4.*DE)+(YN-E(JA+1))*(P(I,JA+1)+P(L,JA+1)
     2-2.*(P(I,JA)+P(L,JA))+P(I,JA-1)+P(L,JA-1))*0.5/(DE**2)))
      FK2=GB/F*DS*V/U
      YN=YL(L)+0.5*FK2
      DO 27 JJ=1,JB
      J=JB2-JJ
      IF(YN.GE.E(J).AND.YN.LT.E(J+1))GO TO 28
   27 CONTINUE
   28 JA=J
      IF(JA.GE.JB)JA=JB-1
      U=QI*(COS(ALP)+F*((P(I,JA+1)-P(L,JA+1))/DS+(YN-E(JA+1))*(P(I,JA+1)
     1-P(I,JA)-P(L,JA+1)+P(L,JA))/(DS*DE)))
      GB=A11/(1.+TAN(PI2*YN)**2)
      V=QI*(SIN(ALP)+GB*((3.*(P(I,JA+1)+P(L,JA+1))-4.*(P(I,JA)+P(L,JA))
     1+P(I,JA-1)+P(L,JA-1))/(4.*DE)+(YN-E(JA+1))*(P(I,JA+1)+P(L,JA+1)
     2-2.*(P(I,JA)+P(L,JA))+P(I,JA-1)+P(L,JA-1))*0.5/(DE**2)))
      FK3=GB/F*DS*V/U
      YN=YL(L)+FK3
      F=FF(I)
      DO 10 JJ=1,JB
      J=JB2-JJ
      IF(YN.GE.E(J).AND.YN.LT.E(J+1))GO TO 11
   10 CONTINUE
   11 JA=J
      IF(JA.GE.JB)JA=JB-1
      U=QI*(COS(ALP)+F*((P(I+1,JA+1)-P(I-1,JA+1))/(2.*DS)+(YN-E(JA+1))*
     1(P(I+1,JA+1)-P(I+1,JA)-P(I-1,JA+1)+P(I-1,JA))/(2.*DS*DE)))
      GB=A11/(1.+TAN(PI2*YN)**2)
      V=QI*(SIN(ALP)+GB*((3.*P(I,JA+1)-4.*P(I,JA)+P(I,JA-1))/(2.*DE)+
     1(YN-E(JA+1))*(P(I,JA+1)-2.*P(I,JA)+P(I,JA-1))/(DE**2)))
      FK4=GB/F*DS*V/U
      YL(I)=YL(L)+(FK1+2.*FK2+2.*FK3+FK4)/6.
```

```
      IF(I.GT.ITE)GO TO 7
      CHANGE=ABS(YL(I)-YOLD)
      IF(CHANGE.GT.DELTAY)DELTAY=CHANGE
    7 CONTINUE
      DO 12 I=ILE1,IMAX
      TEMP1=P(I,JB)
      P(I,JB)=PB(I)
   12 PB(I)=TEMP1
      RETURN
      END
      SUBROUTINE TRID(IL,IH)
C
C ******   TRIDIAGONAL EQUATION SOLVER *******
C
      REAL M
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)
     1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
     1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99
     2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
     3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,
     4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1
     5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,
     1JMAX,JCON,JMAX1,NSSP,IW
      N=IH
      NN=N-1
      SUP(IL)=SUP(IL)/D(IL)
      RS(IL)=RS(IL)/D(IL)
      IDUM=IL+1
      DO 10 L=IDUM,N
      II=L-1
      D(L)=D(L)-SUP(II)*SUB(II)
      IF(L.EQ.N)GO TO 10
      SUP(L)=SUP(L)/D(L)
   10 RS(L)=(RS(L)-SUB(II)*RS(II))/D(L)
      DO 20 K=IL,NN
      L=N-K+IL-1
   20 RS(L)=RS(L)-SUP(L)*RS(L+1)
      RETURN
      END
```

```
      SUBROUTINE HALVE
C
C    ******   THIS SUBROUTINE HALVES THE GRID SPACING ETC. ******
C
      REAL M
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)
     1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
     1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99
     2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
     3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,
     4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1
     5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
      COMMON I,ITE,ITE1,ILE,ILE1,II,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,
     1JMAX,JCON,JMAX1,NSSP,IW
      IMAX=2*IMAX-1
      JMAX=2*JMAX-1
      IMAX1=IMAX-1
      JMAX1=JMAX-1
      J=JMAX1/2+1
      JJ=JMAX
    1 I=IMAX1/2+1
      II=IMAX
    2 P(II,JJ)=P(I,J)
      I=I-1
      II=II-2
      IF(I.GT.0)GO TO 2
      J=J-1
      JJ=JJ-2
      IF(J.GT.0)GO TO 1
      DO 3 J=1,JMAX,2
      DO 3 I=2,IMAX1,2
    3 P(I,J)=0.5*(P(I+1,J)+P(I-1,J))
      DO 4 I=1,IMAX
      DO 4 J=2,JMAX1,2
    4 P(I,J)=0.5*(P(I,J+1)+P(I,J-1))
      I=IMAX1/2+1
      II=IMAX
    5 PB(II)=PB(I)
      I=I-1
      II=II-2
```

```
      IF(I.GT.0)GO TO 5
      DO 6 I=2,IMAX1,2
    6 PB(I)=0.5*(PB(I+1)+PB(I-1))
      RETURN
      END


      SUBROUTINE PLOT(NO,A,N,M,NL,NS)
C
C   ******    THIS CREATES A PLOT OF RESULTS ON THE STANDARD PRINT OUT **
C
      DIMENSION OUT(101),YPR(11),ANG(9),A(500)
    1 FORMAT(1H1,60X,7H CHART ,I3,//)
    2 FORMAT(1H ,F11.4,5X,101A1)
    3 FORMAT(1H )
    7 FORMAT(1H ,16X,101H.         .         .         .         .
     1    .         .         .         .         .         .)
    8 FORMAT(1H0,9X,11F10.3)
      DATA BLANK/1H /,ANG/1HU,1HL,1HT,1HB,1H5,1H6,1H7,1H8,1H9/
      NLL=NL
      IF(NS)  16,  16,  10
   10 DO 15 I=1,N
      DO 14 J=I,N
      IF(A(I)-A(J)) 14,  14,  11
   11 L=I-N
      LL=J-N
      DO 12 K=1,M
      L=L+N
      LL=LL+N
      F=A(L)
      A(L)=A(LL)
   12 A(LL)=F
   14 CONTINUE
   15 CONTINUE
   16 IF(NLL) 20,  18,  20
   18 NLL=50
   20 WRITE(6,1)NO
      XSCAL=(A(N)-A(1))/(FLOAT(NLL-1))
      M1=N+1
```

```
      YMIN=A(M1)
      YMAX=YMIN
      M2=M*N
      DO 40 J=M1,M2
      IF(A(J)-YMIN)  28,26,26
   26 IF(A(J)-YMAX) 40,40,30
   28 YMIN=A(J)
      GO TO 40
   30 YMAX=A(J)
   40 CONTINUE
      YSCAL=(YMAX-YMIN)/100.0
      XB=A(1)
      L=1
      MY=M-1
      I=1
   45 F=I-1
      XPR=XB+F*XSCAL
      IF(A(L)-XPR) 50,50,70
   50 DO 55 IX=1,101
   55 OUT(IX)=BLANK
      DO 60 J=1,MY
      LL=L+J*N
      JP=((A(LL)-YMIN)/YSCAL)+1.0
      OUT(JP)=ANG(J)
   60 CONTINUE
      WRITE(6,2)XPR,(OUT(IZ),IZ=1,101)
      L=L+1
      GOTO80
   70 WRITE(6,3)
   80 I=I+1
      IF(I-NLL) 45,84,86
   84 XPR=A(N)
      GO TO 50
   86 WRITE(6,7)
      YPR(1)=YMIN
      DO 90 KN=1,9
   90 YPR(KN+1)=YPR(KN)+YSCAL*10.0
      YPR(11)=YMAX
```

```
      WRITE(6,8)(YPR(IP),IP=1,11)
      RETURN
      END

      SUBROUTINE ARC(XI,YI,XO,YO,SI,SO,XP,YP,D1Y,D2Y,D3Y,DERIX,DERFX,
     1DERIY,DERFY,NI,NO,INT)
C
C  ********    DETERMINES THE ARC LENGTH OF THE AIRFOIL POINTS *****
C
      DIMENSION XI(1),YI(1),XO(1),YO(1),SI(1),SO(1),XP(1),YP(1),D1Y(1),
     1D2Y(1),D3Y(1)
C         SI - INPUT CHORD LENGTH        SO - OUTPUT CHORD LENGTH
C         COMPUTE ARC LENGTH SI USING CIRCULAR ARC SEGMENTS
C         INT=1      SPLINE XI AND YI VS SI
      EPSI=1.E-10
      N1=NI-1
      SI(1)=0.
                    H1=0.
      DX1=XI(2)-XI(1)
                              DY1=YI(2)-YI(1)
      C1=SQRT(DX1**2+DY1**2)
      SI(2)=C1
      IF(NI.EQ.2)RETURN
      DO 1 I=2,N1
      DX1=XI(I)-XI(I-1)
                                 DY1=YI(I)-YI(I-1)
      DX2=XI(I+1)-XI(I)
                                 DY2=YI(I+1)-YI(I)
      DX=XI(I+1)-XI(I-1)
                                  DY=YI(I+1)-YI(I-1)
      C2=SQRT(DX2**2+DY2**2)
      C=SQRT(DX**2+DY**2)
      A=(DY1*DX-DY*DX1)/2.
      H=4.*A/(C*C1*C2)
      HAV=(H1+H)/2.
      DS=C1*(1.+(C1/2.*HAV)**2/6.)
      SI(I)=SI(I-1)+DS
      C1=C2
      H1=H
    1 CONTINUE
```

```
      DS=C1*(1.+(C1/2.*H)**2/6.)
      SI(NI)=SI(NI-1)+DS
      IF(INT.NE.1)RETURN
    2 CONTINUE
C        SPLINE XI AS A FUNCTION OF SI
      CALL SPLINE(SI,XI,SO,XO,XP,D1Y,D2Y,D3Y,1,3,DERIX,DERFX,NI,NO,0)
    3 CONTINUE
C        SPLINE YI AS A FUNCTION OF SI
      CALL SPLINE(SI,YI,SO,YO,YP,D1Y,D2Y,D3Y,1,3,DERIY,DERFY,NI,NO,1)
      RETURN
      END


      SUBROUTINE SPLINE(XIN,YIN,XOUT,YOUT,DYDX,D1Y,D2Y,D3Y,NDERI,NDERF,
     1DERIVI,DERIVF,NIN,NOUT,INTERP)
C     E B KLUNKER          JANUARY 1973
C     COMPUTE A CUBIC SPLINE THROUGH THE SET OF POINTS XIN(I),YIN(I)
C     XIN  MUST BE MONOTONIC
C     XIN,YIN        INPUT INDEPENDENT AND DEPENDENT VARIABLES
C     XOUT,YOUT      OUTPUT INDEPENDENT AND DEPENDENT VARIABLES
C     D1Y,D2Y,D3Y    1ST, 2ND, AND 3RD DERIVATIVE AT SPLINE POINTS XIN
C     DYDX           DERIVATIVE AT XOUT
C     NIN ,NOUT      NUMBER OF INPUT AND OUTPUT VALUES
C     NDERI          ORDER OF DERIVATIVE AT INITIAL SPLINE POINT (1,2,OR 3
C     NDERF          ORDER OF DERIVATIVE AT FINAL SPLINE POINT (1,2,OR 3)
C     DERIVI         VALUE OF DERIVATIVE AT INITIAL SPLINE POINT
C     INTERP NE 1    INTERPOLATE FOR GIVEN  VALUES YOUT
C     NTIMES NE 1    SPLINE COEFFICIENTS ARE NOT RECOMPUTED
      DIMENSION XIN(1),YIN(1),XOUT(1),YOUT(1),DYDX(1),D1Y(1),D2Y(1),
     1D3Y(1)
      EPSI1=-1.E-10
                         EPSI2=-EPSI1
      NIM1=NIN-1
      DX=XIN(2)-XIN(1)
                           I=2
      IF(DX.EQ.0.)GO TO 35
      DF=(YIN(2)-YIN(1))/DX
      IF(NDERI-2)1,2,3
    1 C=.5
```

```
           F=3.*(DF-DERIVI)/DX
      GO TO 4
    2 C=0.
           F=DERIVI
      GO TO 4
    3 C=-1.
          F=-DX*DERIVI
C        FORWARD LOOP OF TRIDIAGONAL MATRIX COMPUTATION
    4 D1Y(1)=-C
                D2Y(1)=F
      DO 5 I=2,NIM1
      DX1=XIN(I+1)-XIN(I)
      IF(DX1.EQ.0.)GO TO 36
      DF1=(YIN(I+1)-YIN(I))/DX1
      B=2.*(DX+DX1)
      F=6.*(DF1-DF)
      DENOM=B+DX*D1Y(I-1)
      D2Y(I)=(F-DX*D2Y(I-1))/DENOM
      D1Y(I)=-DX1/DENOM
      DX=DX1
      DF=DF1
    5 CONTINUE
      I=NIN
      IF(NDERF-2)6,7,8
    6 A=.5
          F=-3.*(DF1-DERIVF)/DX1
      GO TO 9
    7 A=0.
          F=DERIVF
      GO TO 9
    8 A=-1.
           F=DX1*DERIVF
    9 DENOM=1.+A*D1Y(I-1)
      D2Y(I)=(F-A*D2Y(I-1))/DENOM
      D1Y(I)=0.
C        BACK SUBSTITUTION OF TRIDIAGONAL MATRIX COMPUTATION
      K=NIN
      DO 11 I=1,NIM1
      K=K-1
      D2Y(K)=D2Y(K)+D1Y(K)*D2Y(K+1)
```

```
   10 DX1=XIN(K+1)-XIN(K)
      DF1=(YIN(K+1)-YIN(K))/DX1
      D1Y(K+1)=DF1+DX1/6.*(D2Y(K)+2.*D2Y(K+1))
      D3Y(K+1)=(D2Y(K+1)-D2Y(K))/DX1
   11 CONTINUE
      D1Y(1)=DF1-DX1/6.*(2.*D2Y(1)+D2Y(2))
      D3Y(1)=D3Y(2)
      IF(INTERP.NE.1)GO TO 16
C        INTERPOLATE FOR GIVEN VALUES OF XOUT
      DO 15 J=1,NOUT
      DO 12 I=1,NIN
      DX=XIN(I)-XOUT(J)
      IF(DX.GE.EPSI1.AND.DX.LE.EPSI2)GO TO 13
      IF(DX.GE.EPSI2)GO TO 14
   12 CONTINUE
      GO TO 37
   13 YOUT(J)=YIN(I)
      DYDX(J)=D1Y(I)
      GO TO 15
   14 DX=XOUT(J)-XIN(I)
      YOUT(J)=YIN(I)+DX*(D1Y(I)+DX/2.*(D2Y(I)+DX/3.*D3Y(I)))
      DYDX(J)=D1Y(I)+DX*(D2Y(I)+DX/2.*D3Y(I))
   15 CONTINUE
      GO TO 23
C        INTERPOLATION FOR GIVEN VALUES OF YOUT
   16 DO 22 J=1,NOUT
      DO 17 I=1,NIN
      DY=YIN(I)-YOUT(J)
      IF(DY.GE.EPSI1.AND.DY.LE.EPSI2)GO TO 18
      IF(DY.GE.EPSI2)GO TO 19
   17 CONTINUE
      GO TO 38
   18 YOUT(J)=YIN(I)
      XOUT(J)=XIN(I)
      DYDX(J)=D1Y(I)
      GO TO 22
   19 DX=-DY/D1Y(I)
   20 YO=YIN(I)+DX*(D1Y(I)+DX/2.*(D2Y(I)+DX/3.*D3Y(I)))
      DY=YO-YOUT(J)
      IF(DY.GE.EPSI1.AND.DY.LE.EPSI2)GO TO 21
```

