

UNCLASSIFIED

ERDA Mathematics and Computing Laboratory  
Courant Institute of Mathematical Sciences  
New York University

Mathematics and Computers      COO-3077-140

NUMERICAL CALCULATION OF THE TRANSONIC FLOW

PAST A SWEEP WING

Antony Jameson  
New York University

and

D. A. Caughey  
Cornell University

June 1977

U. S. Energy Research and Development Administration  
Contract EY-76-C-02-3077\*000

This work was supported by NASA Grants  
NGR-33-016-167    and    NGR-33-016-201.

UNCLASSIFIED

NOTICE  
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

## **DISCLAIMER**

**This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.**

## **DISCLAIMER**

**Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.**

THIS PAGE  
WAS INTENTIONALLY  
LEFT BLANK

## Table of Contents

	<u>Page</u>
ABSTRACT.....	v
INTRODUCTION.....	1
ANALYSIS.....	4
RESULTS.....	16
CONCLUSIONS.....	21
REFERENCES.....	22
FIGURES.....	24
APPENDIX A. USER'S GUIDE TO THE PROGRAM.....	38
APPENDIX B. LISTING OF THE PROGRAM.....	51

THIS PAGE  
WAS INTENTIONALLY  
LEFT BLANK.

## ABSTRACT

A numerical method is presented for analyzing the transonic potential flow past a lifting, swept wing. A finite-difference approximation to the full potential equation is solved in a coordinate system which is nearly conformally mapped from the physical space in planes parallel to the symmetry plane, and reduces the wing surface to a portion of one boundary of the computational grid. A coordinate invariant, rotated difference scheme is used, and the difference equations are solved by relaxation. The method is capable of treating wings of arbitrary planform and dihedral, although approximations in treating the tips and vortex sheet make its accuracy suspect for wings of small aspect ratio. Comparisons of calculated results with experimental data are shown for examples of both conventional and supercritical transport wings. Agreement is quite good for both types, but it was found necessary to account for the displacement effect of the boundary layer for the supercritical wing, presumably because of its greater sensitivity to changes in effective geometry.

## INTRODUCTION

The development of profile shapes capable of efficient operation in the transonic regime has spurred interest in flight vehicles designed specifically to operate at near sonic speeds. The ability to predict accurately the aerodynamic characteristics of the complete three-dimensional wing should have a substantial impact on the design of such vehicles by allowing detailed trade-off studies to be performed without recourse to wind tunnel testing of every design variation.

Recent advances in the theoretical prediction of inviscid transonic flow fields are based largely on type-dependent, finite-difference solutions of the steady potential equation. These methods were first applied to the transonic small disturbance equation by Murman and Cole [1], and the full potential equation by Jameson [2] and Garabedian and Korn [3] for the prediction of airfoil flow fields. The three-dimensional small disturbance equation has also been solved for swept wings by Ballhaus and Bailey [4] and for wing-cylinder combinations by Bailey and Ballhaus [5]. Finally, the full potential equation has been solved by Jameson for the transonic flow over an oblique yawed wing [6]. Although an oblique wing should be aerodynamically more efficient than a conventional swept wing [7], it presents problems of stability and control and aeroelastic divergence. We consider here the prediction of the flow over a swept wing.

In Jameson's treatment of the flow over oblique wings, the coordinate system is aligned in planes normal to the wing



leading edge. Thus, for nonzero angles of yaw the free stream velocity vector is not contained in these planes, and the treatment of a symmetry plane in the flow past a swept wing would be difficult in this coordinate system. In the analysis presented here, the flow is analyzed in coordinate planes parallel to the free stream velocity vector, and the symmetry condition is applied on a single coordinate surface. To allow the use of a fine mesh to resolve the details of the flow in the sensitive region near the leading edge, the spanwise coordinate lines are aligned with the leading edge. Thus for wings of appreciable sweep, the resulting coordinate system is highly nonorthogonal.

The type of geometry we shall treat is illustrated in Figure 1. It consists of a wing of arbitrary planform and dihedral extending from a symmetry plane (or wall). We shall solve a finite difference approximation to the full potential equation for the transonic flow past such a configuration using a generalized relaxation method. The finite difference approximation is the rotated difference scheme introduced by Jameson [6], and is not in conservation form. This can introduce substantial errors in the treatment of flows containing strong shock waves. To assure the correct shock jump relations one ought either to introduce a shock fitting scheme or else to use a difference scheme in conservation form. A conservative formulation of the small disturbance equation has been given by Murman [8], and the exact potential flow equation has been solved in conservation form by Jameson [9] for flows past airfoils. Comparisons with experimental data show no clear cut advantage to using the

conservation form without a detailed modeling of the shock wave boundary layer interaction [10]. This is apparently because the error in the shock jump relations which results from the use of the nonconservative schemes is in the same sense as the effect of the boundary layer interaction. A three dimensional scheme in conservation form will be discussed in a later report.

## ANALYSIS

### Geometry

Accurate representation of the finite difference boundary conditions is much simplified if the boundary surfaces lie in coordinate planes. This is achieved in the present analysis by a sequence of transformations based upon a nearly conformal mapping of the physical space in planes containing the wing sections, taken in the streamwise direction. We begin by considering the physical space to be described in a Cartesian coordinate system for which  $x$ ,  $y$ , and  $z$  represent the streamwise, vertical, and spanwise directions, as shown in Figure 1. We then introduce an arbitrary singular line, just inside the leading edge of the profile at each spanwise station. This singular line will be the locus of branch points in subsequent transformations in each of the spanwise planes to unwrap the wing surface to a shallow bump; its location will be chosen to make the bump as smooth as possible. Representing the singular line as

$$x = x_s(z)$$

$$y = y_s(z)$$

we define

$$\bar{x} = x - x_s(z) ,$$

$$\bar{y} = y - y_s(z) , \quad (1)$$

$$\bar{z} = z .$$

This transformation shears out the wing sweep and dihedral, and puts the singular line at the origin of each  $\bar{x}, \bar{y}$  plane. In each

of these planes we introduce the conformal mapping

$$(X_1 + iY_1)^2 = 2(\bar{x} + i\bar{y}) , \quad (2)$$

which maps the entire wing surface to a shallow bump near the plane  $Y_1 = 0$ . If we define the height of this bump as

$$Y_1 = S(X, \bar{z}) ,$$

then the final shearing transformation

$$\begin{aligned} X &= X_1 , \\ Y &= Y_1 - S(X, \bar{z}) , \\ Z &= \bar{z} , \end{aligned} \quad (3)$$

reduces the wing surface to a portion of the plane  $Y = 0$ .

To render the computational domain finite, stretching transformations are introduced. For example,

$$Y = \frac{b\bar{Y}}{(1 - \bar{Y}^2)^a} , \quad 0 \leq a \leq 1 , \quad (4)$$

is used to map the planes  $Y = \pm \infty$  to  $\bar{Y} = \pm 1$ . Similar transformations are used outboard of the wing tip in the  $Z$  direction, and downstream of the trailing edge in the  $X$  direction. A sketch of the resulting rectangular computational domain is shown in Figure 2.

To avoid discontinuities at the wing trailing edge, the branch cut in each spanwise plane is continued smoothly downstream. In the physical plane, the continuation is represented

by

$$\bar{y} = \bar{y}_{te} + \tau(\bar{x}_{te} - \bar{x}^*) \frac{\ln \left( \frac{\bar{x} - \bar{x}^*}{\bar{x}_{te} - \bar{x}^*} \right)}{\left( \frac{\bar{x} - \bar{x}^*}{\bar{x}_{te} - \bar{x}^*} \right)} \quad (5)$$

where  $\tau$  is the mean of the upper and lower surface slopes at the trailing edge,  $\bar{x}_{te}$ ,  $\bar{y}_{te}$  are the trailing edge coordinates, and  $\bar{x}^*$  is a suitably chosen scaling constant (usually taken as the ordinate of the local quarter-chord point). In the solution, this cut is taken as the location of the vortex sheet, across which special difference formulas must be applied. Thus we make the approximation that the vortex sheet lies in a fixed surface near the plane of the wing which leaves the trailing edge smoothly according to the above formula.