```
      YP=D1Y(I)+DX*(D2Y(I)+DX/2.*D3Y(I))
      DELX=-DY/YP
      DX=DX+DELX
      GO TO 20
   21 XOUT(J)=XIN(I)+DX
      DYDX(J)=D1Y(I)+DX*(D2Y(I)+DX/2.*D3Y(I))
   22 CONTINUE
   23 RETURN
   35 PRINT 100
      PRINT 101,XIN(1),XIN(2)
      STOP
   36 PRINT 100
      PRINT 102,I,XIN(I),XIN(I+1)
      STOP
   37 PRINT 100
      PRINT 103,J,XOUT(J),XIN(NIN)
      STOP
   38 PRINT 100
      PRINT 104,J,YOUT(J),YIN(NIN)
      STOP
C
  100 FORMAT(/5X,'SUBROUTINE SPLINE'/)
  101 FORMAT(/5X,'ERROR IN INPUT   XIN(1)='E12.4,5X,'XIN(2)='E12.4/)
  102 FORMAT(/5X,'ERROR IN INPUT   I='I5,5X,'XIN(I)='E12.4,5X,'XIN(I+1)=
     1'E12.4/)
  103 FORMAT(/5X,'XOUT(J) IS OUT OF RANGE   J='I5,5X,'XOUT(J)='E12.4,5X,
     1'XIN(NIN)='E12.4/)
  104 FORMAT(/5X,'YOUT(J) IS OUT OF RANGE   J='I5,5X,'YOUT(J)='E12.4,5X,
     1'YIN(NIN)='E12.4/)
C
      END
```

APPENDIX D
SAMPLE CASES

The input and output for two sample cases are presented on the following pages. Case 1 is a typical inverse design solution, and Case 2 is the analysis of a typical aft-cambered airfoil. Note that Case 2 only has two grid halvings. In actual usage, an analysis case would usually have one additional grid halving and would use a non-zero value for CDCORR in order to obtain accurate drag values (see Appendix B).

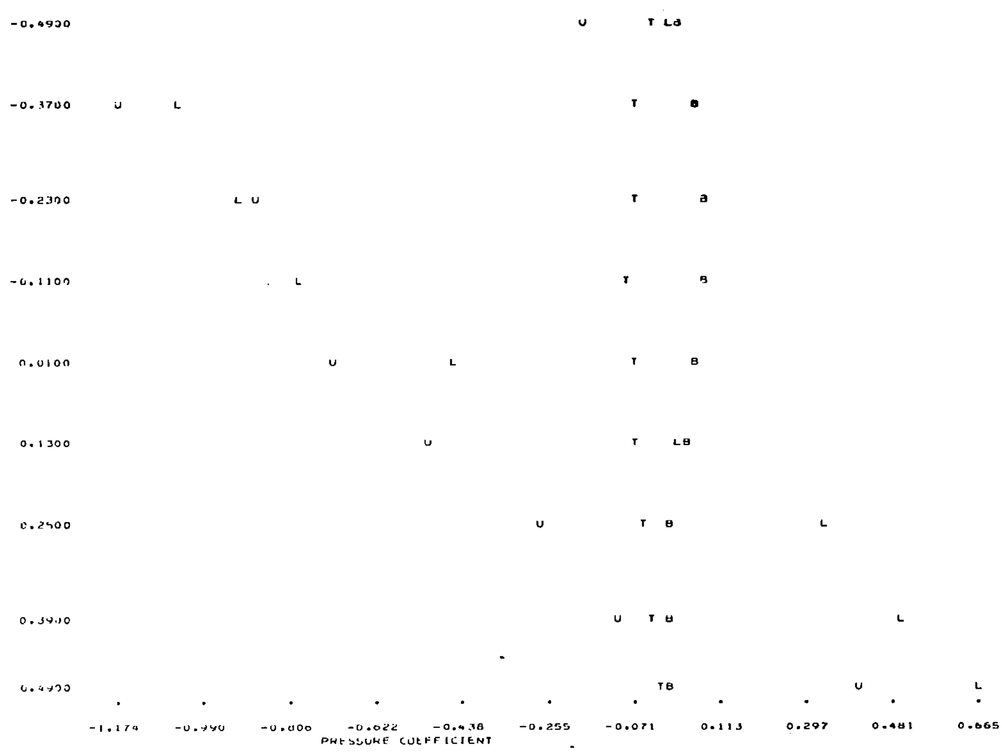
Sample Case No. 1 - Inverse Airfoil Design

```
MACH 0.72 AIRFOIL DESIGN -- SAMPLE CASE 1
&FINP M=0.72,X1=-0.38,X2=0.5,CONV=1.E-06,RN=20.95E06,XIBDLY=-0.34,XSEP=0.46,
&END
&IINP INV=1,&END
  0.50    10000.0
  82
  0.00000   0.00000   0.00012   0.00441   0.00082   0.00872   0.00210   0.01291
  0.00397   0.01693   0.00647   0.02072   0.00966   0.02426   0.01361   0.02759
  0.01833   0.03076   0.02379   0.03377   0.02999   0.03665   0.03692   0.03940
  0.04457   0.04201   0.05293   0.04449   0.06199   0.04685   0.07174   0.04909
  0.08217   0.05122   0.09327   0.05325   0.10499   0.05519   0.11733   0.05703
  0.13025   0.05878   0.14374   0.06044   0.15775   0.06200   0.17228   0.06347
  0.18729   0.06485   0.20276   0.06612   0.21866   0.06729   0.23495   0.06836
  0.25163   0.06932   0.26865   0.07018   0.28598   0.07091   0.30361   0.07154
  0.32150   0.07204   0.33961   0.07242   0.35793   0.07268   0.37642   0.07280
  0.39505   0.07279   0.41380   0.07264   0.43262   0.07234   0.45150   0.07188
  0.47042   0.07126   0.48934   0.07046   0.50825   0.06947   0.52714   0.06829
  0.54599   0.06692   0.56481   0.06536   0.58359   0.06361   0.60232   0.06171
  0.62098   0.05966   0.63956   0.05751   0.65802   0.05528   0.67634   0.05298
  0.69447   0.05065   0.71239   0.04830   0.73005   0.04595   0.74742   0.04360
  0.76446   0.04128   0.78113   0.03898   0.79740   0.03673   0.81323   0.03451
  0.82860   0.03235   0.84346   0.03023   0.85780   0.02818   0.87158   0.02619
  0.88478   0.02426   0.89736   0.02240   0.90932   0.02062   0.92061   0.01891
  0.93123   0.01728   0.94116   0.01574   0.95036   0.01428   0.95882   0.01291
  0.96654   0.01163   0.97349   0.01045   0.97967   0.00938   0.98504   0.00843
  0.98960   0.00759   0.99334   0.00687   0.99624   0.00626   0.99832   0.00578
  0.99958   0.00546   1.00000   0.00534
 0.0       1.0        0.0       0.0
  80
  0.00000   0.00000   0.00046  -0.00448   0.00150  -0.00901   0.00310  -0.01357
  0.00524  -0.01812   0.00794  -0.02263   0.01119  -0.02704   0.01502  -0.03135
  0.01943  -0.03553   0.02446  -0.03957   0.03009  -0.04347   0.03634  -0.04722
  0.04321  -0.05082   0.05068  -0.05427   0.05876  -0.05756   0.06743  -0.06068
  0.07669  -0.06364   0.08653  -0.06642   0.09694  -0.06902   0.10792  -0.07145
  0.11945  -0.07369   0.13151  -0.07576   0.14410  -0.07765   0.15719  -0.07936
  0.17077  -0.08090   0.18481  -0.08227   0.19929  -0.08346   0.21418  -0.08448
  0.22946  -0.08532   0.24509  -0.08599   0.26106  -0.08647   0.27733  -0.08677
```

0.29387	-0.08688	0.31065	-0.08678	0.32765	-0.08647	0.34483	-0.08594
0.36217	-0.08517	0.37965	-0.08414	0.39723	-0.08283	0.41492	-0.08121
0.43270	-0.07926	0.45059	-0.07694	0.46861	-0.07426	0.48680	-0.07121
0.50519	-0.06782	0.52379	-0.06413	0.54262	-0.06019	0.56167	-0.05606
0.58093	-0.05179	0.60037	-0.04745	0.61997	-0.04308	0.63968	-0.03874
0.65947	-0.03448	0.67928	-0.03035	0.69906	-0.02638	0.71876	-0.02262
0.73831	-0.01911	0.75767	-0.01588	0.77676	-0.01296	0.79552	-0.01039
0.81386	-0.00819	0.83172	-0.00637	0.84902	-0.00495	0.86566	-0.00392
0.88158	-0.00327	0.89669	-0.00299	0.91093	-0.00303	0.92423	-0.00338
0.93653	-0.00398	0.94780	-0.00479	0.95799	-0.00575	0.96708	-0.00682
0.97504	-0.00794	0.98187	-0.00905	0.98758	-0.01010	0.99218	-0.01105
0.99566	-0.01185	0.99809	-0.01243	0.99952	-0.01278	1.00000	-0.01290
0.0	-1.0	0.0	0.0				
-0.38	0.50						
-0.91200	-1.00600	-0.94000	-1.01500	-1.02000	-1.04900	-1.08100	-0.58300
-0.55700	-0.36000	-0.27500	-0.21500	-0.15400	-0.10000	-0.04100	
-0.69000	-0.68600	-0.63000	-0.62000	-0.61100	-0.56100	-0.30500	0.04000
0.21800	0.31500	0.40700	0.48100	0.53100	0.54700	0.57000	
-0.38	.500						
-1.10700	-1.10100	-1.09800	-1.06700	-1.07600	-1.07800	-1.06900	-1.06100
-1.06700	-1.05600	-1.05500	-1.01700	-0.91700	-0.80800	-0.70100	-0.60800
-0.52400	-0.43800	-0.37900	-0.32600	-0.28400	-0.24400	-0.20700	-0.16200
-0.14400	-0.11400	-0.08700	-0.05500	-0.00700			
-0.67300	-0.67700	-0.66500	-0.65700	-0.64900	-0.63700	-0.65000	-0.62800
-0.62100	-0.59000	-0.52800	-0.40800	-0.26700	-0.13900	0.00200	0.08900
0.16700	0.24100	0.29200	0.35700	0.39600	0.43600	0.47000	0.49800
0.51600	0.52800	0.53100	0.51000	0.53200			

```
MACH 0.72 AIRFOIL DESIGN -- SAMPLE CASE 1

                     X-Y GRID SYSTEM

   2 -0.1410E 01    3 -0.4900E 00    4 -0.3706E 00    5 -0.2485E 00    6 -0.1247E 00    7  0.0
   8  0.1247E 00    9  0.2485E 00   10  0.3706E 00   11  0.4900E 00   12  0.1410E 01
   2 -0.4261E 00    3 -0.1420E 00    4 -0.2303E-07    5  0.1420E 00    6  0.4261E 00

   MACH NO. IS 0.720 ANGLE OF ATTACK IS 0.0   DEGREES
          DIRECT SOLUTION TC      0.50
                     CASE NUMBER    100

   INVERSE DESIGN CASE
&FINP
M= 0.71999997    ,W=  1.6999998    ,X1= 0.50000000    ,X2=  10000.000    ,ALP= 0.0             ,EPS= 0.0            ,EPSS=
 0.39999998    ,X4= 0.48999995    ,S4=  2.0000000    ,CONV= 0.99999943E-06,A1= 0.24599999    ,A2= 0.14999998    ,A3=  3.8699999
RN=  20950000.     ,XIBDLY=-0.34999996    ,CIR= 0.0                ,CDCORR= 0.0              ,RDEL= 0.25000000    ,RDELFN= 0.12500000    ,
SP= 0.39999969E-02,XSEP= 0.45999998    ,RCPB= 0.19999959    ,CPB= 0.39999998    ,XMON= 0.46999997    ,XLSEP= 0.50000000    ,XPC=
 0.99999964E-01
&END
&IINP
IMAX=          13,JMAX=          7,IKASE=        100,INV=          1,MITER=        800,NHALF=          2,ITACT=          0,ISKP2=
          0,ISKP3=          0,ISKP4=          0,ITERP=          0,IREAD=          0,LP=        1000,ITEUPC=          0,ITELWC=
          0
&END

                     AIRFOIL COORDINATES
    X         YU         YL        UPPER SLOPE LOWER SLOPE
 -0.49000   0.02458  -0.02552   0.93718  -1.31772
 -0.37058   0.05867  -0.07542   0.12998  -0.16455
 -0.24852   0.06931  -0.08621   0.05419  -0.03119
 -0.12470   0.07280  -0.08442   0.00332   0.06268
  0.0       0.06993  -0.06880  -0.05300   0.18773
  0.12470   0.05924  -0.04203  -0.11419   0.22123
  0.24852   0.04345  -0.01738  -0.13570   0.16651
  0.37058   0.02634  -0.00368  -0.14519   0.04513
  0.49000   0.00752  -0.01058  -0.18798  -0.20660
ITERATION  10 CIR =  0.10732 DPM =  0.00441951 AT  5  3 NSSP =    3 DELTAY OR DELSTAR =  0.0
ITERATION  20 CIR =  0.13470 DPM =  0.00231314 AT 12  3 NSSP =    4 DELTAY OR DELSTAR =  0.0
ITERATION  30 CIR =  0.15451 DPM =  0.00187090 AT 10  3 NSSP =    4 DELTAY OR DELSTAR =  0.0
ITERATION  40 CIR =  0.16997 DPM =  0.00133852 AT 10  3 NSSP =    4 DELTAY OR DELSTAR =  0.0
ITERATION  50 CIR =  0.18249 DPM =  0.00126487 AT 10  3 NSSP =    5 DELTAY OR DELSTAR =  0.0
CP BY BACKWARD DIFFERENCES
X          CPU          CPL
    -0.490     -0.174      0.013
    -0.371     -1.174     -1.043
    -0.249     -0.633     -0.823
    -0.125     -0.677     -0.789
     0.0       -0.798     -0.751
     0.125     -0.607      0.211
     0.249     -0.174      0.420
     0.371     -0.087      0.540
     0.490      0.342      0.575
     1.410      0.063      0.049
CP BY CENTRAL DIFFERENCES
X          CPU          CPL
    -0.490     -0.174      0.013
    -0.371     -1.174     -1.043

    -0.249     -0.871     -0.909
    -0.125     -0.785     -0.780
     0.0       -0.712     -0.442
     0.125     -0.501      0.035
     0.249     -0.264      0.337
     0.371     -0.103      0.508
     0.490      0.416      0.665
     1.410      0.028      0.028
 CP BY CENTRAL DIFFERENCES
X          CPU          CPL
    -0.490     -0.174      0.013
    -0.371     -1.174     -1.043
    -0.249     -0.871     -0.909
    -0.125     -0.785     -0.780
     0.0       -0.712     -0.442
     0.125     -0.501      0.035
     0.249     -0.264      0.337
     0.371     -0.103      0.508
     0.490      0.416      0.665
     1.410      0.028      0.028
X            YU          YL          SLU        SLL
  -0.49000   0.02458  -0.02552   0.93718  -1.31772
  -0.37058   0.05867  -0.07542   0.12998  -0.16455
  -0.24852   0.06931  -0.08621   0.05419  -0.03119
  -0.12470   0.07280  -0.08442   0.00332   0.06268
   0.0       0.06993  -0.06880  -0.05300   0.18773
   0.12470   0.05924  -0.04203  -0.11419   0.22123
   0.24852   0.04345  -0.01738  -0.13570   0.16651
   0.37058   0.02634  -0.00368  -0.14519   0.04513
   0.49000   0.00752  -0.01058  -0.18798  -0.20660
71 70 69 70 71
63 60  0 71 73
79 99  0103 83
83102  0102 86
83 96  0100 85
78 86  0 97 87
73 72  0 89 83
69 63  0 81 79
67 60  0 76 75
51 45  0 60 66
70 70 70 71 71

                    WAVE CD =  -0.031180
```

CHART 100



CPSTAR = -0.6993 CLCIR = 0.3650

CL = 0.2870 CD = -0.031180 CMLE = -0.1951 CDF = 0.0 CMC4 = -0.1232

MACH 0.72 AIRFOIL DESIGN -- SAMPLE CASE 1

```
                         X-Y GRID SYSTEM

   2 -0.3872E 01    3 -0.1410E 01    4 -0.6471E 00    5 -0.4900E 00    6 -0.4307E 00    7 -0.3706E 00
   8 -0.3098E 00    9 -0.2485E 00   10 -0.1868E 00   11 -0.1247E 00   12 -0.6240E-01   13  0.0
  14  0.6240E-01   15  0.1247E 00   16  0.1868E 00   17  0.2485E 00   18  0.3098E 00   19  0.3706E 00
  20  0.4307E 00   21  0.4900E 00   22  0.6471E 00   23  0.1410E 01   24  0.3872E 01
   2 -0.9181E 00    3 -0.4261E 00    4 -0.2460E 00    5 -0.1420E 00    6 -0.6592E-01    7 -0.9213E-07
   8  0.6592E-01    9  0.1420E 00   10  0.2460E 00   11  0.4261E 00   12  0.9181E 00