### Equation of Motion

In the absence of strong shock waves, the steady, inviscid motion of a compressible fluid is well approximated by the well-known equation for the velocity potential  $\phi$ :

$$(a^2 - u^2)\phi_{xx} + (a^2 - v^2)\phi_{yy} + (a^2 - w^2)\phi_{zz} - 2uv\phi_{xy} - 2uw\phi_{xz} - 2vw\phi_{yz} = 0, \quad (6)$$

where  $u$ ,  $v$ , and  $w$  are the velocity components (i.e., the derivatives of  $\phi$ ) in the  $x$ ,  $y$ , and  $z$  directions, and  $a$  is the speed of sound. For the steady, potential flow of a perfect gas with specific heat ratio  $\gamma$ ,

$$a^2 = a_0^2 - \frac{\gamma - 1}{2} (u^2 + v^2 + w^2), \quad (7)$$

where  $a_0$  is the stagnation speed of sound. If the flow is uniform at infinity, parallel to the  $x$ - $y$  plane, and inclined at an angle  $\alpha$

to the x-axis, the far field singularity can be removed by defining the reduced potential  $G$  as

$$\begin{aligned} G &= \phi - x \cos \alpha - y \sin \alpha \\ &= \phi - \left\{ \frac{1}{2}(X_1^2 - Y_1^2) + x_s(z) \right\} \cos \alpha - \left\{ X_1 Y_1 + y_s(z) \right\} \sin \alpha. \end{aligned} \quad (8)$$

The transformations of equations (1), (2), and (3) applied to equation (6) then result in an equation of the form

$$A G_{XX} + B G_{YY} + C G_{ZZ} + D G_{XY} + E G_{XZ} + F G_{YZ} + R = 0. \quad (9)$$

If we introduce the notation

$$\begin{aligned} \xi &= -X_{1\bar{x}} x'_s - X_{1\bar{y}} y'_s, \\ \eta &= X_{1\bar{y}} x'_s - X_{1\bar{x}} y'_s, \end{aligned} \quad (10)$$

$$\begin{aligned} U &= \frac{1}{h} \phi_{X_1} = \frac{1}{h} \left\{ X_1 \cos \alpha + Y_1 \sin \alpha + G_X - S_X G_Y \right\}, \\ V &= \frac{1}{h} \phi_{Y_1} = \frac{1}{h} \left\{ -Y_1 \cos \alpha + X_1 \sin \alpha + G_Y \right\}, \\ w &= \phi_z = h\xi U + h\eta V + x'_s \cos \alpha + y'_s \sin \alpha + G_z - S_z G_Y, \end{aligned} \quad (11)$$

and

$$\begin{aligned} \bar{U} &= U + h\xi w, \\ \bar{V} &= V + h\eta w, \end{aligned} \quad (12)$$

where

$$h^2 = \left| \frac{d(\bar{x} + i\bar{y})}{d(X_1 + iY_1)} \right|^2 = X_1^2 + Y_1^2, \quad (13)$$

then the coefficients in equation (9) can be written as

$$\begin{aligned}
 A &= a^2 \left\{ 1 + h^2 \xi^2 \right\} - \bar{U}^2 \\
 B &= \left\{ a^2 (1 + h^2 \xi^2) - \bar{U}^2 \right\} S_X^2 + \left\{ a^2 (1 + h^2 \eta^2) - \bar{V}^2 \right\} \\
 &\quad + h^2 (a^2 - w^2) S_Z^2 - \left\{ 2h^2 a^2 \xi \eta - 2\bar{U}\bar{V} \right\} S_X \\
 &\quad + \left\{ 2h^2 \xi a^2 - 2hw\bar{U} \right\} S_X S_Z - \left\{ 2h^2 \eta a^2 - 2hw\bar{V} \right\} S_Z , \\
 C &= h^2 \left\{ a^2 - w^2 \right\} , \\
 D &= - 2 \left\{ a^2 (1 + h^2 \xi^2) - \bar{U}^2 \right\} S_X + \left\{ 2h^2 \xi \eta a^2 - 2\bar{U}\bar{V} \right\} - \left\{ 2h^2 \xi a^2 - 2hw\bar{U} \right\} S_Z , \\
 E &= 2h^2 \xi a^2 - 2hw\bar{U} , \\
 F &= -2h^2 (a^2 - w^2) S_Z - \left\{ 2h^2 \xi a^2 - 2hw\bar{U} \right\} S_X + 2h^2 \eta a^2 - 2hw\bar{V} , \\
 R &= \left\{ - \left\{ a^2 (1 + h^2 \xi^2) - \bar{U}^2 \right\} S_{XX} - h^2 (a^2 - w^2) S_{ZZ} - \left\{ 2h^2 \xi a^2 - 2hw\bar{U} \right\} S_{XZ} \right\} G_Y \\
 &\quad + h^3 (a^2 - w^2) \left\{ \left\{ (x_s'^2 - y_s'^2) X_{1_{\bar{x}\bar{x}}} + 2x_s' y_s' X_{1_{\bar{x}\bar{y}}} - x_s'' X_{1_{\bar{x}}} - y_s'' X_{1_{\bar{y}}} \right\} U \right. \\
 &\quad \left. + \left\{ -(x_s'^2 - y_s'^2) X_{1_{\bar{x}\bar{y}}} + 2x_s' y_s' X_{1_{\bar{x}\bar{x}}} + x_s'' X_{1_{\bar{y}}} - y_s'' X_{1_{\bar{x}}} \right\} V \right\} \\
 &\quad + 2h^4 w \left\{ (X_{1_{\bar{x}}} x_s' - X_{1_{\bar{y}}} y_s') X_{1_{\bar{x}\bar{x}}} + (X_{1_{\bar{y}}} x_s' + X_{1_{\bar{x}}} y_s') X_{1_{\bar{x}\bar{y}}} \right\} (U^2 + V^2) \\
 &\quad + \frac{1}{h} \left\{ X_1 U + Y_1 V \right\} (U^2 + V^2) + \cos \alpha \left\{ h^2 (\xi^2 - \eta^2) a^2 - \bar{U}^2 - \bar{V}^2 + h^2 (a^2 - w^2) x_s'' \right\} \\
 &\quad + \sin \alpha \left\{ 2h^2 \xi \eta a^2 - 2\bar{U}\bar{V} + h^2 (a^2 - w^2) y_s'' \right\} .
 \end{aligned} \tag{14}$$

Note that for the transformation defined by equation (2),

$$\begin{aligned}
 X_{1_{\bar{x}}} &= X_1 / h^2 , \\
 Y_{1_{\bar{y}}} &= Y_1 / h^2 ,
 \end{aligned} \tag{15}$$

and

$$\begin{aligned} X_{1\bar{x}\bar{x}} &= -\frac{X_1}{h^6} (h^2 - 4Y_1^2) , \\ X_{1\bar{x}\bar{y}} &= \frac{Y_1}{h^6} (h^2 - 4X_1^2) . \end{aligned} \quad (16)$$

The symmetry condition that  $w = 0$  on the plane  $z = 0$  requires

$$G_Z + \xi G_X - \{S_Z + \xi S_X - \eta\} G_Y = 0 , \quad (17)$$

and the boundary condition that the flow be tangent to the wing surface requires

$$\begin{aligned} &\left\{ \frac{1}{h^2} (1 + S_X^2) + \{S_Z + \xi S_X - \eta\}^2 \right\} G_Y \\ &+ \left\{ -\frac{1}{h^2} S_X + \xi \{-S_Z - \xi S_X + \eta\} \right\} G_X + \left\{ -S_Z - \xi S_X + \eta \right\} G_Z \\ &+ \left\{ -X_{1\bar{x}} \cos \alpha - X_{1\bar{y}} \sin \alpha \right\} S_X - X_{1\bar{y}} \cos \alpha + X_{1\bar{x}} \sin \alpha = 0 , \end{aligned} \quad (18)$$

on  $Y = 0$ .

Downstream of a finite lifting wing there will be a vortex sheet. Across the sheet the pressure is continuous, but there may be discontinuities in the tangential velocity components. Convection and roll-up of the vortex sheet are ignored. In reality, the component of velocity normal to the sheet must be zero, but in our approximation it is simply required to be continuous. Thus, the equation

$$\phi_{YY} = 0$$

is used at points lying on the vortex sheet. Also the disconti-



nunity in potential is assumed to be constant along streamwise coordinate lines downstream of the trailing edge. The value of this discontinuity is determined by the Kutta condition, and its spanwise variation determines the strength of the vortex sheet.