   MACH NO. IS 0.720 ANGLE OF ATTACK IS 0.0    DEGREES
         DIRECT SOLUTION TO     -0.38
                         CASE NUMBER    100

   INVERSE DESIGN CASE
&FINP
M= 0.71999997    ,W=  1.0999998    ,X1=-0.38000000    ,X2= 0.50000000    ,ALP= 0.0             ,EPS= 0.0             ,EPSS=
 0.39999998    ,X4= 0.48999995    ,S4=  2.0000000    ,CONV= 0.99999943E-06,A1= 0.24599999    ,A2= 0.14999998    ,A3=  3.8699999
RN=  20950000.    ,XIBDLY=-0.34999996    ,CIR= 0.18249327    ,CDCORR= 0.0            ,RDEL= 0.25000000    ,RDELFN= 0.12500000    ,
SP= 0.39999969E-02,XSEP= 0.45999998    ,RCPB= 0.19999999    ,CPB= 0.39999998    ,XMON= 0.46999997    ,XLSEP= 0.50000000    ,XPC=
 0.99999964E-01
&END
&IINP
IMAX=          25,JMAX=          13,IKASE=         100,INV=           1,MITER=         400,NHALF=          2,ITACT=          0,ISKP2=
          0,ISKP3=          0,ISKP4=          0,ITERP=          0,IREAD=          0,LP=       1000,ITEUPC=          0,ITELWC=
          0
&END

                         AIRFOIL COORDINATES
     X          YU          YL      UPPER SLOPE LOWER SLOPE
  -0.49000    0.02458   -0.02552    0.93718   -1.31772
  -0.43067    0.04656   -0.06132    0.22258   -0.33090
  -0.37058    0.05667   -0.07542    0.12946   -0.16455
  -0.30982    0.06513   -0.08274    0.08506   -0.06394
  -0.24852    0.06931   -0.08621    0.05419   -0.03119
  -0.18677    0.07183   -0.08675    0.02756    0.01396
  -0.12470    0.07280   -0.08442    0.00332    0.06268
  -0.06240    0.07223   -0.07866   -0.02240    0.12527
   0.0        0.06593   -0.06886   -0.05300    0.18773
   0.06240    0.06557   -0.05540   -0.08685    0.21997
   0.12470    0.05924   -0.04203   -0.11419    0.22123
   0.18677    0.05165   -0.02683   -0.12870    0.20186
   0.24852    0.04345   -0.01736   -0.13570    0.16051
   0.30982    0.03499   -0.00865   -0.14054    0.11509
   0.37058    0.02634   -0.00368   -0.14519    0.04513
   0.43067    0.01737   -0.00366   -0.15402   -0.04915
   0.49000    0.00752   -0.01058   -0.18758   -0.20660

                         UPPER CP INPUT
   -0.912     -1.006     -0.940     -1.015     -1.020     -1.049     -1.081     -0.583
   -0.537     -0.360     -0.275     -0.215     -0.154     -0.100     -0.041

                         LOWER CP INPUT
   -0.690     -0.686     -0.630     -0.620     -0.611     -0.561     -0.305      0.040
    0.212      0.315      0.407      0.481      0.531      0.547      0.570
ITERATION  10 CIR =  0.29997 DPM =  0.00931138 AT 16   2 NSSP =   15 DELTAY OR DELSTAR =  0.0
ITERATION  20 CIR =  0.29601 DPM =  0.00303018 AT  6   2 NSSP =   14 DELTAY OR DELSTAR =  0.0
ITERATION  30 CIR =  0.29985 DPM =  0.00103907 AT 14   3 NSSP =   15 DELTAY OR DELSTAR =  0.0
ITERATION  40 CIR =  0.30067 DPM =  0.00074875 AT 10   2 NSSP =   15 DELTAY OR DELSTAR =  0.0
ITERATION  50 CIR =  0.30111 DPM =  0.00032488 AT 11   3 NSSP =   14 DELTAY OR DELSTAR =  0.0
ITERATION  60 CIR =  0.30190 DPM =  0.00059426 AT 14   3 NSSP =   14 DELTAY OR DELSTAR =  0.0222
ITERATION  70 CIR =  0.30175 DPM =  0.00026259 AT 10   2 NSSP =   14 DELTAY OR DELSTAR =  0.0073
ITERATION  80 CIR =  0.30192 DPM =  0.00020400 AT 15   3 NSSP =   14 DELTAY OR DELSTAR =  0.0037
ITERATION  90 CIR =  0.30191 DPM =  0.00014624 AT 10   3 NSSP =   14 DELTAY OR DELSTAR =  0.0025
ITERATION 100 CIR =  0.30190 DPM =  0.00010759 AT 10   3 NSSP =   14 DELTAY OR DELSTAR =  0.0018
ITERATION 110 CIR =  0.30188 DPM =  0.00007814 AT 11   5 NSSP =   14 DELTAY OR DELSTAR =  0.0013
ITERATION 120 CIR =  0.30188 DPM =  0.00005704 AT 10   5 NSSP =   14 DELTAY OR DELSTAR =  0.0010
ITERATION 130 CIR =  0.30187 DPM =  0.00004121 AT 10   4 NSSP =   14 DELTAY OR DELSTAR =  0.0007
ITERATION 140 CIR =  0.30196 DPM =  0.00002998 AT 11   5 NSSP =   14 DELTAY OR DELSTAR =  0.0005
ITERATION 150 CIR =  0.30180 DPM =  0.00002164 AT  8   3 NSSP =   14 DELTAY OR DELSTAR =  0.0004
ITERATION 160 CIR =  0.30186 DPM =  0.00001615 AT 12   5 NSSP =   14 DELTAY OR DELSTAR =  0.0003
ITERATION 170 CIR =  0.30186 DPM =  0.00001162 AT 10   3 NSSP =   14 DELTAY OR DELSTAR =  0.0002
ITERATION 180 CIR =  0.30185 DPM =  0.00000857 AT 12   3 NSSP =   14 DELTAY OR DELSTAR =  0.0002
ITERATION 190 CIR =  0.30185 DPM =  0.00000650 AT  4   5 NSSP =   14 DELTAY OR DELSTAR =  0.0001
ITERATION 200 CIR =  0.30185 DPM =  0.00000542 AT 22   6 NSSP =   14 DELTAY OR DELSTAR =  0.0001
ITERATION 210 CIR =  0.30185 DPM =  0.00000381 AT 12   3 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 220 CIR =  0.30185 DPM =  0.00000274 AT  9   2 NSSP =   14 DELTAY OR DELSTAR =  0.0001
ITERATION 230 CIR =  0.30185 DPM =  0.00000203 AT 16   2 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 240 CIR =  0.30185 DPM =  0.00000781 AT 23   6 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 250 CIR =  0.30185 DPM =  0.00000179 AT  2  10 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 260 CIR =  0.30185 DPM =  0.00000232 AT  2   8 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 270 CIR =  0.30185 DPM =  0.00000507 AT 24   6 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 280 CIR =  0.30185 DPM =  0.00000253 AT 23   7 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 290 CIR =  0.30185 DPM =  0.00000240 AT 23   7 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 300 CIR =  0.30185 DPM =  0.00000135 AT 23   6 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 310 CIR =  0.30185 DPM =  0.00000198 AT 23   7 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 320 CIR =  0.30185 DPM =  0.00000451 AT 24   6 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 330 CIR =  0.30185 DPM =  0.00000272 AT 22   7 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 340 CIR =  0.30185 DPM =  0.00000425 AT 24   7 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 350 CIR =  0.30185 DPM =  0.00000264 AT 23   7 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 360 CIR =  0.30185 DPM =  0.00000507 AT 24   6 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 370 CIR =  0.30185 DPM =  0.00000296 AT 23   7 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 380 CIR =  0.30185 DPM =  0.00000484 AT 23   7 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 390 CIR =  0.30185 DPM =  0.00000397 AT 23   7 NSSP =   14 DELTAY OR DELSTAR =  0.0000
ITERATION 400 CIR =  0.30185 DPM =  0.00000387 AT 24   7 NSSP =   14 DELTAY OR DELSTAR =  0.0000
```

CP BY BACKWARD DIFFERENCES

X	CPU	CPL
-0.490	-0.268	0.502
-0.431	-1.028	-0.324
-0.371	-0.915	-0.687
-0.310	-1.007	-0.686
-0.249	-0.941	-0.631
-0.187	-1.016	-0.621
-0.125	-1.021	-0.611
-0.062	-1.050	-0.560
0.0	-1.031	-0.304
0.062	-0.584	0.037
0.125	-0.357	0.215
0.187	-0.361	0.314
0.249	-0.276	0.407
0.310	-0.211	0.431
0.371	-0.154	0.532
0.431	-0.130	0.543
0.490	-0.033	0.566
0.647	0.102	0.046
1.410	0.036	0.036
3.872	0.005	0.005

CP BY CENTRAL DIFFERENCES

X	CPU	CPL
-0.490	-0.269	0.502
-0.431	-1.028	-0.324
-0.371	-1.018	-0.533
-0.310	-0.996	-0.605
-0.249	-0.982	-0.616
-0.187	-1.004	-0.613
-0.125	-1.023	-0.586
-0.062	-1.049	-0.479
0.0	-0.905	-0.253
0.062	-0.679	-0.016
0.125	-0.536	0.160
0.187	-0.357	0.294
0.249	-0.258	0.395
0.310	-0.223	0.472
0.371	-0.159	0.522
0.431	-0.092	0.551
0.490	0.148	0.460
0.647	0.086	0.086
1.410	0.026	0.026
3.872	0.004	0.004

CP BY CENTRAL DIFFERENCES

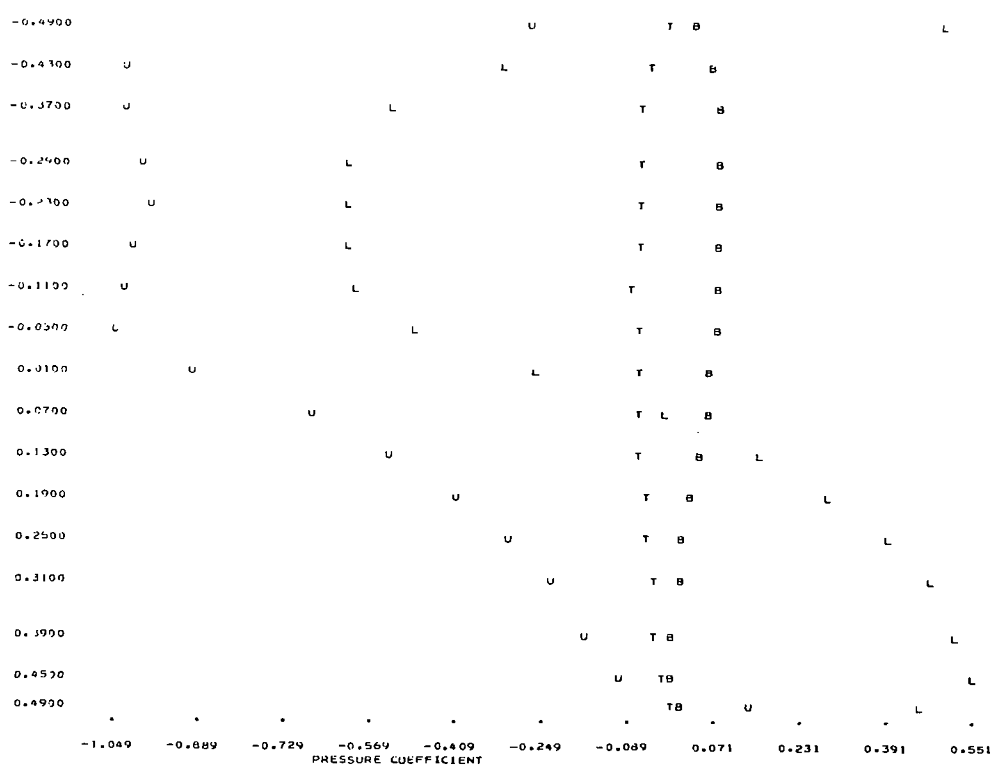
X	CPU	CPL
-0.490	-0.268	0.502
-0.431	-1.028	-0.324
-0.371	-1.018	-0.533
-0.310	-0.996	-0.605
-0.249	-0.982	-0.616
-0.187	-1.004	-0.613
-0.125	-1.023	-0.586
-0.062	-1.049	-0.479
0.0	-0.905	-0.253
0.062	-0.679	-0.016
0.125	-0.536	0.160
0.187	-0.357	0.294
0.249	-0.258	0.395
0.310	-0.223	0.472
0.371	-0.159	0.522
0.431	-0.052	0.551
0.490	0.148	0.460
0.647	0.186	0.086
1.410	0.025	0.026
3.872	0.004	0.004

X	YU	YL	SLU	SLL
-0.49000	0.02458	-0.02552	0.93718	-1.31772
-0.43067	0.04856	-0.06132	0.22258	-0.33099
-0.37058	0.05905	-0.07502	0.12977	-0.16525
-0.30982	0.06553	-0.08247	0.08432	-0.08434
-0.24852	0.06960	-0.08600	0.05405	-0.03140
-0.18677	0.07211	-0.08654	0.02741	0.01387
-0.12470	0.07306	-0.08418	0.00312	0.06265
-0.06240	0.07245	-0.07843	-0.02263	0.12524
0.0	0.07013	-0.06883	-0.05353	0.18729
0.06240	0.06595	-0.05619	-0.07943	0.21943
0.12470	0.05975	-0.04236	-0.11425	0.22128
0.18677	0.05220	-0.02912	-0.12857	0.20241
0.24852	0.04403	-0.01764	-0.13553	0.16710
0.30982	0.03556	-0.00891	-0.14042	0.11546
0.37058	0.02688	-0.00399	-0.14526	0.04496
0.43067	0.01790	-0.00414	-0.15427	-0.05072
0.49000	0.00815	-0.01211	-0.18591	-0.20427

71	71	71	71	71	71	71	71	71	71	71
70	69	69	69	69	69	69	70	70	70	71
69	67	65	63	61	58	64	67	69	72	74
70	69	53	55	50	0	77	76	80	84	83
71	72	73	76	63	0	100	94	88	84	80
71	74	78	85	0	0	112	103	95	88	81
71	76	82	89	0	0	112	106	99	91	82
71	77	84	91	0	0	0	107	102	94	82
71	78	85	91	0	0	0	109	103	95	83
71	77	83	90	0	0	0	110	104	96	83
71	76	80	85	0	0	0	113	104	95	83
71	74	76	79	0	0	0	104	94	92	82
70	72	72	72	72	0	0	96	93	89	82
70	70	69	67	66	0	93	91	89	86	81
70	66	66	63	61	0	87	86	86	84	80
69	67	64	61	58	0	83	93	83	82	80
69	66	62	59	55	0	80	80	80	80	79
69	65	61	58	54	0	78	73	73	74	76
69	65	62	58	54	0	75	75	76	77	77
67	61	57	54	53	0	69	71	74	77	79
66	67	67	67	67	68	69	70	71	72	74
70	70	70	70	70	70	71	71	71	71	72
71	71	71	71	71	71	71	71	71	71	72

WAVE CD = -0.003593

CHART 100



CPSTAR = -0.6993 CLCIR = 0.6037

CL = 0.5748 CD = -0.003593 CMLE = -0.2991 CDF = 0.0 CMC4 = -0.1554

```
                    X-Y GRID SYSTEM

  2 -0.8027E 01    3 -0.3872E 01    4 -0.2276E 01    5 -0.1410E 01    6 -0.9115E 00    7 -0.6471E 00
  8 -0.5317E 00    9 -0.4900E 00   10 -0.4604E 00   11 -0.4307E 00   12 -0.4007E 00   13 -0.3706E 00
 14 -0.3403E 00   15 -0.3098E 00   16 -0.2792E 00   17 -0.2485E 00   18 -0.2177E 00   19 -0.1868E 00
 20 -0.1558E 00   21 -0.1247E 00   22 -0.9357E-01   23 -0.6240E-01   24 -0.3121E-01   25  0.0
 26  0.3121E-01   27  0.6240E-01   28  0.9357E-01   29  0.1247E 00   30  0.1558E 00   31  0.1868E 00
 32  0.2177E 00   33  0.2485E 00   34  0.2792E 00   35  0.3098E 00   36  0.3403E 00   37  0.3706E 00
 38  0.4007E 00   39  0.4307E 00   40  0.4604E 00   41  0.4900E 00   42  0.5317E 00   43  0.6471E 00
 44  0.9115E 00   45  0.1410E 01   46  0.2276E 01   47  0.3872E 01   48  0.8027E 01
  2 -0.1869E 01    3 -0.9181E 00    4 -0.5939E 00    5 -0.4261E 00    6 -0.3206E 00    7 -0.2460E 00
  8 -0.1888E 00    9 -0.1420E 00   10 -0.1019E 00   11 -0.6592E-01   12 -0.3239E-01   13 -0.9213E-07
 14  0.3239E-01   15  0.6592E-01   16  0.1019E 00   17  0.1420E 00   18  0.1888E 00   19  0.2460E 00
 20  0.3206E 00   21  0.4261E 00   22  0.5939E 00   23  0.9181E 00   24  0.1869E 01

  MACH NO. IS 0.720 ANGLE OF ATTACK IS 0.0   DEGREES
        DIRECT SOLUTION TO    -0.38
                     CASE NUMBER   100

   INVERSE DESIGN CASE
&FINP
M= 0.71999997    ,W=  1.6999998    ,X1=-0.38000000    ,X2= 0.50000000    ,ALP= 0.0             ,EPS= 0.0             ,EPSS=
 0.39999998    ,X4= 0.48999995    ,S4=  2.0000000    ,CONV= 0.99999943E-06,A1= 0.24599999    ,A2= 0.14999998    ,A3=  3.8699999
RN=  29950000.    ,XIBDLY=-0.34999996    ,CIR= 0.30185080    ,CDCORR= 0.0              ,RDEL= 0.25000000    ,RDELFN= 0.12500000    ,
SP= 0.39999969E-02,XSEP= 0.45999998    ,RCPB= 0.19999999    ,CPB= 0.39999998    ,XMUN= 0.46999997    ,XLSEP= 0.50000000    ,XPC=
 0.99999964E-01
&END
&IINP
IMAX=          49,JMAX=          25,IKASE=         100,INV=           1,MITER=         400,NHALF=           2,ITACT=           0,ISKP2=
        0,ISKP3=          0,ISKP4=          0,ITERP=          0,IREAD=          0,LP=         1000,ITEUPC=          0,ITELWC=
        0
&END

                    AIRFOIL COORDINATES
    X         YU         YL      UPPER SLOPE LOWER SLOPE
 -0.49000   0.02458   -0.02552   0.93718   -1.31772
 -0.46044   0.04034   -0.04897   0.34842   -0.52528
 -0.43067   0.04856   -0.06132   0.22258   -0.33099
 -0.40072   0.05427   -0.06956   0.16531   -0.22929

                    UPPER CP INPUT
   -1.107    -1.101    -1.098    -1.067    -1.076    -1.078    -1.069    -1.061
   -1.067    -1.056    -1.055    -1.017    -0.917    -0.808    -0.701    -0.608
   -0.524    -0.438    -0.379    -0.326    -0.284    -0.244    -0.207    -0.162
   -0.144    -0.114    -0.087    -0.055    -0.007

                    LOWER CP INPUT
   -0.673    -0.677    -0.665    -0.657    -0.649    -0.637    -0.650    -0.628
   -0.621    -0.590    -0.528    -0.408    -0.267    -0.139     0.002     0.099
    0.167     0.241     0.292     0.357     0.396     0.436     0.470     0.498
    0.516     0.528     0.531     0.510     0.532
ITERATION  10 CIR =  0.30264 DPM =  0.00127091 AT 39 17 NSSP =   67 DELTAY OR DELSTAR =  0.0
ITERATION  20 CIR =  0.33161 DPM =  0.00018459 AT 42 13 NSSP =   68 DELTAY OR DELSTAR =  0.0
ITERATION  30 CIR =  0.30163 DPM =  0.00005847 AT 35  2 NSSP =   67 DELTAY OR DELSTAR =  0.0
ITERATION  40 CIR =  0.30165 DPM =  0.00004399 AT 35  2 NSSP =   67 DELTAY OR DELSTAR =  0.0
ITERATION  50 CIR =  0.30163 DPM =  0.00003612 AT 36  2 NSSP =   67 DELTAY OR DELSTAR =  0.0

ITERATION  60 CIR =  0.30113 DPM =  0.00007552 AT 46  2 NSSP =   68 DELTAY OR DELSTAR =  0.0036
ITERATION  70 CIR =  0.30094 DPM =  0.00002682 AT 27  2 NSSP =   67 DELTAY OR DELSTAR =  0.0018
ITERATION  80 CIR =  0.30094 DPM =  0.00003499 AT 47 12 NSSP =   67 DELTAY OR DELSTAR =  0.0004
ITERATION  90 CIR =  0.30094 DPM =  0.00002229 AT 21  2 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 100 CIR =  0.30094 DPM =  0.00006038 AT 48 11 NSSP =   67 DELTAY OR DELSTAR =  0.0000
ITERATION 110 CIR =  0.30094 DPM =  0.00002879 AT 49 13 NSSP =   67 DELTAY OR DELSTAR =  0.0000
ITERATION 120 CIR =  0.30094 DPM =  0.00002983 AT 48 13 NSSP =   67 DELTAY OR DELSTAR =  0.0000
ITERATION 130 CIR =  0.30094 DPM =  0.00002205 AT 47  7 NSSP =   67 DELTAY OR DELSTAR =  0.0000
ITERATION 140 CIR =  0.30094 DPM =  0.00001734 AT 21  2 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 150 CIR =  0.30094 DPM =  0.00002346 AT  2  7 NSSP =   67 DELTAY OR DELSTAR =  0.0000
ITERATION 160 CIR =  0.30094 DPM =  0.00002998 AT 48  7 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 170 CIR =  0.30094 DPM =  0.00002834 AT 48 13 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 180 CIR =  0.30094 DPM =  0.00002784 AT 48 12 NSSP =   67 DELTAY OR DELSTAR =  0.0000
ITERATION 190 CIR =  0.30094 DPM =  0.00002900 AT 48 13 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 200 CIR =  0.30094 DPM =  0.00002027 AT 44 12 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 210 CIR =  0.30094 DPM =  0.00002140 AT 46 11 NSSP =   67 DELTAY OR DELSTAR =  0.0000
ITERATION 220 CIR =  0.30094 DPM =  0.00002801 AT 47 12 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 230 CIR =  0.30094 DPM =  0.00003087 AT 48 13 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 240 CIR =  0.30094 DPM =  0.00002354 AT  2  7 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 250 CIR =  0.30094 DPM =  0.00002956 AT 48  8 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 260 CIR =  0.30094 DPM =  0.00001323 AT 44  3 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 270 CIR =  0.30094 DPM =  0.00002342 AT 46  9 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 280 CIR =  0.30094 DPM =  0.00003570 AT 47 12 NSSP =   67 DELTAY OR DELSTAR =  0.0000
ITERATION 290 CIR =  0.30094 DPM =  0.00001252 AT 47 13 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 300 CIR =  0.30094 DPM =  0.00003431 AT 48 13 NSSP =   67 DELTAY OR DELSTAR =  0.0000
ITERATION 310 CIR =  0.30094 DPM =  0.00002944 AT 48  7 NSSP =   67 DELTAY OR DELSTAR =  0.0000
ITERATION 320 CIR =  0.30094 DPM =  0.00001991 AT 47 12 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 330 CIR =  0.30094 DPM =  0.00003204 AT 48 13 NSSP =   67 DELTAY OR DELSTAR =  0.0000
ITERATION 340 CIR =  0.30094 DPM =  0.00003159 AT 48 12 NSSP =   67 DELTAY OR DELSTAR =  0.0000
ITERATION 350 CIR =  0.30094 DPM =  0.00001216 AT 46  4 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 360 CIR =  0.30094 DPM =  0.00005831 AT 48 13 NSSP =   67 DELTAY OR DELSTAR =  0.0000
ITERATION 370 CIR =  0.30094 DPM =  0.00001198 AT 47  9 NSSP =   67 DELTAY OR DELSTAR =  0.0001
ITERATION 380 CIR =  0.30094 DPM =  0.00001224 AT 43 13 NSSP =   67 DELTAY OR DELSTAR =  0.0000
ITERATION 390 CIR =  0.30094 DPM =  0.00002964 AT 48 13 NSSP =   67 DELTAY OR DELSTAR =  0.0000
ITERATION 400 CIR =  0.30094 DPM =  0.00001361 AT 46 13 NSSP =   67 DELTAY OR DELSTAR =  0.0001
```

CP BY BACKWARD DIFFERENCES

X	CPU	CPL
-0.490	-0.561	0.504
-0.460	-0.979	-0.111
-0.431	-1.278	-0.631
-0.401	-1.119	-0.584
-0.371	-1.108	-0.672
-0.340	-1.102	-0.677
-0.310	-1.099	-0.665
-0.279	-1.068	-0.658
-0.249	-1.077	-0.650
-0.218	-1.079	-0.638
-0.187	-1.070	-0.651
-0.156	-1.062	-0.629
-0.125	-1.068	-0.622
-0.094	-1.057	-0.590
-0.062	-1.056	-0.528
-0.031	-1.018	-0.408
0.0	-0.918	-0.267
0.031	-0.809	-0.139
0.062	-0.702	0.001
0.094	-0.609	0.088
0.125	-0.525	0.167
0.156	-0.439	0.241
0.187	-0.379	0.292
0.218	-0.326	0.357
0.249	-0.284	0.396
0.279	-0.244	0.436
0.310	-0.207	0.470
0.340	-0.162	0.498
0.371	-0.144	0.517
0.401	-0.114	0.529
0.431	-0.087	0.532
0.460	-0.055	0.511
0.490	-0.003	0.527
0.532	0.175	0.046
0.647	0.117	0.117
0.912	0.064	0.064
1.410	0.025	0.025
2.276	0.008	0.008
3.872	0.003	0.003
6.027	0.001	0.001

CP BY CENTRAL DIFFERENCES

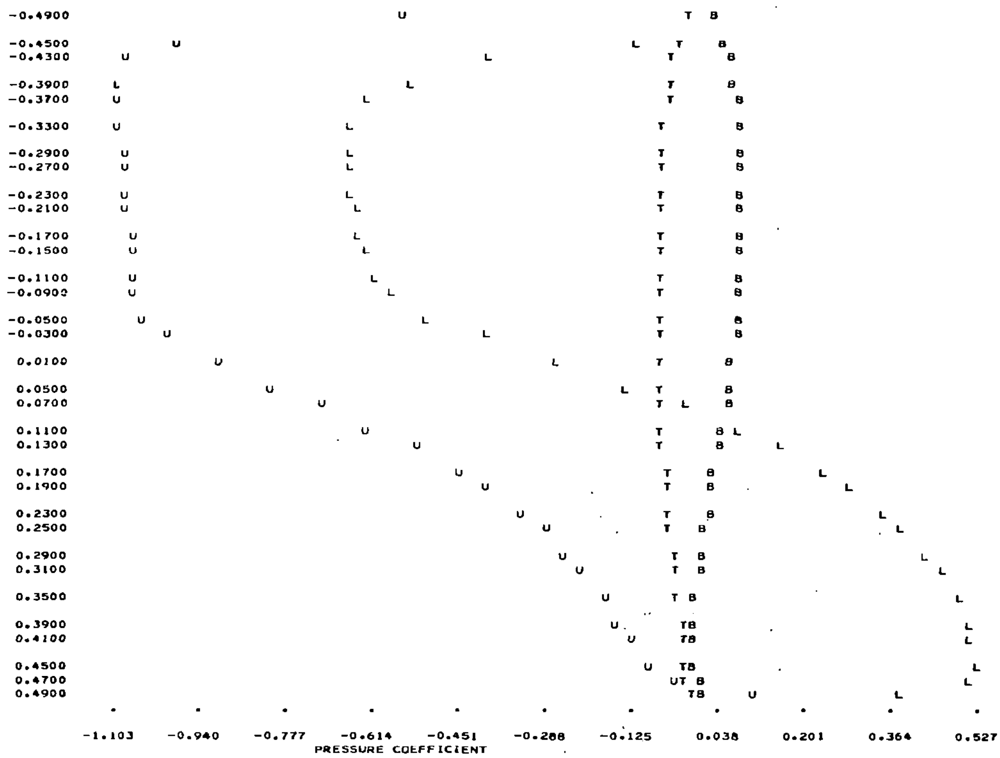
X	CPU	CPL
-0.490	-0.561	0.504
-0.460	-0.979	-0.111
-0.431	-1.081	-0.400
-0.401	-1.097	-0.535
-0.371	-1.103	-0.619
-0.340	-1.099	-0.649
-0.310	-1.066	-0.655
-0.279	-1.076	-0.653
-0.249	-1.076	-0.647
-0.218	-1.074	-0.645
-0.187	-1.068	-0.640
-0.156	-1.066	-0.629
-0.125	-1.063	-0.611
-0.094	-1.058	-0.573
-0.062	-1.043	-0.501
-0.031	-0.993	-0.393
0.0	-0.905	-0.262
0.031	-0.805	-0.135
0.062	-0.708	-0.023
0.094	-0.615	0.076
0.125	-0.527	0.161
0.156	-0.450	0.256
0.187	-0.386	0.297
0.218	-0.333	0.348
0.249	-0.287	0.394
0.279	-0.246	0.434
0.310	-0.206	0.468
0.340	-0.172	0.495
0.371	-0.143	0.514
0.401	-0.115	0.527
0.431	-0.084	0.527
0.460	-0.041	0.520
0.490	0.104	0.387
0.532	0.179	0.179
0.647	0.105	0.105
0.912	0.053	0.053
1.410	0.023	0.023
2.276	0.009	0.009
3.872	0.003	0.003
6.027	0.001	0.001

X	YU	YL	SLU	SLL
-0.49000	0.02456	-0.02552	0.93718	-1.31772
-0.46044	0.04334	-0.06897	0.34942	-0.42524
-0.43067	0.04390	-0.00132	0.22256	-0.13099
-0.40072	0.05427	-0.06409	0.16531	-0.22929
-0.37058	0.05871	-0.07543	0.13068	-0.16452
-0.34028	0.07727	-0.07967	0.10486	-0.11838
-0.30982	0.06316	-0.08270	0.08535	-0.08371
-0.27923	0.06759	-0.06419	0.06899	-0.05543
-0.24852	0.06939	-0.09621	0.05454	-0.03089
-0.21769	0.07135	-0.08642	0.04091	-0.00677
-0.18677	0.07141	-0.08674	0.02807	0.01440
-0.15577	0.07261	-0.08593	0.01664	0.03747
-0.12470	0.07293	-0.08437	0.00408	0.06318
-0.09357	0.07247	-0.08190	-0.00777	0.09243
-0.06240	0.07244	-0.07857	-0.02097	0.12590
-0.03121	0.07194	-0.07415	-0.03342	0.15919
0.0	0.07133	-0.06874	-0.04716	0.18829
0.03121	0.06953	-0.06236	-0.06837	0.20890
0.06240	0.06611	-0.05536	-0.06716	0.22069
0.09357	0.06302	-0.04871	-0.10408	0.22437
0.12470	0.05964	-0.04136	-0.11738	0.22230
0.15577	0.05563	-0.03517	-0.12573	0.21441
0.18677	0.05140	-0.02872	-0.13151	0.20278
0.21769	0.04771	-0.02208	-0.13531	0.18710
0.24852	0.04351	-0.01726	-0.13632	0.16773
0.27923	0.03922	-0.01240	-0.14080	0.14427
0.30982	0.03468	-0.00841	-0.14315	0.11639
0.34028	0.03050	-0.00530	-0.14571	0.08341
0.37058	0.02605	-0.00339	-0.14799	0.04627
0.40072	0.02154	-0.00283	-0.15140	0.00360
0.43067	0.01692	-0.00329	-0.15745	-0.04764
0.46044	0.01208	-0.00504	-0.16882	-0.11002
0.49000	0.00682	-0.01047	-0.19413	-0.20120