### Finite Difference Approximation

The success of the type dependent difference scheme applied to the transonic small disturbance equation by Murman and Cole [1] can be attributed to the fact that it effectively adds a directional bias to the equation at points where the local flow is supersonic. In constructing an analogous scheme for the full potential equation in general curvilinear coordinates (which may not be aligned, even approximately, with the local flow direction), care must be taken to ensure that this bias is added in the upwind direction, i.e., in the direction parallel to the velocity vector.

A method with this property has been proposed by Jameson [6]. To illustrate it, we return to the potential equation in the physical coordinates. The equation is rearranged as if it were expressed in a Cartesian coordinate system aligned with the local flow direction,  $s$ , at the point under consideration. Then equation (6) assumes the canonical form

$$(a^2 - q^2)\phi_{ss} + a^2(\nabla^2\phi - \phi_{ss}) = 0 \quad (19)$$

where  $q$  is the magnitude of the velocity.

The relaxation scheme is designed to simulate an artificial time dependent process which converges to the desired solution of the steady state equation. In the finite difference approximation to the potential equation, central differences are used to calculate all first derivatives, from which the velocities can be determined using equations (11). At grid points where the flow is subsonic, central differences are also used to approximate the second-order derivatives in equation (9). A typical central difference formula for  $G_{XX}$  is

$$G_{XX} = \frac{G_{i-1,j,k}^{(n+1)} - \left(\frac{2}{\omega}\right) G_{i,j,k}^{(n+1)} - 2\left(1 - \frac{1}{\omega}\right) G_{i,j,k}^{(n)} + G_{i+1,j,k}^{(n)}}{\Delta X^2}, \quad (20)$$

where the superscripts denote the iteration level and  $\omega$  is the relaxation factor [6]. If we regard each iteration as representing an advance  $\Delta t$  in an artificial time coordinate, this formula can be interpreted as an approximation to

$$G_{XX} - \frac{\Delta t}{\Delta X} \left\{ G_{Xt} + \frac{1}{\Delta X} \left( \frac{2}{\omega} - 1 \right) G_t \right\}$$

Similarly, the formula

$$G_{XY} = \frac{G_{i+1,j+1,k}^{(n)} - G_{i+1,j-1,k}^{(n)} - G_{i-1,j+1,k}^{(n+1)} + G_{i-1,j-1,k}^{(n+1)}}{4\Delta X\Delta Y} \quad (21)$$

can be interpreted as an approximation to

$$G_{XY} - \frac{1}{2} \frac{\Delta t}{\Delta X} G_{Yt}.$$

The relaxation process can thus be regarded as an approximation to the time dependent equation

$$(M^2-1)G_{ss} - G_{mm} - G_{nn} + 2\alpha_1 G_{st} + 2\alpha_2 G_{mt} + 2\alpha_3 G_{nt} + \delta G_t = Q, \quad (22)$$

where  $M = q/a$  is the local Mach number,  $m$  and  $n$  are suitably scaled coordinates in the plane normal to the velocity vector, and  $Q$  contains all the terms in the equation other than the principal part. The coefficients  $\alpha_1, \alpha_2, \alpha_3$ , and  $\delta$ , depend on the mix of old and updated values in the difference equations as well as any explicit time-like or mixed terms that have been added for stability.

Introducing the new time coordinate

$$T = t - \frac{\alpha_1}{M^2-1} s + \alpha_2 m + \alpha_3 n,$$

transforms equation (22) to

$$(M^2-1)G_{ss} - G_{mm} - G_{nn} - \left\{ \frac{\alpha_1^2}{M^2-1} - \alpha_2^2 - \alpha_3^2 \right\} G_{TT} + \delta G_T = Q. \quad (23)$$

In order to ensure the convergence of the scheme, we require that equation (23) should be a damped three-dimensional wave equation. This will be the case if

$$\alpha_1^2 > (M^2-1)(\alpha_2^2 + \alpha_3^2). \quad (24)$$

At points where the velocity is supersonic, upwind differences are used to represent contributions to  $G_{ss}$  in the first term of equation (19). This is done using formulas of the type

$$G_{XX} = \frac{2G_{i,j,k}^{(n+1)} - G_{i,j,k}^{(n)} - 2G_{i-1,j,k}^{(n+1)} + G_{i-2,j,k}^{(n)}}{\Delta X^2}, \quad (25)$$

$$G_{XY} = \frac{G_{i,j,k}^{(n+1)} - G_{i-1,j,k}^{(n+1)} - G_{i,j-1,k}^{(n+1)} + G_{i-1,j-1,k}^{(n+1)}}{\Delta X \Delta Y}.$$

These formulas also have the property of guaranteeing diagonal dominance for the updated values on each line. The formula for  $G_{XX}$  can be interpreted as representing

$$G_{XX} + 2 \frac{\Delta t}{\Delta X} G_{Xt}$$

Together with analogous formulas for  $G_{YY}$  and  $G_{ZZ}$ , this introduces a term equal to

$$2(M^2-1)G_{st}$$

into equation (22). To ensure that equation (24) is satisfied at points near the sonic line where  $(M^2-1)$  is small, the coefficient of  $G_{st}$  can be further augmented by adding a term of the form

$$\beta \frac{\Delta t}{\Delta X} \left\{ uG_{Xt} + vG_{Yt} + h^2 wG_{Zt} \right\}, \quad (26)$$

where  $\beta > 0$  is appropriately chosen. The required mixed derivatives can be constructed in the form

$$\frac{\Delta t}{\Delta X} G_{Xt} = \frac{G_{i,j,k}^{(n+1)} - G_{i,j,k}^{(n)} - G_{i-1,j,k}^{(n+1)} + G_{i-1,j,k}^{(n)}}{\Delta X^2}. \quad (27)$$

The supersonic difference scheme is completed by using central difference formulas similar to equations (20) and (21) to evaluate contributions to the second term of equation (19), but with  $w$  set to unity, as suggested by a local von Neumann test [6].

## Boundary Conditions

The boundary condition at infinity is particularly simple because the square root transformation reduces the entire vortex wake to the X-Z plane at downstream infinity. Therefore, since the uniform stream singularity has been removed by the introduction of the reduced potential, the Dirichlet condition

$$G = 0$$

is appropriate.

On the X-Y and X-Z planes, finite difference approximations to the Neumann boundary conditions specified by equations (17) and (18) must be applied to those portions representing solid boundaries (i.e., the symmetry plane and the wing surface). At the wing surface, central difference approximations are used in equation (18) to define values of the reduced potential at image points located one mesh spacing below the X-Z plane. A similar method is used on the symmetry plane, but due to the high degree of nonorthogonality of the coordinate system when the wing is highly swept, simple central differences become unstable. Thus, to set the potential values at the image points for the symmetry plane, the X-differences required in equation (17) are evaluated by averaging one-sided differences on either side of the symmetry plane, taken in the upwind direction in the image plane, and in the downwind direction in the first plane in the flow region. The symmetry condition thus remains formally second order accurate, and the incorporation of the image point whose value is being set into the X-difference adds to the stability of the scheme. This method of handling the symmetry condition has proved stable for

sweep angles in excess of 35 degrees.

At points on the X-Z plane which do not lie on the wing surface, the values of the reduced potential at the image points are taken to be those of the associated point on the other side of the branch cut, allowing for a discontinuity across the vortex sheet. The value of this discontinuity is taken to be independent of X at each spanwise station, and its value is determined by the Kutta condition that the flow leave the trailing edge smoothly.

One final note concerns points which lie on the continuation of the singular line outboard of the wing tip. At these points the mapping is singular, and a special limiting form of the difference equations must be used. At points where the solution is regular, the nonlinear terms of the potential equation are of  $O(1/h)$ , while the Laplacian transforms to

$$\frac{1}{h^2} (\phi_{X_1 X_1} + \phi_{Y_1 Y_1}) + \phi_{ZZ} .$$

Thus, in the limit as h tends to zero,

$$\phi_{X_1 X_1} + \phi_{Y_1 Y_1} = 0 \tag{28}$$

is a suitable limiting form.

## RESULTS

### Computational procedure

The potential formulation is particularly attractive for three-dimensional calculations because it requires the storage of only one quantity at each grid point, and the number of grid points required to accurately describe these flow fields is large. Even so, it is impractical to store the entire solution array in the high speed core of many current computing machines. Fortunately, since the analysis presented here depends on a relaxation solution of the difference equations, it is not necessary to have the entire solution immediately available at all times. It is, therefore, stored on a disk file, and read into core one X-Y plane at a time. At any time during the solution procedure, the values of the potential on four such planes are in the core. Old values are buffered in and new values buffered out of core while other calculations are being performed as much as possible, to keep the process efficient.