```
71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 72
71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71
71 70 70 70 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 72
70 70 69 69 65 67 69 64 65 64 65 69 67 65 69 70 70 70 70 70 71 72 73
70 69 68 67 67 66 66 66 66 66 66 66 66 66 67 67 68 63 69 70 71 73 74
70 69 58 67 65 63 62 60 59 57 56 56 56 60 62 64 66 68 70 72 74 76 75
70 70 70 68 66 64 60 57 52 47 38 31 51 59 64 68 71 71 76 77 78 78 76
70 70 71 70 69 68 66 63 59 54 50  0 40 82 81 81 81 82 82 82 81 80 76
70 71 71 72 72 71 71 71 71 73  0  0  0101 94 89 87 85 84 83 81 79 76
70 71 72 73 73 74 75 77 80 66  0  0  0113102 97 92 89 87 85 83 80 76
70 71 72 74 75 77 79 83 87  0  0  0  3116108102 97 93 90 87 84 80 76
70 71 73 75 77 80 83 87 92  0  0  0  0116111106101 97 93 89 85 81 76
70 71 73 76 79 82 86 90 94  0  0  0  0117113109104100 95 91 86 81 76
70 71 74 77 80 84 87 91 95  0  0  0  0116113110106102 98 93 88 82 76
70 72 74 78 81 85 89 92 96  0  0  0  0  0114111108104 99 94 89 82 76
70 72 74 78 82 86 89 93 96  0  0  0  0  0114111106105101 96 89 83 76
70 72 75 78 82 86 89 93 96  0  0  0  0  0114112109106102 97 90 83 76
70 72 75 78 82 86 89 93 96  0  0  0  0  0114112109106102 97 91 83 76
70 72 74 78 82 85 89 92 95  0  0  0  0  0114112109106102 97 91 83 76
70 72 74 78 81 84 87 91 94  0  0  0  0  0114112109106102 97 91 83 76
70 71 74 77 80 83 86 89 92  0  0  0  0  0113111109105101 96 90 83 76
70 71 74 76 79 81 84 86 89  0  0  0  0  0113110108104100 95 90 83 76
70 71 73 75 77 79 81 83 85  0  0  0  0  3110108105102 98 94 89 83 76
70 71 73 74 76 77 78 79 81  0  0  0  0  0106104101 98 95 92 88 83 76
70 71 72 73 74 75 75 76 76 77  0  0  0  3102 99 97 95 93 91 87 82 76
70 71 71 72 72 73 73 73 73 73  0  0  0  0 98 96 95 93 91 89 86 82 76
70 70 71 71 71 71 70 70 69 69  0  0  0 96 95 93 92 91 89 88 85 82 76
70 70 70 73 70 69 68 67 67 66  0  0  0 92 91 91 90 89 88 86 84 81 76
70 70 70 69 68 67 66 65 64 63  0  0  0 89 89 88 88 87 86 85 83 81 76
70 70 69 68 67 66 66 63 62 61 60  0  0 87 86 86 86 85 85 84 83 80 76
70 70 69 68 66 65 63 62 61 59 58  0  0 85 84 84 84 84 83 83 82 80 76
70 69 68 67 65 64 62 61 59 58 56  0  0 83 83 83 83 82 82 82 81 79 76
70 69 68 66 65 63 61 60 58 56 55  0  0 81 81 81 81 81 81 81 80 79 76
70 69 68 66 64 62 61 59 57 55 53  0  0 80 80 80 80 80 80 80 79 79 76
70 69 67 66 64 62 60 58 56 55 53  0 78 78 78 78 79 79 79 79 79 78 76
70 69 67 65 64 62 60 58 56 54 52  0 77 77 77 77 77 78 78 78 78 78 75
70 69 67 65 63 62 60 58 56 54 52  0 76 76 76 76 76 76 77 77 77 77 75
70 69 67 65 64 62 60 59 57 55 52  0 75 74 75 75 75 75 76 76 77 77 75
70 69 67 65 64 62 61 59 58 56 53  0 73 73 73 73 74 75 75 76 76 77 75
70 68 66 64 63 61 60 59 58 57 56  0 69 70 71 72 73 74 75 76 77 77 76
70 68 66 65 64 63 62 62 62 62 63 64 66 68 69 70 71 72 73 74 75 76 75
70 69 67 67 66 66 66 66 66 66 67 67 68 68 69 70 70 71 72 73 74 75 75
70 69 69 69 65 69 69 69 69 69 69 69 70 70 70 70 70 71 71 72 72 73 74
70 70 70 70 70 70 70 70 70 70 71 71 71 71 71 71 71 71 71 71 72 72 73
71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 72 72
71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 72
71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 71 72 72 72
```

WAVE CD = -0,016110

CHART 100



CPSTAR = -0.6993 CLCIR = 0.6019

CL = 0.5875 CD = -0.016110 CMLE = -0.2989 CDF = 0.0 CMC4 = -0.1520

BOUNDARY LAYER ANALYSIS FOR REYNOLDS NO. OF 0.210E 08

X	Y	YNEW	P	DELS	THETA	SEP	H	PI	TAU		
-0.34028	0.06227	0.06226	1.17571	0.00001	0.00001	0.00000	0.98663	0.00612		12	0.00186
-0.30982	0.06516	0.06503	1.16937	0.00014	0.00007	0.00001	1.94562	0.01163		13	0.00155
-0.27923	0.06750	0.06727	1.16482	0.00023	0.00012	0.00000	1.90204	-0.00089		12	0.00143
-0.24852	0.06939	0.06909	1.16504	0.00031	0.00016	0.00000	1.88197	0.00562		11	0.00136
-0.21769	0.07045	0.07047	1.16403	0.00038	0.00020	0.00001	1.87019	0.01942		11	0.00130
-0.18677	0.07191	0.07146	1.16128	0.00046	0.00025	0.00001	1.85981	0.01085		11	0.00126
-0.15577	0.07261	0.07208	1.16002	0.00053	0.00029	0.00001	1.85084	0.01224		10	0.00123
-0.12470	0.07293	0.07233	1.15879	0.00060	0.00032	0.00001	1.84466	0.02632		10	0.00120
-0.09357	0.07287	0.07220	1.15651	0.00067	0.00036	0.00004	1.84147	0.09002		10	0.00117
-0.06240	0.07243	0.07168	1.14975	0.00075	0.00040	0.00014	1.84754	0.37327		9	0.00110
-0.03121	0.07158	0.07073	1.12683	0.00085	0.00046	0.00033	1.85997	0.81990		10	0.00101
0.0	0.07033	0.06935	1.08779	0.00098	0.00052	0.00052	1.86034	1.16956		12	0.00095
0.03121	0.06853	0.06741	1.04429	0.00111	0.00061	0.00064	1.84241	1.36936		12	0.00092
0.06240	0.06611	0.06484	1.00333	0.00126	0.00069	0.00073	1.81897	1.57480		12	0.00088
0.09357	0.06309	0.06166	0.96512	0.00142	0.00079	0.00082	1.80016	1.82663		13	0.00085
0.12470	0.05953	0.05802	0.92913	0.00159	0.00090	0.00087	1.77761	1.84233		13	0.00084
0.15577	0.05535	0.05408	0.89825	0.00175	0.00100	0.00086	1.74630	1.72344		12	0.00084
0.18677	0.05146	0.04995	0.87277	0.00190	0.00111	0.00062	1.71513	1.60013		12	0.00084
0.21769	0.04773	0.04568	0.85150	0.00203	0.00120	0.00077	1.68841	1.49107		12	0.00085
0.24852	0.04351	0.04132	0.83348	0.00217	0.00130	0.00075	1.66864	1.47861		12	0.00084
0.27923	0.03922	0.03688	0.81730	0.00232	0.00140	0.00078	1.65934	1.61199		12	0.00083
0.30982	0.03488	0.03239	0.80160	0.00247	0.00150	0.00079	1.64865	1.52716		11	0.00083
0.34028	0.03050	0.02788	0.78786	0.00259	0.00159	0.00073	1.62729	1.31734		11	0.00084
0.37058	0.02605	0.02330	0.77658	0.00272	0.00168	0.00071	1.61511	1.44769		11	0.00083
0.40072	0.02154	0.01863	0.76533	0.00288	0.00178	0.00080	1.62124	1.75212		11	0.00080
0.43067	0.01672	0.01375	0.75330	0.00313	0.00188	0.00109	1.66644	3.26876		12	0.00069
0.46044	0.01238	0.00698	0.73642	0.00502	0.00202	0.00373	2.47992	**********		25	0.00000
0.49000	0.00692	-0.00288	0.67834	0.00953	0.00290	0.00851	3.28981	**********		25	0.00000

X	YOLD	YNEW	DELSTAR
-0.46044	0.04034	0.04034	0.0
-0.43067	0.04856	0.04856	0.0
-0.40072	0.05427	0.05427	0.00000
-0.37058	0.05871	0.05870	0.00001
-0.34028	0.06227	0.06222	0.00005
-0.30982	0.06516	0.06503	0.00013
-0.27923	0.06750	0.06727	0.00022
-0.24852	0.06939	0.06909	0.00030
-0.21769	0.07045	0.07047	0.00038
-0.18677	0.07191	0.07146	0.00046
-0.15577	0.07261	0.07208	0.00053
-0.12470	0.07293	0.07233	0.00060
-0.09357	0.07287	0.07220	0.00067
-0.06240	0.07243	0.07167	0.00076
-0.03121	0.07158	0.07072	0.00086
0.0	0.07033	0.06934	0.00098
0.03121	0.06853	0.06741	0.00112
0.06240	0.06611	0.06483	0.00127
0.09357	0.06307	0.06165	0.00143
0.12470	0.05963	0.05803	0.00159
0.15577	0.05535	0.05409	0.00174
0.18677	0.05146	0.04995	0.00189
0.21769	0.04773	0.04568	0.00203
0.24852	0.04351	0.04131	0.00217
0.27923	0.03922	0.03688	0.00232
0.30982	0.03488	0.03240	0.00246
0.34028	0.03050	0.02788	0.00259
0.37058	0.02605	0.02326	0.00274
0.40072	0.02154	0.01849	0.00302
0.43067	[illegible]	[illegible]	0.01301
0.46044	[illegible]	[illegible]	0.00611
0.49000	[illegible]	[illegible]	0.00829

BOUNDARY LAYER ANALYSIS FOR REYNOLDS NO. OF 0.210E 08

X	Y	YNEW	M	DELS	THETA	SEP	H	PI	TAU	
-0.34026	-0.07969	-0.07968	0.97886	0.00001	0.00001	-0.00000	0.90427	-0.00294	12	0.00192
-0.30982	-0.08276	-0.08263	0.98148	0.00013	0.00007	-0.00000	1.78243	0.00260	13	0.00160
-0.27923	-0.08489	-0.08468	0.98060	0.00021	0.00012	0.00001	1.74496	0.01190	12	0.00148
-0.24852	-0.08621	-0.08592	0.97808	0.00029	0.00017	0.00001	1.72590	0.00574	11	0.00140
-0.21769	-0.08682	-0.08646	0.97719	0.00036	0.00021	0.00001	1.71315	0.01401	11	0.00134
-0.18677	-0.08674	-0.08631	0.97549	0.00043	0.00025	0.00002	1.70517	0.04544	10	0.00130
-0.15577	-0.08593	-0.08543	0.97098	0.00050	0.00030	0.00005	1.69911	0.08937	10	0.00125
-0.12470	-0.08437	-0.08379	0.96359	0.00058	0.00034	0.00011	1.69683	0.23219	8	0.00119
-0.09357	-0.08196	-0.08129	0.94788	0.00067	0.00039	0.00026	1.70275	0.35666	10	0.00111
-0.06240	-0.07857	-0.07778	0.91869	0.00079	0.00046	0.00051	1.71805	1.14462	12	0.00101
-0.03121	-0.07415	-0.07317	0.87553	0.00096	0.00055	0.00085	1.74244	2.06364	14	0.00089
0.0	-0.06874	-0.06752	0.82354	0.00120	0.00068	0.00119	1.76345	2.94353	14	0.00080
0.03121	-0.06258	-0.06100	0.77351	0.00149	0.00084	0.00144	1.77328	3.70564	15	0.00073
0.06240	-0.05588	-0.05401	0.72909	0.00182	0.00102	0.00166	1.78474	4.80806	15	0.00066
0.09357	-0.04893	-0.04066	0.68969	0.00221	0.00123	0.00187	1.79958	5.81726	15	0.00061
0.12470	-0.04196	-0.03924	0.65553	0.00266	0.00146	0.00208	1.82415	7.49914	16	0.00054
0.15577	-0.03516	-0.03194	0.62494	0.00315	0.00172	0.00222	1.83561	7.80788	15	0.00053
0.18677	-0.02872	-0.02502	0.59963	0.00362	0.00198	0.00229	1.83148	8.56462	15	0.00050
0.21769	-0.02268	-0.01845	0.57777	0.00415	0.00225	0.00240	1.84576	9.98870	16	0.00046
0.24852	-0.01720	-0.01240	0.55824	0.00474	0.00254	0.00253	1.86721	11.32496	17	0.00043
0.27923	-0.01240	-0.00704	0.54025	0.00531	0.00284	0.00258	1.87079	11.39692	16	0.00042
0.30982	-0.00841	-0.00264	0.52580	0.00573	0.00314	0.00247	1.82602	8.86838	12	0.00048
0.34028	-0.00536	0.00054	0.51372	0.00588	0.00340	0.00212	1.72612	5.45317	10	0.00058
0.37058	-0.00339	0.00242	0.50498	0.00580	0.00362	0.00159	1.60312	2.81727	11	0.00070
0.40072	-0.00263	0.00283	0.49946	0.00546	0.00377	0.00067	1.45314	0.09675	14	0.00094
0.43067	-0.00329	0.00109	0.49920	0.00497	0.00380	-0.00037	1.30921	-0.88923	13	0.00113
0.46044	-0.00564	-0.00104	0.50252	0.00457	0.00376	-0.00611	1.21603	-1.50000	12	0.00143
0.49000	-0.01047	-0.00723	0.56114	0.00316	0.00281	-0.01678	1.12255	-1.50000	12	0.00143

X	YOLD	YNEW	DELSTAR
-0.46044	-0.04897	-0.04897	0.0
-0.43067	-0.06132	-0.06132	0.0
-0.40072	-0.06936	-0.06956	0.00000
-0.37058	-0.07543	-0.07542	0.00001
-0.34028	-0.07969	-0.07964	0.00005
-0.30982	-0.08276	-0.08264	0.00012
-0.27923	-0.08489	-0.08466	0.00021
-0.24852	-0.08621	-0.08593	0.00029
-0.21769	-0.08682	-0.08646	0.00036
-0.18677	-0.08674	-0.08630	0.00043
-0.15577	-0.08593	-0.08543	0.00051
-0.12470	-0.08437	-0.08379	0.00059
-0.09357	-0.08196	-0.08127	0.00069
-0.06240	-0.07857	-0.07775	0.00082
-0.03121	-0.07415	-0.07314	0.00099
0.0	-0.06874	-0.06750	0.00123
0.03121	-0.06258	-0.06104	0.00151
0.06240	-0.05536	-0.05398	0.00185
0.09357	-0.04872	-0.04663	0.00224
0.12470	-0.04196	-0.03922	0.00268
0.15577	-0.03516	-0.03194	0.00315
0.18677	-0.02872	-0.02500	0.00365
0.21769	-0.02268	-0.01843	0.00418
0.24852	-0.01720	-0.01241	0.00473
0.27923	-0.01240	-0.00711	0.00523
0.30982	-0.00841	-0.00277	0.00560
0.34028	-0.00536	0.00042	0.00576
0.37058	-0.00339	0.00230	0.00568
0.40072	-0.00263	0.00270	0.00539
0.43067	-0.00229	0.00165	0.00493
0.46044	-0.00564	-0.00142	0.00420
0.49000	-0.01047	-0.00694	0.00346

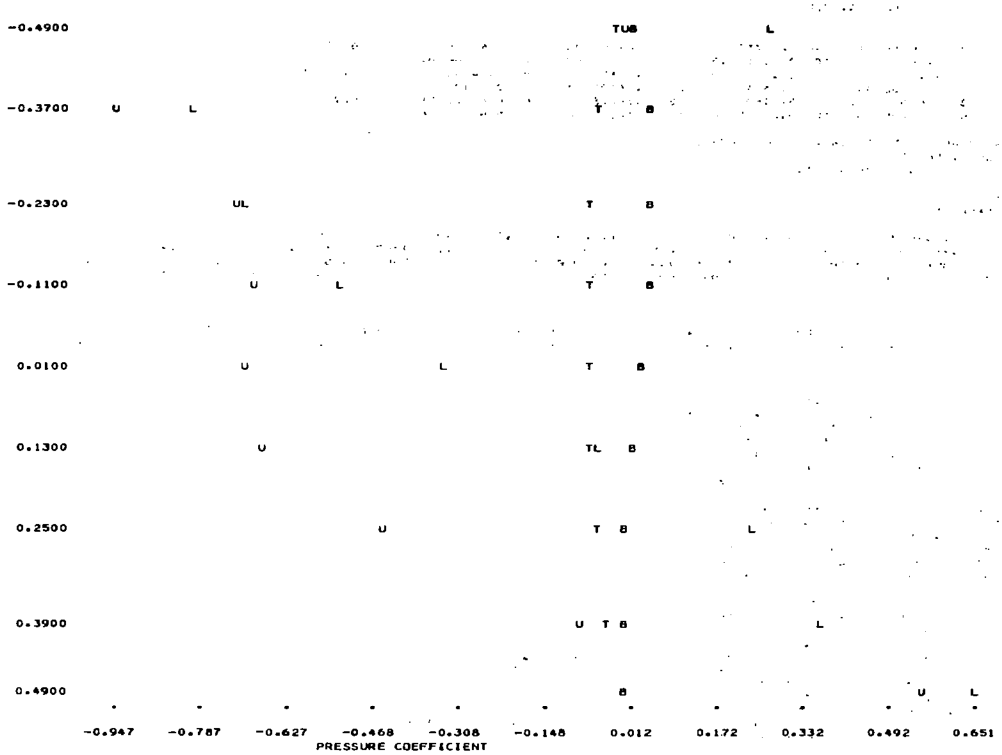
COF = 0.767483
NO UPPER SURFACE SEPARATION BEFORE 0.46000

Sample Case No. 2 - Airfoil Analysis with Viscous Interaction