In each X-Y plane, the equations are solved by successive line overrelaxation. The plane is divided into three regions, as shown in Figure 3. In the central region the equations are relaxed along horizontal lines, sweeping from infinity to the wing surface. In the outer regions the equations are relaxed along vertical lines, sweeping away from the central region to infinity. Such a sweep pattern ensures that the sweep direction will not be opposed to the flow direction in any supersonic zones,

which would result in instability. In many cases, the central region can be taken to cover the entire plane; that is, only horizontal line relaxation is used.

To speed convergence, an initial calculation is usually performed on a coarse grid, typically containing  $48 \times 6 \times 8$  grid cells in the X, Y, and Z directions respectively. This solution is then interpolated onto a finer grid containing twice as many mesh cells in each direction, and is used as a starting guess for an intermediate solution. The process is repeated once again to give the final solution on a grid containing  $192 \times 24 \times 32$  mesh cells. A typical run consists of 100 relaxation sweeps on each grid, requiring a total of approximately 85 minutes of CPU time on a CDC 6600. The same program has been run on the CDC 7600, for which a similar calculation requires about 15 minutes.

### Examples

In this section we present the results of calculations using the swept wing program, and compare the predicted surface pressure distributions with those measured in experiments. The comparisons are made for two different wings, each typical of a class of swept wings of the subsonic transport type.

The first wing geometry is representative of the tip panel of a relatively simple wing of conventional high speed section shape. It has a uniform section of 9.8 percent thickness ratio,



and the planform has a leading edge sweep angle of  $30^\circ$ , a taper ratio of 0.7, and an aspect ratio of 3.8. A program generated projection drawing of the wing is shown in Figure 4. The wing was tested by Monnerie and Charpin [11] of the ONERA, and carries their designation of wing M-6.

The first results presented are at a free stream Mach number of 0.9226 and zero angle of attack, resulting in zero lift for this symmetrical wing. Figure 5 compares the calculated and measured streamwise surface pressure distributions at the 20, 45, 65, and 95 percent semispan locations [11,12]. Agreement is quite good, including the predicted shock location.

Figure 6 shows similar results for the same wing at a Mach number of 0.919 and an angle of attack of 3.07 degrees. Again, agreement between the computed and experimental results is quite good, with the exception of the shock location on the lower surface, which is somewhat further aft than predicted by the calculation.

Figure 7 shows a program generated, three-dimensional, projection view of the wing surface pressure distribution at a Mach number of 0.840 and an angle of attack of 3.06 degrees. This is a particularly interesting case because of the merging of two shocks into one on the wing upper surface as one proceeds outboard. This pattern is graphically illustrated in the projection view. Figure 8 shows comparisons of the calculated results with experimental data, again at the 20, 45, 65, and 95 percent semispan stations. Agreement is quite good, including the

prediction of the double-shock pattern at the inboard stations.

Figure 9 shows the projection view of the wing surface pressure distribution at a Mach number of 0.837 and an angle of attack of 6.06 degrees. Again, the calculation predicts the merging of a double shock pattern inboard to a single shock further outboard. Comparisons with data, shown in Figure 10 show that agreement is still quite good.

The second geometry is representative of wings being considered for the next generation of subsonic transport aircraft. The wing is twisted, both aerodynamically and geometrically, is highly tapered, and has a discontinuity in trailing edge sweep angle at the 35 percent semispan location. The planform has a leading edge sweep angle of 35 degrees and an aspect ratio of 7. It has 5 degrees of dihedral. It is defined by four distinct streamwise sections (at the 12, 35, 70, and 100 percent semispan stations), with linearly interpolated coordinates between. The streamwise thickness ratio varies from 16.3 percent at the root to 11.9 percent at the tip. For the wind tunnel tests the wing was mounted on a quasicylindrical fuselage which extended to the 12 percent semispan. For the computations, the symmetry plane was assumed to be at the same spanwise station as the wing-fuselage intersection in the tests. A projection drawing of the wing (extended to the fuselage centerline) is shown in Figure 11. For these calculations, the wing geometry was modified to account for boundary layer effects by adding the displacement thickness obtained from two-dimensional boundary layer calculations

multiplied by an empirically determined spanwise weighting factor. The wing was one of several tested in a cooperative program by the Douglas Aircraft Company and the NASA Ames Research Center in the Ames 11-foot tunnel at a Reynolds number of approximately  $5 \times 10^6$ , based on the mean aerodynamic chord.

A program generated three-dimensional projection drawing of the upper and lower surface pressure distributions for this wing is shown in Figure 12. (This particular case was run with no correction for boundary layer displacement effect, and with the wing extended to the fuselage centerline.)

Comparisons with experimental data are shown in Figures 13 and 14. The first case, Figure 13, shows streamwise surface pressure distributions at a number of spanwise stations for a Mach number of 0.75 and an angle of attack of 2.2 degrees. Agreement with experiment is seen to be excellent, including the location and strength of the rather strong shock near the leading edge on the wing upper surface.

Figure 14 shows similar comparisons at a Mach number of 0.84 and an angle of attack of 1.85 degrees. Again, agreement is quite good, although the resolution of the first (rather weak) shock of the inboard double shock pattern seems lost between the 35.5 and 50 percent semispan locations.

The results displayed in Figures 13 and 14 were kindly supplied by R. M. Hicks and P. A. Henne. Further details of the wing geometry, calculations, and test conditions are contained in [13].

## CONCLUSIONS

A numerical method has been presented for determining the inviscid transonic flow past a swept wing. The method is based on a type-dependent, finite difference approximation to the full potential equation, solved in a computational domain designed for accurate application of the wing surface and symmetry plane boundary conditions. Calculated surface pressure distributions agree well with experimental data for wings of conventional and supercritical section shape (when the geometry in the latter cases is corrected for the displacement effect of the boundary layer).

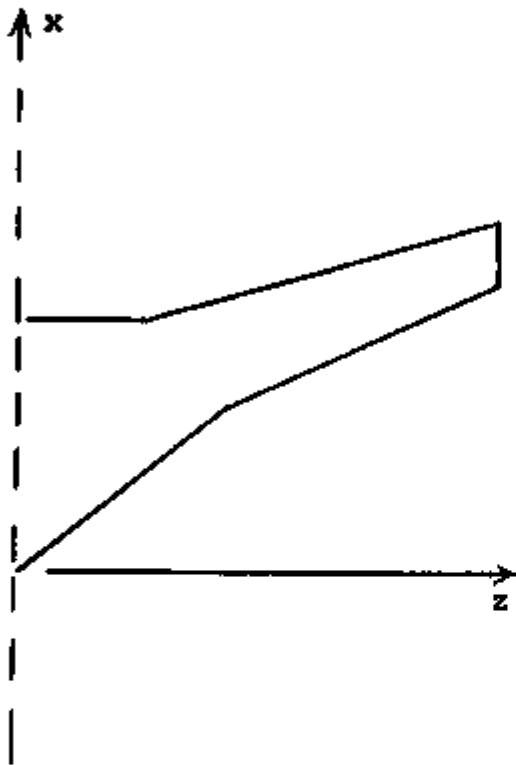
Mapping techniques similar to those used here could be used to treat more realistic geometries, e.g., a wing mounted on a fuselage [14]. The recasting of the finite difference approximation into conservation form would also be an important theoretical contribution.

Finally, as was mentioned in the preceding section, these calculations require a substantial amount of computer time. Thus, methods of accelerating the convergence of the iterative scheme are particularly important in three-dimensional problems. A number of techniques to achieve this have met with success in two-dimensional calculations, including a hybrid Poisson-solver/relaxation technique [15,16], a multi-grid method [17], and an alternating-direction method [18]. The extension of these methods to three-dimensional calculations should result in great savings.