```
SAMPLE CASE NO. 2--KORN AIRFOIL 75-06-12  TWO GRID HALVINGS ONLY
&FINP  M=0.752,RN=20.95E+06,&END
&IINP ITACT=1,&END
   61
0.0          0.00074     0.00003     0.00399     0.00056     0.00717     0.00171     0.01015
0.00365      0.01296     0.00640     0.01573     0.00993     0.01852     0.01415     0.02133
0.01906      0.02415     0.02461     0.02696     0.03080     0.02973     0.03761     0.03245
0.04503      0.03510     0.05304     0.03766     0.06163     0.04009     0.07080     0.04237
0.08057      0.04444     0.09101     0.04629     0.10215     0.04800     0.11398     0.04963
0.12642      0.05118     0.13946     0.05266     0.15306     0.05408     0.16720     0.05543
0.18183      0.05670     0.19694     0.05791     0.21251     0.05903     0.22850     0.06007
0.24488      0.06104     0.27873     0.06271     0.31385     0.06403     0.35000     0.06499
0.38695      0.06556     0.42446     0.06575     0.46229     0.06553     0.50019     0.06489
0.53793      0.06379     0.57526     0.06219     0.61198     0.06003     0.64797     0.05727
0.68302      0.05390     0.71710     0.04996     0.75007     0.04552     0.78179     0.04069
0.81214      0.03558     0.84096     0.03032     0.86810     0.02508     0.89338     0.02003
0.91660      0.01533     0.93749     0.01114     0.95580     0.00759     0.96390     0.00607
0.97125      0.00472     0.97783     0.00355     0.98360     0.00256     0.98854     0.00175
0.99262      0.00109     0.99582     0.00060     0.99813     0.00026     0.99953     0.00006
1.00000      0.00000
0.0          1.0          0.0         0.0
   51
0.00000      0.00074     0.00049    -0.00240     0.00165    -0.00538     0.00351    -0.00827
0.00605     -0.01105     0.00932    -0.01370     0.01336    -0.01626     0.01814    -0.01877
0.02366     -0.02122     0.02987    -0.02364     0.03676    -0.02602     0.04431    -0.02834
0.05250     -0.03061     0.07074    -0.03491     0.09134    -0.03886     0.11420    -0.04237
0.13920     -0.04538     0.16622    -0.04789     0.19512    -0.04978     0.22574    -0.05112
0.25793     -0.05186     0.27455    -0.05200     0.30873    -0.05181     0.34401    -0.05097
0.38020     -0.04946     0.41714    -0.04726     0.45467    -0.04436     0.49263    -0.04077
0.53091     -0.03653     0.56937    -0.03168     0.60792    -0.02634     0.64646    -0.02065
0.68491     -0.01485     0.72309    -0.00930     0.74198    -0.00675     0.76068    -0.00442
0.79713     -0.00059     0.83177     0.00206     0.86398     0.00362     0.89331     0.00428
0.90679      0.00435     0.91943     0.00427     0.93122     0.00407     0.95212     0.00342
0.96934      0.00255     0.98278     0.00163     0.99237     0.00080     0.99572     0.00047
0.99810      0.00022     0.99953     0.00006     1.00000     0.00000
0.0         -1.0         0.0         0.0
```

```
SAMPLE CASE NO. 2--KORN AIRFOIL 75-06-12  TWO GRID HALVINGS ONLY

                         X-Y GRID SYSTEM

   2 -0.1410E 01    3 -0.4900E 00    4 -0.3706E 00    5 -0.2485E 00    6 -0.1247E 00    7  0.0
   8  0.1247E 00    9  0.2485E 00   10  0.3706E 00   11  0.4900E 00   12  0.1410E 01
   2 -0.4261E 00    3 -0.1420E 00    4 -0.2303E-07    5  0.1420E 00    6  0.4261E 00

   MACH NO. IS 0.752 ANGLE OF ATTACK IS 0.0   DEGREES
          DIRECT SOLUTION TO      0.50
                         CASE NUMBER    100

   INVISCID ANALYSIS CASE
   WITH VISCCUS INTERACTION
&FINP
M= 0.75199997    ,W=  1.6999998    ,X1= 0.50000000    ,X2=  10000.000    ,ALP= 0.0           ,EPS= 0.0          ,EPSS=
 0.39999998    ,X4= 0.48999995    ,S4=  2.0000000    ,CONV= 0.99999997E-05,A1= 0.24599999    ,A2= 0.14999998    ,A3=  3.8699999
RN=  20950000.    ,XIBDLY=-0.44000000    ,CIR= 0.0           ,CDCORR= 0.0           ,RDEL= 0.25000000    ,RDELFN= 0.12500000
SP= 0.39999969E-02,XSEP= 0.44000000    ,RCPB= 0.19999999    ,CPB= 0.39999998    ,XMON= 0.46999997    ,XLSEP= 0.50000000    ,XPC=
 0.99999964E-C1
&END
&IINP
IMAX=         13,JMAX=          7,IKASE=        100,INV=           0,MITER=        800,NHALF=          2,ITACT=          1,ISKP2=
         0,ISKP3=          0,ISKP4=          0,ITERP=          0,IREAD=          0,LP=        1000,ITEUPC=          0,ITELWC=
 .       0
&END

                       AIRFOIL COORDINATES
    X        YU        YL      UPPER SLOPE LCWER SLOPE
 -0.49000   0.01857  -0.01417   0.71872  -0.67988
 -0.37058   C.05153  -0.04429   0.11615  -0.11678
 -0.24852   0.C6140  -0.05176   0.05325  -0.01692
 -0.12470   0.C6542  -0.04970   0.01340   0.04823
  0.0       0.C6489  -0.04000  -0.02273   0.10586
  0.12470   0.C5913  -0.02390  -0.07384   0.14744
  0.24852   0.04574  -0.00591  -0.14295   0.12644
  0.37058   C.02459   0.00383  -0.19774   0.02909
  0.49000   0.00151   0.00102  -0.16299  -0.09073
ITERATION  10 CIR =  0.09421 DPM =  0.00389451 AT  4  3 NSSP =    0 DELTAY OR DELSTAR =  0.0
ITERATION  20 CIR =  0.12328 DPM =  0.00239694 AT 12  3 NSSP =    1 DELTAY OR DELSTAR =  0.0
ITERATION  30 CIR =  0.14425 DPM =  0.00196538 AT 10  3 NSSP =    3 DELTAY OR DELSTAR =  0.0
ITERATION  40 CIR =  0.16120 DPM =  0.00168639 AT 10  3 NSSP =    4 DELTAY OR DELSTAR =  0.0
ITERATION  50 CIR =  0.17540 DPM =  0.00143486 AT 10  3 NSSP =    4 DELTAY OR DELSTAR =  0.0
 CP BY CENTRAL DIFFERENCES
X          CPU         CPL
    -0.490     0.001     0.280
    -0.371    -0.947    -0.791
    -0.249    -0.722    -0.692
    -0.125    -0.679    -0.529
     0.0      -0.705    -0.324
     0.125    -0.659    -0.047
     0.249    -0.444     0.238
     0.371    -0.083     0.376
     0.490     0.570     0.651
     1.410     0.031     0.030
X            YU          YL          SLU         SLL
  -0.49000   0.01857  -0.01417   0.71872  -0.67988
  -0.37058   C.05153  -0.04429   0.11615  -0.11678
  -0.24852   0.06140  -0.05176   0.05325  -C.01692
  -0.12470   0.06542  -0.04970   0.01340   0.04823
   0.0       0.06489  -0.04000  -0.02273   0.10586
   0.12470   C.05913  -0.02390  -0.07384   C.14744
   0.24852   0.04574  -0.00591  -0.14295   0.12644
   0.37058   0.02459   0.00383  -0.19774   0.02909
   0.49000   0.00151   0.00102  -0.16299  -0.09073
 74 73 72 74 75
 68 62  0 71 75
 80 95  0101 85
 83 97  0101 90
 83 92  0101 93
 80 85  0102 93
 76 77  0 99 89
 73 69  0 90 84
 70 65  0 79 78
 55 49  0 58 68
 74 73 73 74 74

                    WAVE CD =  -0.002310
```

CHART 100



CPSTAR = -0.5842 CLCIR = 0.3508

CL = 0.3275 CD = -0.002310 CMLE = -0.1982 CDF = 0.0 CMC4 = -0.1163

```
SAMPLE CASE NO. 2--KORN AIRFOIL 76-08-12  TWO GRID HALVINGS ONLY

                         X-Y GRID SYSTEM

  2 -0.3872E 01    3 -0.1410E 01    4 -0.6471E 00    5 -0.4900E 00    6 -0.4307E 00    7 -0.3706E 00
  8 -0.3098E 00    9 -0.2485E 00   10 -0.1860E 00   11 -0.1247E 00   12 -0.6240E-01   13  0.0
 14  0.6240E-01   15  0.1247E 00   16  0.1860E 00   17  0.2485E 00   18  0.3098E 00   19  0.3706E 00
 20  0.4307E 00   21  0.4900E 00   22  0.6471E 00   23  0.1410E 01   24  0.3872E 01
  2 -0.9181E 00    3 -0.4261E 00    4 -0.2460E 00    5 -0.1420E 00    6 -0.6592E-01    7 -0.9213E-07
  8  0.6592E-01    9  0.1420E 00   10  0.2460E 00   11  0.4261E 00   12  0.9181E 00

  MACH NO. IS 0.752 ANGLE OF ATTACK IS 0.0   DEGREES
        DIRECT SOLUTION TO      0.50
                         CASE NUMBER   100

  INVISCID ANALYSIS CASE
  WITH VISCOUS INTERACTION
&FINP
M= 0.75199997    ,W=  1.4999998    ,X1= 0.50000000    ,X2=  10000.000    ,ALP= 0.0             ,EPS= 0.0             ,EPSS=
 0.39999998    ,XA= 0.48999995    ,SA=  2.0000000    ,CONV= 0.99999997E-05,A1= 0.24599999    ,A2= 0.14999998    ,A3=  3.8699999
RN=  20950000.    ,XIBDLY=-0.44000000    ,CIR= 0.17540183    ,CDCORR= 0.0              ,RDEL= 0.25000000    ,RDELFN= 0.12500000    ,
SP= 0.39999985E-02,XSEP= 0.44000000    ,RCPB= 0.19999999    ,CPB= 0.39999998    ,XMDN= 0.46999997    ,XLSEP= 0.50000000    ,XPC=
 0.99999984E-01
&END
&IINP
IMAX=          25,JMAX=          13,IKASE=         100,INV=            0,NITER=          400,NHALF=           2,ITACT=           1,ISKP2=
          0,ISKP3=          0,ISKP4=          0,ITERP=          0,IREAD=          0,LP=        1000,ITEUPC=          0,ITELWC=
          0
&END

                        AIRFOIL COORDINATES
     X          YU          YL        UPPER SLOPE LOWER SLOPE
  -0.49000    0.01657   -0.01417    0.71872   -0.67988
  -0.43067    0.04203   -0.03461    0.23468   -0.21554
  -0.37058    0.05153   -0.04429    0.11615   -0.11678
  -0.30982    0.05736   -0.04951    0.07984   -0.05735
  -0.24852    0.06140   -0.05176    0.05325   -0.01692
  -0.18677    0.06401   -0.05174    0.03230    0.01706
  -0.12470    0.06542   -0.04970    0.01340    0.04823
  -0.06240    0.06572   -0.04576   -0.00416    0.07808
   0.0        0.06489   -0.04000   -0.02273    0.10586
   0.06240    0.06280   -0.03260   -0.04508    0.13057
   0.12470    0.05913   -0.02390   -0.07384    0.14744
   0.18677    0.05350   -0.01457   -0.10807    0.14944
   0.24852    0.04574   -0.00591   -0.14295    0.12644
   0.30982    0.03599    0.00056   -0.17483    0.08037
   0.37058    0.02459    0.00383   -0.19774    0.02909
   0.43067    0.01250    0.00408   -0.19995   -0.02139
   0.49000    0.00151    0.00102   -0.18299   -0.09073
ITERATION  10 CIR =  0.18502 DPM =  0.00200450 AT 12   3 NSSP =   19 DELTAY OR DELSTAR =  0.0
ITERATION  20 CIR =  0.19592 DPM =  0.00099790 AT 24   6 NSSP =   17 DELTAY OR DELSTAR =  0.0
ITERATION  30 CIR =  0.20530 DPM =  0.00087315 AT 24   6 NSSP =   17 DELTAY OR DELSTAR =  0.0
ITERATION  40 CIR =  0.21345 DPM =  0.00076777 AT 24   6 NSSP =   17 DELTAY OR DELSTAR =  0.0
ITERATION  50 CIR =  0.22063 DPM =  0.00070268 AT 22   6 NSSP =   17 DELTAY OR DELSTAR =  0.0
ITERATION  60 CIR =  0.21917 DPM =  0.00061113 AT 15   6 NSSP =   16 DELTAY OR DELSTAR =  0.0057
ITERATION  70 CIR =  0.21802 DPM =  0.00046831 AT 15   6 NSSP =   15 DELTAY OR DELSTAR =  0.0077
ITERATION  80 CIR =  0.21730 DPM =  0.00035739 AT 15   6 NSSP =   15 DELTAY OR DELSTAR =  0.0088
ITERATION  90 CIR =  0.21700 DPM =  0.00027806 AT 15   6 NSSP =   14 DELTAY OR DELSTAR =  0.0095
ITERATION 100 CIR =  0.21704 DPM =  0.00022197 AT 15   6 NSSP =   14 DELTAY OR DELSTAR =  0.0098
ITERATION 110 CIR =  0.21734 DPM =  0.00018233 AT 15   6 NSSP =   14 DELTAY OR DELSTAR =  0.0101
ITERATION 120 CIR =  0.21781 DPM =  0.00015372 AT 15   6 NSSP =   14 DELTAY OR DELSTAR =  0.0102
ITERATION 130 CIR =  0.21837 DPM =  0.00013280 AT 16   6 NSSP =   14 DELTAY OR DELSTAR =  0.0103
ITERATION 140 CIR =  0.21900 DPM =  0.00011708 AT 16   6 NSSP =   14 DELTAY OR DELSTAR =  0.0103
ITERATION 150 CIR =  0.21965 DPM =  0.00010478 AT 16   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 160 CIR =  0.22029 DPM =  0.00009495 AT 17   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 170 CIR =  0.22092 DPM =  0.00008661 AT 17   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 180 CIR =  0.22153 DPM =  0.00007939 AT 18   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 190 CIR =  0.22210 DPM =  0.00007308 AT 18   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 200 CIR =  0.22264 DPM =  0.00006717 AT 18   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 210 CIR =  0.22314 DPM =  0.00006223 AT 18   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 220 CIR =  0.22361 DPM =  0.00005734 AT 17   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 230 CIR =  0.22405 DPM =  0.00005305 AT 18   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 240 CIR =  0.22445 DPM =  0.00004905 AT 18   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 250 CIR =  0.22483 DPM =  0.00004530 AT 19   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 260 CIR =  0.22518 DPM =  0.00004190 AT 18   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 270 CIR =  0.22550 DPM =  0.00003862 AT 18   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 280 CIR =  0.22580 DPM =  0.00003576 AT 19   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 290 CIR =  0.22607 DPM =  0.00003392 AT 24   5 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 300 CIR =  0.22633 DPM =  0.00003052 AT 18   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 310 CIR =  0.22656 DPM =  0.00002825 AT 18   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 320 CIR =  0.22678 DPM =  0.00002611 AT 17   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 330 CIR =  0.22698 DPM =  0.00002414 AT 19   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 340 CIR =  0.22716 DPM =  0.00002217 AT 18   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 350 CIR =  0.22733 DPM =  0.00002122 AT 22   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 360 CIR =  0.22749 DPM =  0.00001907 AT 17   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 370 CIR =  0.22764 DPM =  0.00001752 AT 19   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 380 CIR =  0.22777 DPM =  0.00001609 AT 16   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 390 CIR =  0.22790 DPM =  0.00001574 AT 22   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
ITERATION 400 CIR =  0.22801 DPM =  0.00001377 AT 19   6 NSSP =   14 DELTAY OR DELSTAR =  0.0104
```

BOUNDARY LAYER ANALYSIS FOR REYNOLDS NUMBER OF 0.210E 08

X	YUORIG	DU	SLU	YLORIG	DL	SLL
-0.49000	0.01857	0.0	0.71872	-0.01417	0.0	-0.67988
-0.43067	0.04203	0.00008	0.17769	-0.03461	0.00007	-0.17098
-0.37058	0.05153	0.00021	0.13012	-0.04429	0.00019	-0.13127
-0.30982	0.05738	0.00035	0.07848	-0.04951	0.00033	-0.05765
-0.24852	0.06140	0.00050	0.05615	-0.05176	0.00047	-0.02020
-0.18677	0.06401	0.00063	0.03398	-0.05174	0.00062	0.01480
-0.12470	0.06542	0.00076	0.01570	-0.04970	0.00078	0.04534
-0.06240	0.06572	0.00088	-0.00212	-0.04576	0.00098	0.07463
0.0	0.06489	0.00101	-0.02054	-0.04000	0.00122	0.10156
0.06240	0.06288	0.00116	-0.04229	-0.03260	0.00155	0.12413
0.12470	0.05913	0.00138	-0.06975	-0.02390	0.00199	0.13926
0.18677	0.05350	0.00170	-0.10175	-0.01457	0.00253	0.14060
0.24852	0.04574	0.00217	-0.13320	-0.00591	0.00303	0.11936
0.30982	0.03589	0.00291	-0.15907	0.00050	0.00333	0.07761
0.37058	0.02459	0.00439	-0.15981	0.00383	0.00343	0.02962
0.43067	0.01250	0.00742	-0.14217	0.00408	0.00342	-0.02284
0.49000	0.00151	0.01040	-0.12734	0.00102	0.00341	-0.08048

CP BY CENTRAL DIFFERENCES

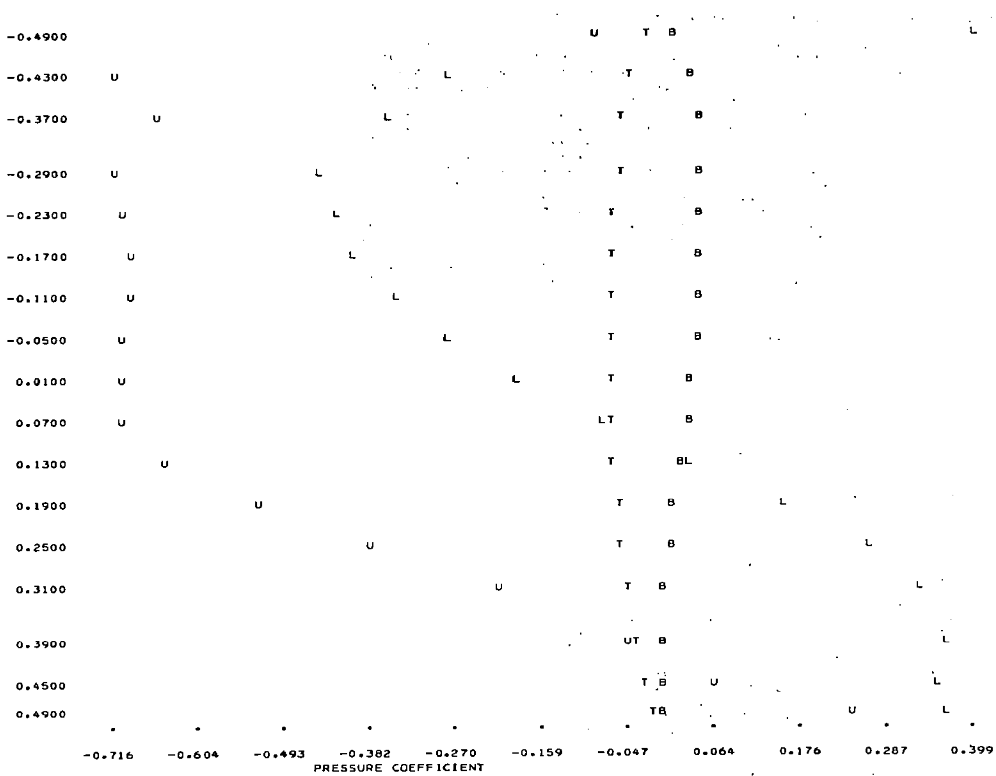
X	CPU	CPL
-0.490	-0.083	0.399
-0.431	-0.716	-0.271
-0.371	-0.654	-0.351
-0.310	-0.715	-0.437
-0.249	-0.703	-0.421
-0.187	-0.698	-0.396
-0.125	-0.689	-0.345
-0.062	-0.702	-0.276
0.0	-0.703	-0.186
0.062	-0.657	-0.080
0.125	-0.641	0.038
0.187	-0.524	0.163
0.249	-0.380	0.273
0.310	-0.210	0.342
0.371	-0.045	0.369
0.431	0.065	0.361
0.490	0.243	0.371
0.647	0.083	0.083
1.410	0.023	0.023
3.872	0.003	0.003

X	YU	YL	SLU	SLL
-0.49000	0.01857	-0.01417	0.71872	-0.67988
-0.43067	0.04211	-0.03468	0.17769	-0.17098
-0.37058	0.05174	-0.04448	0.13012	-0.13127
-0.30982	0.05774	-0.04983	0.07848	-0.05765
-0.24852	0.06190	-0.05224	0.05615	-0.02020
-0.18677	0.06464	-0.05236	0.03398	0.01480
-0.12470	0.06618	-0.05049	0.01570	0.04534
-0.06240	0.06660	-0.04674	-0.00212	0.07463
0.0	0.06590	-0.04123	-0.02054	0.10156
0.06240	0.06397	-0.03416	-0.04229	0.12413
0.12470	0.06051	-0.02591	-0.06975	0.13926
0.18677	0.05521	-0.01712	-0.10175	0.14060
0.24852	0.04793	-0.00896	-0.13320	0.11936
0.30982	0.03894	-0.00285	-0.15907	0.07761
0.37058	0.02904	0.00040	-0.15981	0.02962
0.43067	0.01999	0.00066	-0.14217	-0.02284
0.49000	0.01199	-0.00240	-0.12734	-0.08048

74	74	74	74	74	74	75	75	75	75	75
74	73	73	73	73	73	73	73	73	74	75
73	71	70	69	67	64	68	71	73	74	77
73	72	68	62	58	0	75	76	80	83	83
74	75	75	77	81	0	98	90	86	84	81
74	76	79	83	87	0	101	96	91	87	82
74	78	82	86	92	0	105	100	95	89	83
74	78	83	87	52	0	105	101	97	91	83
74	79	83	87	51	0	104	101	98	92	84
74	78	82	85	88	0	0	102	98	93	84
74	77	80	83	85	0	0	102	98	93	84
74	76	78	80	82	0	105	102	98	93	84
74	75	76	77	78	0	104	101	97	91	84
74	74	73	73	73	0	102	98	94	90	83
73	72	71	70	68	0	96	93	90	87	82
73	71	69	67	65	0	90	88	87	85	82
73	70	68	66	63	0	83	83	83	82	81
73	70	67	65	62	0	77	79	80	80	80
72	70	67	65	63	0	73	75	77	79	79
71	67	64	61	60	0	68	71	75	78	81
72	71	71	71	71	71	72	73	74	75	76
74	74	74	74	74	74	74	74	74	74	75
74	74	75	75	75	75	75	75	75	75	75

WAVE CD = 0.003733

CHART 100



CPSTAR = -0.5842 CLCIR = 0.4560

CL = 0.4384 CD = 0.010745 CMLE = -0.2306 CDF = 0.007012 CMC4 = -0.1210

```
SAMPLE CASE NO. 2--KORN AIRFOIL 75-06-12  TWO GRID HALVINGS ONLY

                         X-Y GRID SYSTEM

   2 -0.8027E 01    3 -0.3872E 01    4 -0.2276E 01    5 -0.1410E 01    6 -0.9115E 00    7 -0.6471E 00
   8 -0.5317E 00    9 -0.4900E 00   10 -0.4604E 00   11 -0.4307E 00   12 -0.4007E 00   13 -0.3706E 00
  14 -0.3403E 00   15 -0.3098E 00   16 -0.2792E 00   17 -0.2485E 00   18 -0.2177E 00   19 -0.1868E 00
  20 -0.1558E 00   21 -0.1247E 00   22 -0.9357E-01   23 -0.6240E-01   24 -0.3121E-01   25  0.0
  26  0.3121E-01   27  0.6240E-01   28  0.9357E-01   29  0.1247E 00   30  0.1558E 00   31  0.1868E 00
  32  0.2177E 00   33  0.2485E 00   34  0.2792E 00   35  0.3098E 00   36  0.3403E 00   37  0.3706E 00
  38  0.4007E 00   39  0.4307E 00   40  0.4604E 00   41  0.4900E 00   42  0.5317E 00   43  0.6471E 00
  44  0.9115E 00   45  0.1410E 01   46  0.2276E 01   47  0.3872E 01   48  0.8027E 01
   2 -0.1869E 01    3 -0.9181E 00    4 -0.5939E 00    5 -0.4261E 00    6 -0.3206E 00    7 -0.2460E 00
   8 -0.1888E 00    9 -0.1420E 00   10 -0.1019E 00   11 -0.6592E-01   12 -0.3239E-01   13 -0.9213E-07
  14  0.3239E-01   15  0.6592E-01   16  0.1019E 00   17  0.1420E 00   18  0.1888E 00   19  0.2460E 00
  20  0.3206E 00   21  0.4261E 00   22  0.5939E 00   23  0.9181E 00   24  0.1869E 01

   MACH NO. IS 0.752 ANGLE OF ATTACK IS 0.0   DEGREES
          DIRECT SOLUTION TO      0.50
                     CASE NUMBER   100

   INVISCID ANALYSIS CASE
   WITH VISCOUS INTERACTION
&FINP
M= 0.75199997    ,W=  1.6999998    ,X1= 0.50000000    ,X2=  10000.000    ,ALP= 0.0            ,EPS= 0.0           ,EPSS=
 0.39999998    ,X4= 0.48999995    ,S4=  2.0000000    ,CONV= 0.99999997E-05,A1= 0.24599999    ,A2= 0.14999998    ,A3= 3.8699999
RN=  20950000.    ,XIBDLY=-0.44000000    ,CIR= 0.22801352    ,CDCORR= 0.0            ,RDEL= 0.25000000    ,RDELFN= 0.12500000    ,
SP= 0.39999969E-02,XSEP= 0.44000000    ,RCPB= 0.19999999    ,CPB= 0.39999998    ,XMON= 0.46999997    ,XLSEP= 0.50000000    ,XPC=
 0.99999964E-01
&END
&IINP
IMAX=          49,JMAX=          25,IKASE=         100,INV=            0,MITER=         200,NHALF=           2,ITACT=           1,ISKP2=
           0,ISKP3=           0,ISKP4=           0,ITERP=           0,IREAD=           0,LP=        1000,ITEUPC=           0,ITELWC=
           0
&END