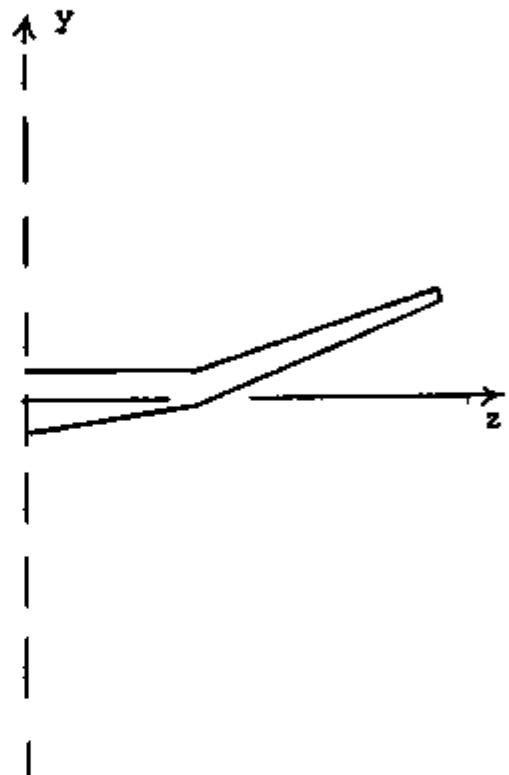
## REFERENCES

1. Murman, E. M., and Cole, J. D., "Calculation of plane, steady transonic flows," AIAA J. 9, (1971) 114.
2. Jameson, Antony, "Transonic flow calculations for airfoils and bodies of revolution," Grumman Aerodynamics Report 370-71-1, (December 1971).
3. Garabedian, P. R., and Korn, D. G., "Analysis of transonic airfoils," Comm. Pure Appl. Math. 24 (1972) 841.
4. Ballhaus, W. F. and Bailey, F. R., "Numerical calculation of transonic flow about swept wings," AIAA Paper 72-677 (1972).
5. Bailey, F. R. and Ballhaus, W. F., "Relaxation methods for transonic flow about wing-cylinder combinations and lifting, swept wings," Third International Conference on Numerical Methods in Fluid Dynamics, Paris, France, (July 3-7, 1972).
6. Jameson, Antony, "Iterative solution of transonic flows over airfoils and wings, including flows at Mach 1," Comm. Pure Appl. Math. 27 (1974) 283.
7. Jones, R. T., "Reduction of wave drag by antisymmetric arrangement of wings and bodies," AIAA J. 10 (1972) 171-176.
8. Murman, E. M., "Analysis of embedded shock waves calculated by relaxation methods," AIAA J. 12 (1974) 626-633.
9. Jameson, Antony, "Transonic potential flow calculations using conservation form," proceedings of AIAA 2<sup>nd</sup> Computational Fluid Dynamics Conf., (June 1975) 148-155.

10. Bauer, F. and Korn, D., "Computer simulation of transonic flows past airfoils with boundary layer correction," Proc. AIAA 2nd Computational Fluid Dynamics Conf. (6/1975) 184-189.
11. Monnerie, B. and Charpin, F., "Essais de Buffeting d'une aile en fleche en transsonique," 10<sup>e</sup> Colloque d'Aerodynamique Appliquee, Institut de Mechanique des Fluides, Lille (11/73).
12. van der Vooren, J., Private Communication.
13. Hicks, R. M., and Henne, P. A., NASA TMX, to appear.
14. Caughey, D. A., and Jameson, Antony, "Calculation of transonic potential flow fields about complex, three dimensional configurations," Project SQUID Workshop on Transonic Flow Problems in Turbomachinery, Monterey, California, (February 1976).
15. Jameson, Antony, "Accelerated iterative schemes for transonic flow calculations using fast Poisson solvers," New York Univ. ERDA Report COO-3077-82.
16. Caughey, D. A., and Jameson, Antony, "Accelerated iterative calculation of transonic nacelle flow fields," AIAA Paper 76-100, presented at AIAA 14<sup>th</sup> Aerospace Sciences Meeting, Washington, D. C., (January 26-28, 1976).
17. South, Jerry C., Jr., and Brandt, Achi, "Application of a multi-level grid method to transonic flow calculations," Project SQUID Workshop on Transonic Flow Problems in Turbomachinery, Monterey, California (February 1976).
18. Jameson, Antony, "An alternating-direction method for the solution of the transonic small disturbance equation," New York Univ. ERDA Report COO-3077-96.



(a) Plan View



(b) Front View

Figure 1. Geometry of Swept Wing.

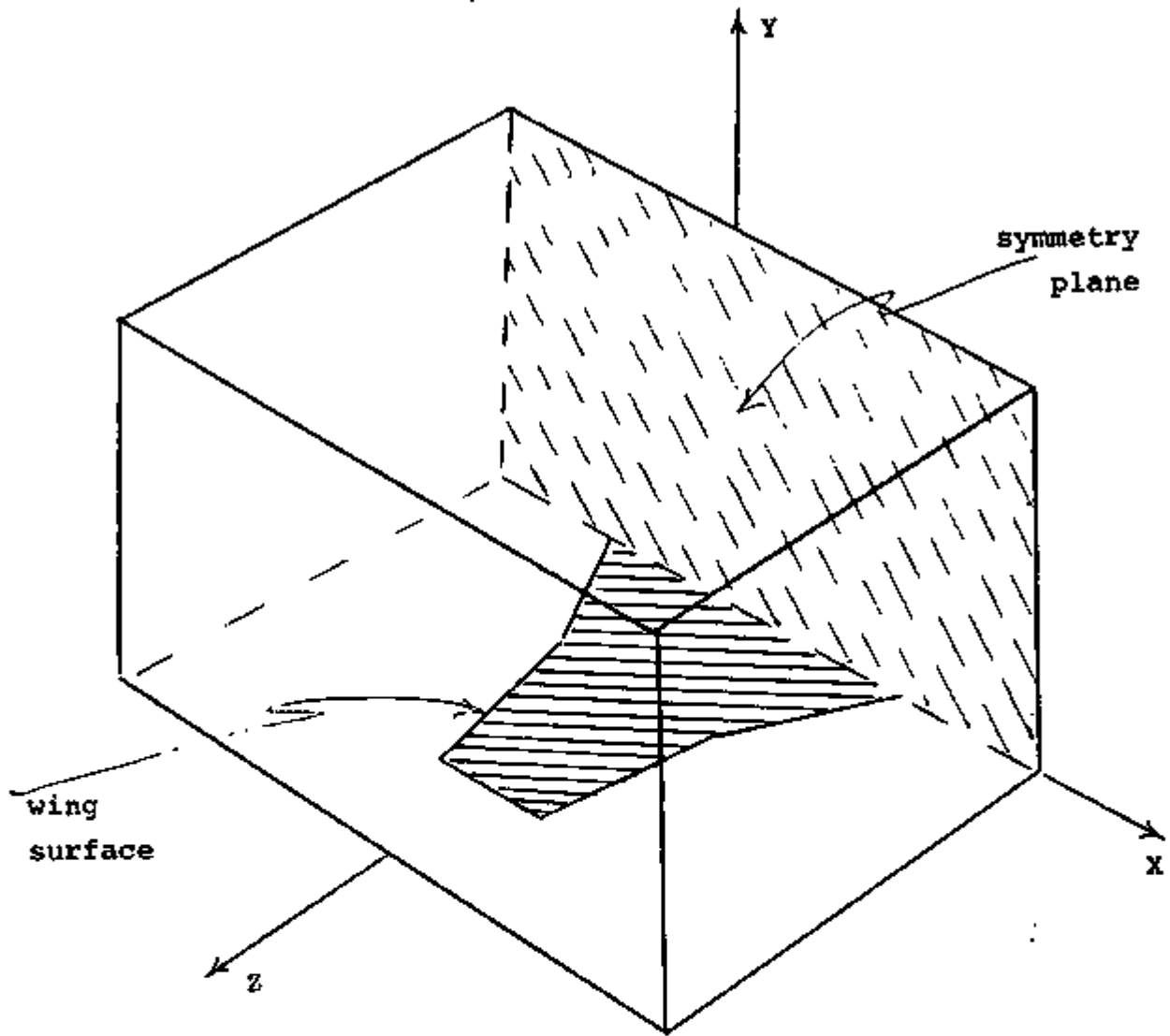


Figure 2. Sketch of Computational Domain.



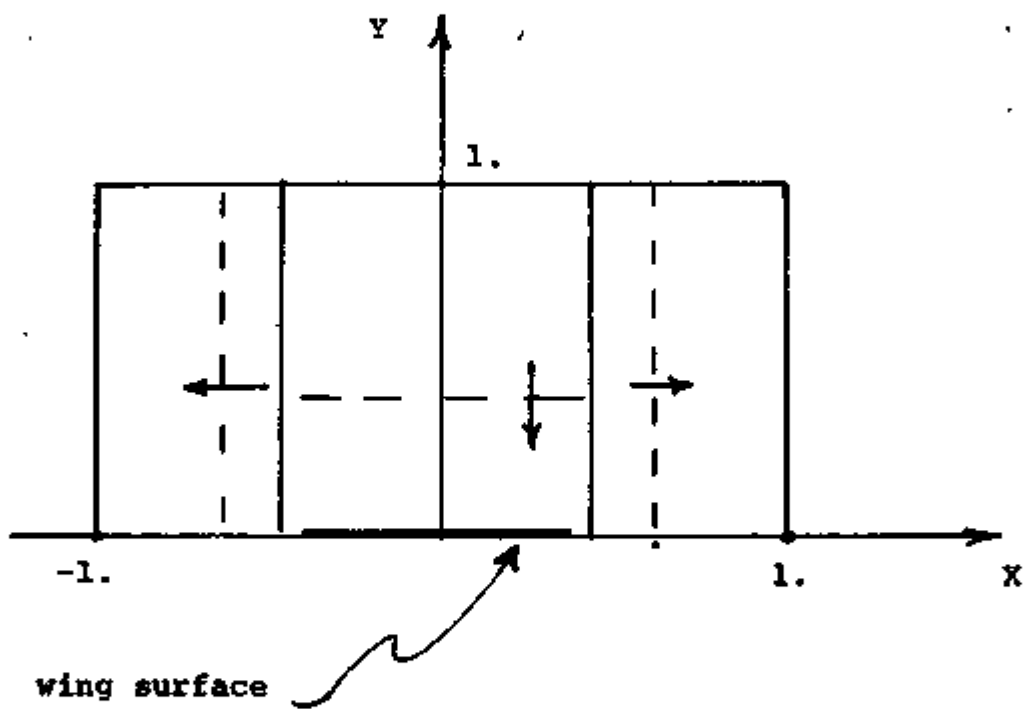


Figure 3. Sweep Directions in Computational Plane.

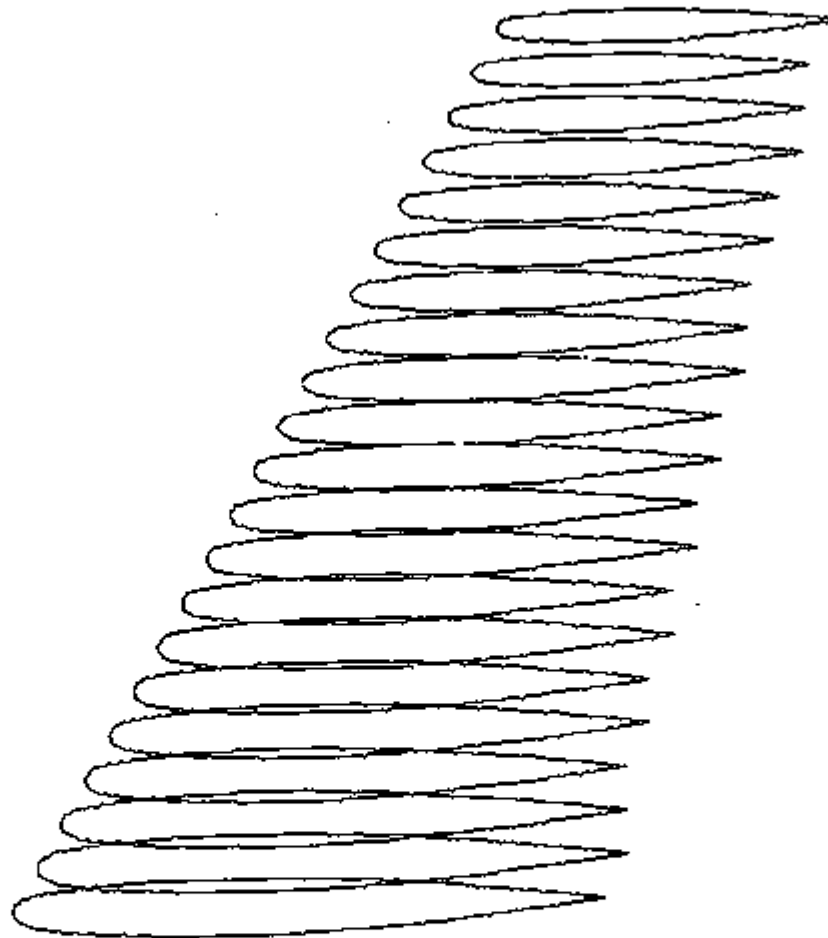


FIGURE 4. GEOMETRY OF ONERA WING.

VIEW OF WING

ONERA WING M6	L.E. SWEEP 30 DEG	ASPECT RATIO 3.8
MACH .923	YAW 0.000	ALPHA 0.000
L/D -.00	CL -.0000	CD .0246

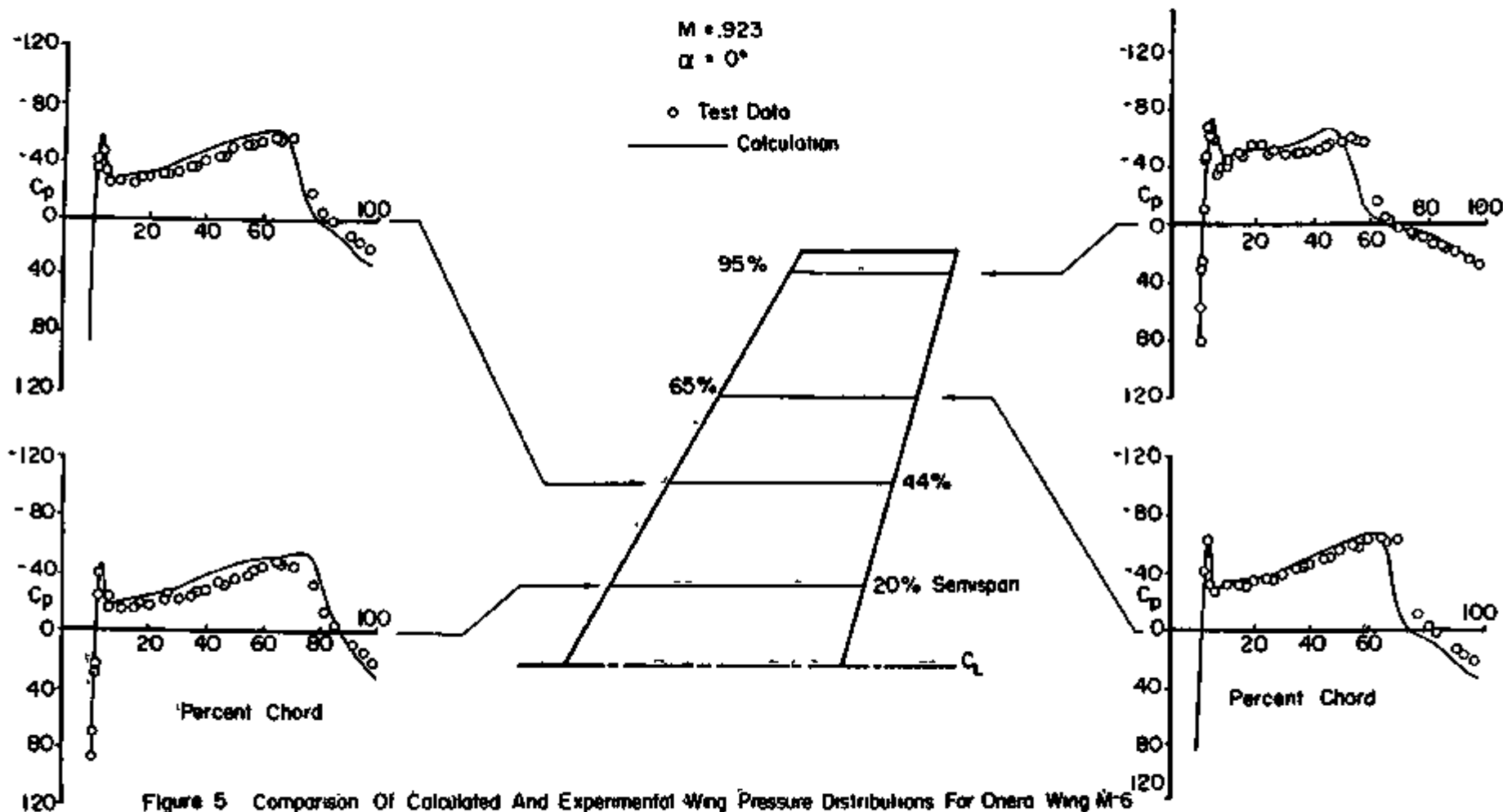


Figure 5 Comparison Of Calculated And Experimental Wing Pressure Distributions For Onera Wing M-6