                      AIRFOIL COORDINATES
    X         YU         YL       UPPER SLOPE LOWER SLOPE
 -0.49000   0.01857   -0.01417   0.71872   -0.67988
 -0.46044   0.03318   -0.02691   0.36638   -0.31117
 -0.43067   0.04211   -0.03468   0.17769   -0.17098
 -0.40072   0.04692   -0.03958   0.15391   -0.15113
 -0.37058   0.05174   -0.04448   0.13012   -0.13127
 -0.34028   0.05474   -0.04716   0.10430   -0.09446
 -0.30982   0.05774   -0.04983   0.07848   -0.05765
 -0.27923   0.05582   -0.05103   0.06731   -0.03893
 -0.24852   0.06190   -0.05224   0.05615   -0.02020
 -0.21769   0.06327   -0.05230   0.04506   -0.00270
 -0.18677   0.06464   -0.05236   0.03398    0.01480
 -0.15577   0.06541   -0.05142   0.02484    0.03007
 -0.12470   0.06618   -0.05049   0.01570    0.04534
 -0.09357   0.06639   -0.04861   0.00679    0.05999
 -0.06240   0.06660   -0.04674  -0.00212    0.07463
 -0.03121   0.06625   -0.04398  -0.01133    0.08809
  0.0       0.06590   -0.04123  -0.02054    0.10156
  0.03121   0.06494   -0.03769  -0.03142    0.11284
  0.06240   0.06397   -0.03416  -0.04229    0.12413
  0.09357   0.06224   -0.03003  -0.05602    0.13169
  0.12470   0.06051   -0.02591  -0.06975    0.13926
  0.15577   0.05786   -0.02152  -0.08575    0.13993
  0.18677   0.05520   -0.01712  -0.10175    0.14060
  0.21769   0.05157   -0.01304  -0.11748    0.12998
  0.24852   0.04793   -0.00896  -0.13320    0.11936
  0.27923   0.04344   -0.00590  -0.14614    0.09849
  0.30982   0.03894   -0.00285  -0.15907    0.07761
  0.34028   0.03399   -0.00122  -0.15944    0.05362
  0.37058   0.02904    0.00040  -0.15981    0.02962
  0.40072   0.02451    0.00053  -0.15099    0.00339
  0.43067   0.01999    0.00066  -0.14217   -0.02284
  0.46044   0.01599   -0.00087  -0.13475   -0.05166
  0.49000   0.01199   -0.00240  -0.12734   -0.08048
ITERATION  10 CIR =  0.22783 DPM =  0.00033033 AT 32  7 NSSP =   57 DELTAY OR DELSTAR =  0.0104
ITERATION  20 CIR =  0.22797 DPM =  0.00007644 AT 24 24 NSSP =   58 DELTAY OR DELSTAR =  0.0104
ITERATION  30 CIR =  0.22812 DPM =  0.00004059 AT 37  2 NSSP =   58 DELTAY OR DELSTAR =  0.0104
ITERATION  40 CIR =  0.22824 DPM =  0.00002819 AT 35  2 NSSP =   58 DELTAY OR DELSTAR =  0.0104
ITERATION  50 CIR =  0.22834 DPM =  0.00002676 AT 46 12 NSSP =   57 DELTAY OR DELSTAR =  0.0104
ITERATION  60 CIR =  0.22724 DPM =  0.00029546 AT 12 15 NSSP =   60 DELTAY OR DELSTAR =  0.0101
ITERATION  70 CIR =  0.22756 DPM =  0.00009030 AT 14 15 NSSP =   62 DELTAY OR DELSTAR =  0.0094
ITERATION  80 CIR =  0.22719 DPM =  0.00004506 AT 14 15 NSSP =   62 DELTAY OR DELSTAR =  0.0095
ITERATION  90 CIR =  0.22795 DPM =  0.00005800 AT 48 12 NSSP =   61 DELTAY OR DELSTAR =  0.0089
ITERATION 100 CIR =  0.22785 DPM =  0.00002313 AT 15 15 NSSP =   62 DELTAY OR DELSTAR =  0.0091
ITERATION 110 CIR =  0.22768 DPM =  0.00001861 AT 47 13 NSSP =   62 DELTAY OR DELSTAR =  0.0092
ITERATION 120 CIR =  0.22799 DPM =  0.00002056 AT 47 13 NSSP =   62 DELTAY OR DELSTAR =  0.0090
ITERATION 130 CIR =  0.22788 DPM =  0.00002754 AT 48 11 NSSP =   62 DELTAY OR DELSTAR =  0.0092
ITERATION 140 CIR =  0.22771 DPM =  0.00001436 AT 44 13 NSSP =   62 DELTAY OR DELSTAR =  0.0093
ITERATION 150 CIR =  0.22755 DPM =  0.00001826 AT 48 13 NSSP =   62 DELTAY OR DELSTAR =  0.0093
ITERATION 160 CIR =  0.22741 DPM =  0.00001532 AT 48 12 NSSP =   62 DELTAY OR DELSTAR =  0.0094
ITERATION 170 CIR =  0.22728 DPM =  0.00001511 AT 46 13 NSSP =   62 DELTAY OR DELSTAR =  0.0094
ITERATION 180 CIR =  0.22731 DPM =  0.00001040 AT 31 15 NSSP =   62 DELTAY OR DELSTAR =  0.0093
ITERATION 190 CIR =  0.22720 DPM =  0.00001144 AT 42 13 NSSP =   62 DELTAY OR DELSTAR =  0.0094
ITERATION 200 CIR =  0.22709 DPM =  0.00001460 AT 47  7 NSSP =   62 DELTAY OR DELSTAR =  0.0094
```

BOUNDARY LAYER ANALYSIS FOR REYNOLDS NUMBER OF 0.210E 08

X	YUORIG	DU	SLU	YLORIG	DL	SLL
-0.49000	0.01657	0.0	0.71872	-0.01417	0.0	-0.67988
-0.46044	0.03318	0.00001	0.36638	-0.02691	0.00001	-0.31117
-0.43067	0.04203	0.00005	0.24494	-0.03461	0.00005	-0.22472
-0.40072	0.04758	0.00012	0.15056	-0.04017	0.00011	-0.15910
-0.37058	0.05153	0.00021	0.11970	-0.04429	0.00019	-0.12047
-0.34028	0.05473	0.00030	0.09825	-0.04736	0.00026	-0.08720
-0.30982	0.05738	0.00038	0.08150	-0.04951	0.00033	-0.06020
-0.27923	0.05958	0.00046	0.06783	-0.05095	0.00040	-0.03884
-0.24852	0.06140	0.00053	0.05565	-0.05176	0.00048	-0.01937
-0.21769	0.06286	0.00061	0.04433	-0.05201	0.00055	-0.00181
-0.18677	0.06401	0.00068	0.03448	-0.05174	0.00062	0.01457
-0.15577	0.06486	0.00074	0.02471	-0.05096	0.00070	0.03026
-0.12470	0.06542	0.00080	0.01538	-0.04970	0.00079	0.04542
-0.09357	0.06571	0.00086	0.00668	-0.04797	0.00088	0.06022
-0.06240	0.06572	0.00092	-0.00227	-0.04576	0.00098	0.07468
-0.03121	0.06545	0.00097	-0.01127	-0.04310	0.00109	0.08857
0.0	0.06489	0.00103	-0.02074	-0.04000	0.00122	0.10139
0.03121	0.06402	0.00110	-0.03105	-0.03649	0.00137	0.11360
0.06240	0.06280	0.00118	-0.04234	-0.03260	0.00154	0.12447
0.09357	0.06119	0.00127	-0.05515	-0.02838	0.00174	0.13299
0.12470	0.05913	0.00139	-0.06954	-0.02390	0.00198	0.13908
0.15577	0.05658	0.00154	-0.08554	-0.01925	0.00225	0.14153
0.18677	0.05350	0.00171	-0.10202	-0.01457	0.00256	0.13927
0.21769	0.04989	0.00191	-0.11823	-0.01006	0.00286	0.13190
0.24852	0.04574	0.00217	-0.13341	-0.00591	0.00313	0.11894
0.27923	0.04110	0.00249	-0.14692	-0.00234	0.00332	0.10064
0.30982	0.03599	0.00292	-0.15846	0.00050	0.00341	0.07903
0.34028	0.03045	0.00349	-0.16610	0.00256	0.00343	0.05558
0.37058	0.02459	0.00424	-0.16965	0.00383	0.00338	0.03145
0.40072	0.01855	0.00514	-0.17077	0.00434	0.00331	0.00710
0.43067	0.01250	0.00623	-0.15510	0.00408	0.00319	-0.01580
0.46044	0.00672	0.00779	-0.13173	0.00304	0.00301	-0.04429
0.49000	0.00151	0.00933	-0.11632	0.00102	0.00282	-0.08015

CP BY CENTRAL DIFFERENCES

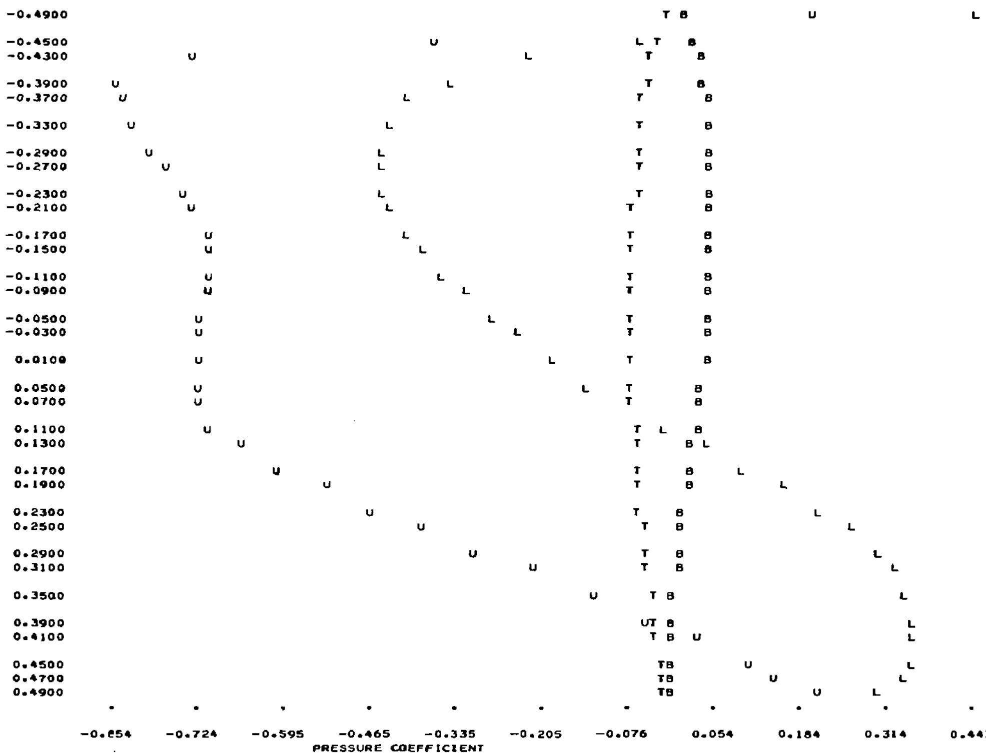
X	CPU	CPL
-0.490	0.157	0.443
-0.460	-0.367	-0.059
-0.431	-0.733	-0.229
-0.401	-0.854	-0.345
-0.371	-0.840	-0.400
-0.340	-0.824	-0.439
-0.310	-0.800	-0.450
-0.279	-0.774	-0.449
-0.249	-0.750	-0.443
-0.218	-0.726	-0.429
-0.187	-0.710	-0.410
-0.156	-0.702	-0.385
-0.125	-0.701	-0.356
-0.094	-0.706	-0.322
-0.062	-0.713	-0.282
-0.031	-0.721	-0.237
0.0	-0.724	-0.188
0.031	-0.723	-0.135
0.062	-0.719	-0.074
0.094	-0.699	-0.014
0.125	-0.655	0.044
0.156	-0.595	0.105
0.187	-0.529	0.164
0.218	-0.457	0.220
0.249	-0.380	0.268
0.279	-0.300	0.307
0.310	-0.215	0.335
0.340	-0.126	0.352
0.371	-0.045	0.361
0.401	0.035	0.360
0.431	0.113	0.354
0.460	0.154	0.343
0.490	0.213	0.310
0.532	0.181	0.181
0.647	0.097	0.097
0.912	0.047	0.047
1.410	0.020	0.020
2.276	0.008	0.008
3.872	0.003	0.003
8.027	0.001	0.001

X	YU	YL	SLU	SLL
-0.49000	0.01857	-0.01417	0.71872	-0.67988
-0.46044	0.03319	-0.02692	0.36638	-0.31117
-0.43067	0.04208	-0.03466	0.24494	-0.22472
-0.40072	0.04771	-0.04029	0.15056	-0.15910
-0.37058	0.05175	-0.04448	0.11970	-0.12047
-0.34028	0.05503	-0.04762	0.09825	-0.08720
-0.30982	0.05776	-0.04984	0.08150	-0.06020
-0.27923	0.06004	-0.05135	0.06783	-0.03884
-0.24852	0.06193	-0.05224	0.05565	-0.01937
-0.21769	0.06347	-0.05256	0.04433	-0.00181
-0.18677	0.06469	-0.05236	0.03448	0.01457
-0.15577	0.06560	-0.05166	0.02471	0.03026
-0.12470	0.06622	-0.05049	0.01538	0.04542
-0.09357	0.06657	-0.04884	0.00668	0.06022
-0.06240	0.06664	-0.04674	-0.00227	0.07468
-0.03121	0.06643	-0.04419	-0.01127	0.08857
0.0	0.06593	-0.04122	-0.02074	0.10139
0.03121	0.06512	-0.03787	-0.03105	0.11360
0.06240	0.06398	-0.03415	-0.04234	0.12447
0.09357	0.06247	-0.03013	-0.05515	0.13299
0.12470	0.06053	-0.02589	-0.06954	0.13908
0.15577	0.05812	-0.02152	-0.08554	0.14153
0.18677	0.05522	-0.01715	-0.10202	0.13927
0.21769	0.05181	-0.01295	-0.11823	0.13190
0.24852	0.04793	-0.00907	-0.13341	0.11894
0.27923	0.04362	-0.00568	-0.14692	0.10064
0.30982	0.03894	-0.00293	-0.15846	0.07903
0.34028	0.03399	-0.00087	-0.16610	0.05558
0.37058	0.02889	0.00044	-0.16965	0.03145
0.40072	0.02376	0.00103	-0.17077	0.00710
0.43067	0.01880	0.00089	-0.15510	-0.01580
0.46044	0.01457	0.00003	-0.13173	-0.04429
0.49000	0.01090	-0.00181	-0.11632	-0.08015

```
75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75
74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 75 75 75
74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 75 75
74 73 73 73 73 73 73 73 73 73 73 73 73 73 73 73 73 74 74 74 74 75 76
74 73 72 72 71 71 71 71 71 71 71 71 71 71 71 71 72 72 73 74 75 76 77
74 73 72 71 70 69 68 67 66 65 64 64 65 66 67 69 70 72 73 75 76 78 78
73 73 73 72 70 68 66 64 61 57 52 44 55 61 65 69 72 74 76 78 79 80 78
73 74 74 73 72 71 70 68 65 62 58  0 67 70 74 76 78 80 81 81 82 81 78
74 74 74 74 74 74 73 73 73 73 75  0  0 86 84 83 83 83 83 82 82 81 78
74 74 75 75 75 76 76 77 78 81  0  0  0100 94 91 88 86 85 84 83 81 78
74 74 75 76 77 79 79 80 83 86  0  0  0108101 97 93 90 88 86 84 81 78
74 74 75 77 78 79 81 83 86 89  0  0  0109105101 97 93 90 87 85 82 78
74 74 76 77 79 81 83 86 88 91  0  0  0109106103 99 96 92 89 86 82 78
74 74 76 78 80 82 84 87 90 92  0  0  0109106104101 97 94 90 87 83 78
74 75 76 78 81 83 85 88 90 93  0  0  0108106103101 98 95 91 87 83 78
74 75 76 79 81 83 86 88 90 92  0  0  0107105103101 98 96 92 88 83 79
74 75 77 79 81 84 86 88 90 92  0  0  0106104103101 99 96 93 89 84 79
74 75 77 79 81 83 85 87 89 91  0  0  0105104102100 99 96 93 89 84 79
74 75 76 79 81 83 85 87 88 90  0  0  0105103102100 99 96 93 89 84 79
74 75 76 78 80 82 84 86 87 89  0  0  0   0103102101 99 97 94 90 84 79
74 74 76 78 80 81 83 85 86 87  0  0  0   0104102101 99 97 94 90 84 79
74 74 76 78 79 81 82 83 84 86  0  0  0   0104103101 99 97 94 90 84 79
74 74 76 77 78 79 81 82 83 84  0  0  0   0104103101 99 97 94 89 84 79
74 74 75 76 77 78 79 80 81 82  0  0  0   0104103101 99 96 93 89 84 79
74 74 75 76 76 77 78 78 79 80  0  0  0106104102101 98 96 93 89 84 79
74 74 74 75 78 76 76 77 77 77  0  0  0105104102100 98 95 92 88 84 79
74 74 74 74 74 75 75 75 75 75 75  0  0104103101 98 96 94 91 88 83 79
74 74 74 74 74 73 73 73 73 73 73  0  0102100 99 97 94 92 90 87 83 79
74 73 73 73 73 72 72 71 71 71 70  0  0100 98 96 94 93 91 88 86 83 78
73 73 73 72 72 71 71 70 69 69 68  0  0 97 95 93 92 91 89 87 85 82 78
73 73 73 72 71 70 69 69 68 67 66  0  0 93 92 91 90 89 87 86 84 82 78
73 73 72 71 70 69 68 67 66 65 64  0  0 90 89 88 87 87 86 85 83 82 78
73 73 72 71 70 69 68 66 65 64 63  0  0 87 86 86 85 85 84 83 83 81 78
73 73 72 71 69 68 67 66 65 63 62  0  0 83 83 83 83 83 83 82 82 81 78
73 73 72 70 69 68 67 65 64 63 61  0  0 80 80 81 81 81 81 81 81 80 78
73 73 71 70 69 68 66 65 64 62 61  0 77 77 78 78 79 79 80 80 80 80 78
73 73 71 70 69 67 66 65 64 62 61  0 74 75 76 76 77 78 79 79 80 80 78
73 72 71 70 69 68 66 65 64 63 61  0 71 72 74 75 76 77 78 78 79 79 78
73 72 71 70 69 68 67 66 65 63 62  0 69 71 72 74 75 76 77 78 79 79 78
73 72 71 69 68 67 66 65 64 63 62  0 67 69 71 72 74 75 76 77 79 79 78
73 72 71 70 69 68 67 67 67 67 67 67 68 69 70 72 73 74 75 76 77 78 78
73 72 71 71 70 70 70 70 70 70 70 71 71 71 72 72 73 73 74 75 76 77 77
73 73 72 72 72 72 72 72 72 73 73 73 73 73 73 73 73 74 74 74 75 76 77
74 73 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 75 75 76
74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 75 75 75 75
74 74 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75
75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75 75
```

WAVE CD = 0.006996

CHART 100



CPSTAR = -0.5842 CLCIR = 0.4542

CL = 0.4481 CD = 0.014111 CMLE = -0.2323 CDF = 0.007115 CMC4 = -0.1203

REFERENCES

1. Carlson, L.A.: "Transonic Airfoil Flowfield Analysis Using Cartesion Coordinates," NASA CR-2577, August 1975.

2. Carlson, L.A.: "Transonic Airfoil Design Using Cartesion Coordinates," NASA CR-2578, April 1976.

3. Carlson, L.A.: "Transonic Airfoil Analysis and Design Using Cartesian Coordinates," AIAA Journal, Vol. 13, No. 5, May 1976, pp. 349-356.

4. South, J.C., Jr. and Jameson, A.: "Relaxation Solutions for Inviscid Axisymmetric Transonic Flow over Blunt or Pointed Bodies," Proc. AIAA Computational Fluid Dynamics Conf., Palm Springs, Calif., July 1973, pp. 8-17.

5. Nash, J.F. and Macdonald, A.G.J.: "The Calculation of Momentum Thickness in a Turbulent Boundary Layer at Mach Numbers up to Unity," Aero. Res. Council C.P. No. 963, 1967.

6. Bauer, F.; Garabedian, P,; Korn, D.; and Jameson, A.: "Supercritical Wing Sections II," Springer-Verlag, New York, 1975.

7. Bavitz, P.C.: "An Analysis Method for Two-Dimensional Transonic Viscous Flow," NASA TN-D-7718, January 1975.

NASA-Langley, 1977