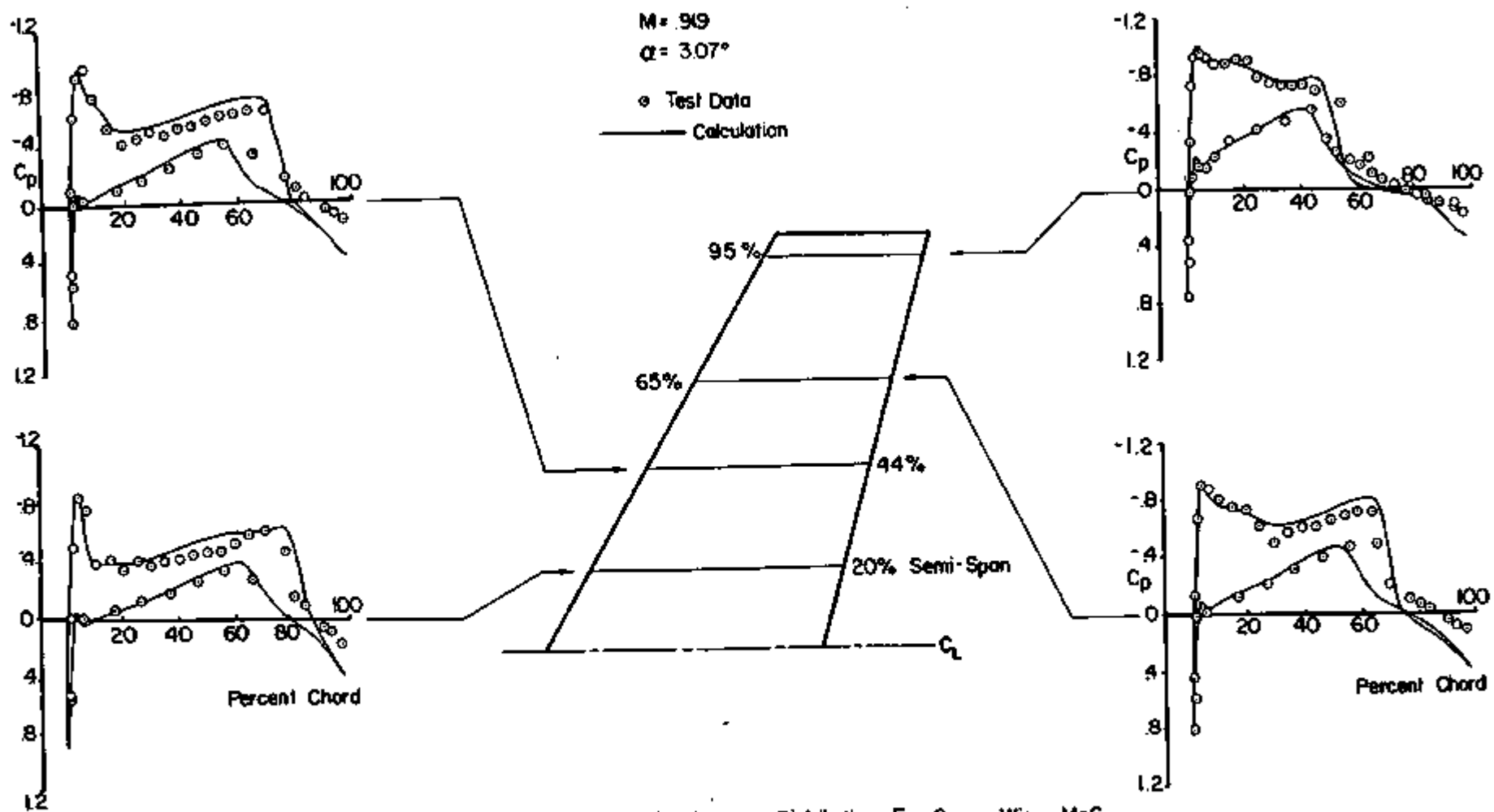


Figure 6. Comparison Of Calculated And Experimental Wing Pressure Distributions For Onera Wing M-6.

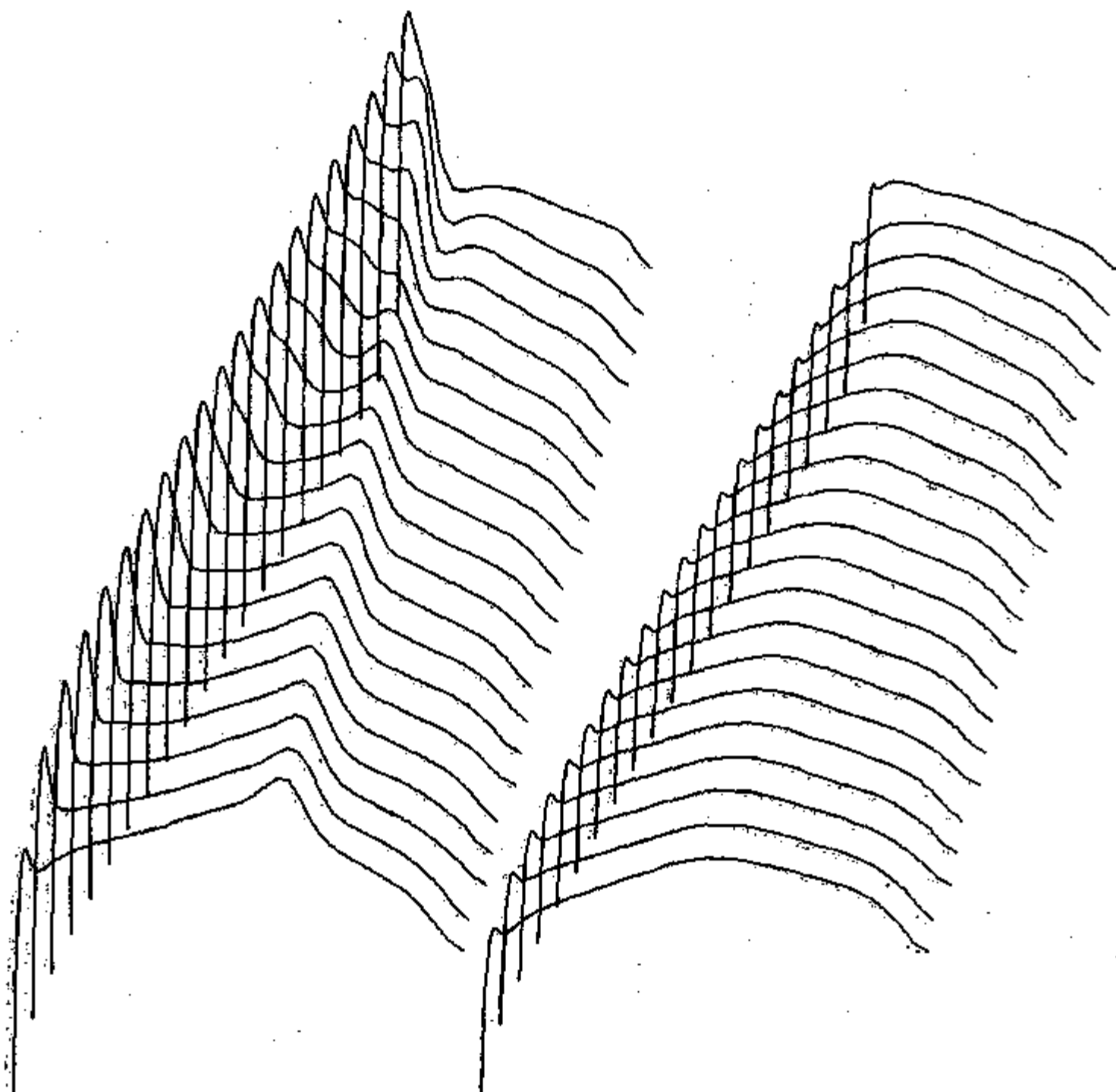


FIGURE 7

UPPER SURFACE PRESSURE

LOWER SURFACE PRESSURE

ONERA WING M6	L. E. SWEEP 30. DEG.	ASPECT RATIO 3.8
MACH .840	YAW 0.000	ALPHA 3.060
L/D 13.89	CL .2860	CD .0206

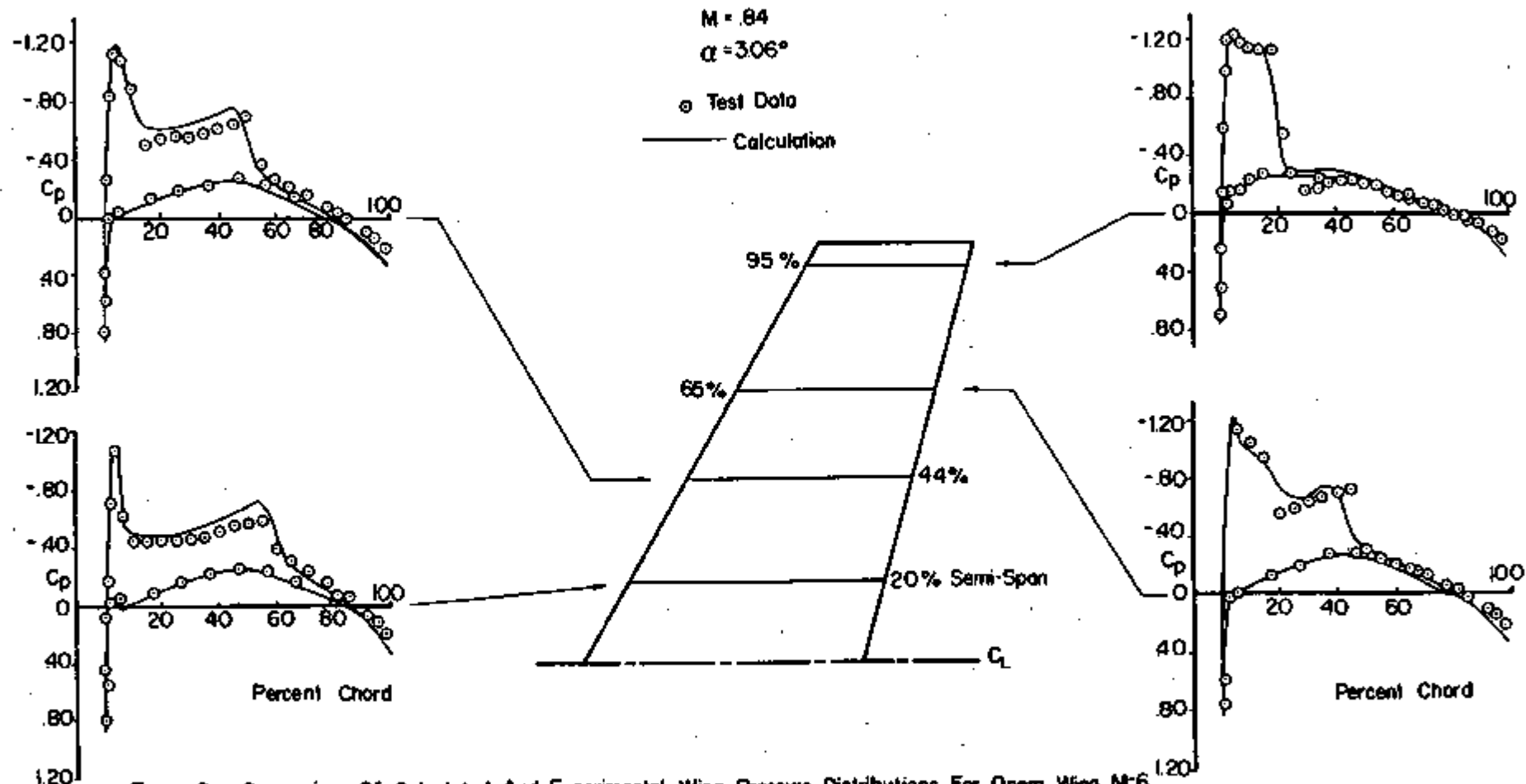


Figure 6. Comparison Of Calculated And Experimental Wing Pressure Distributions For Onera Wing N°6.

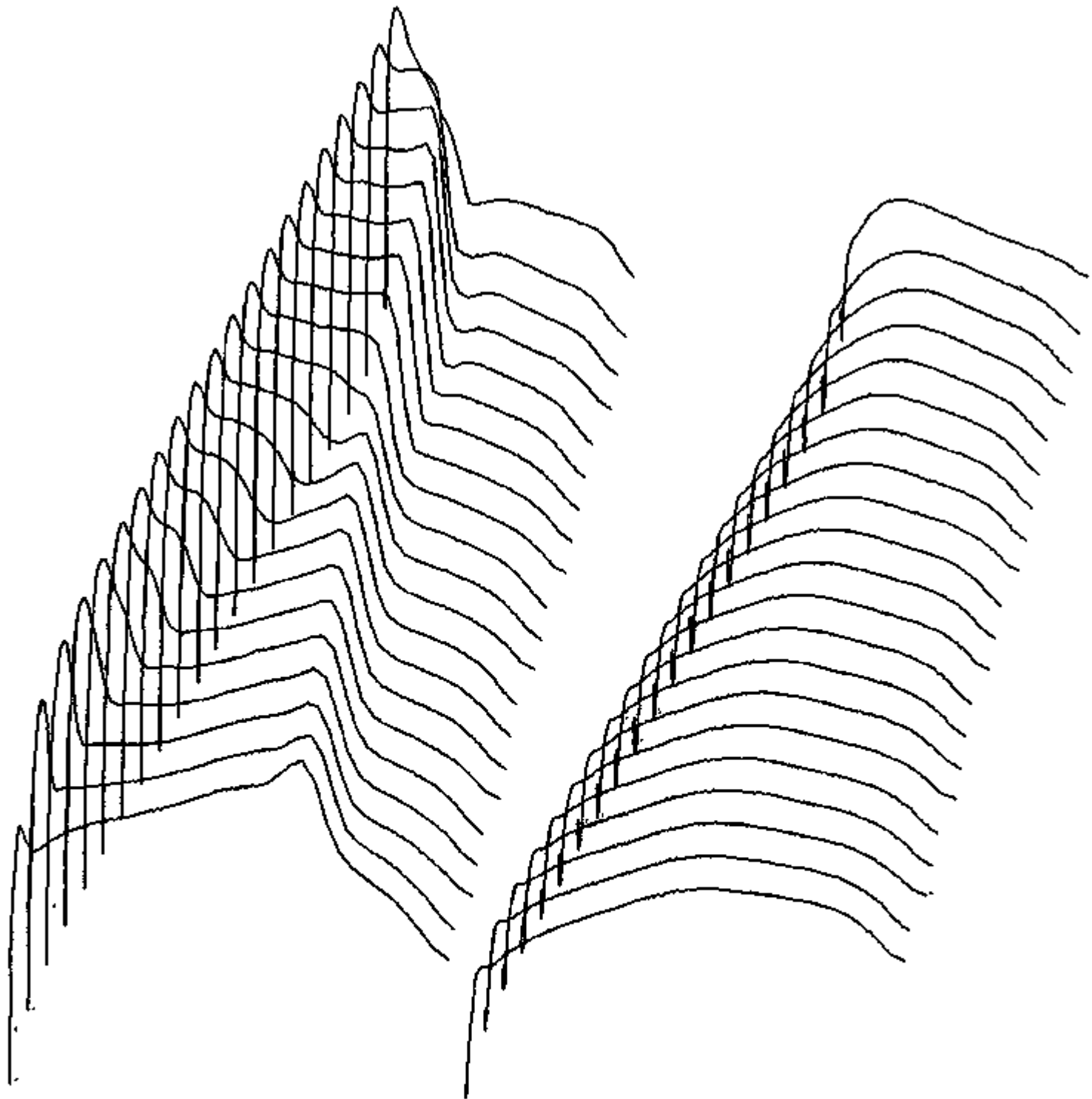


FIGURE 9

UPPER SURFACE PRESSURE

LOWER SURFACE PRESSURE

ONERA WING M6	L. E. SWEEP 30. DEG.	ASPECT RATIO 3.8
MACH .837	YAW 0.000	ALPHA 6.060
L/D 9.61	CL .5587	CD .0581

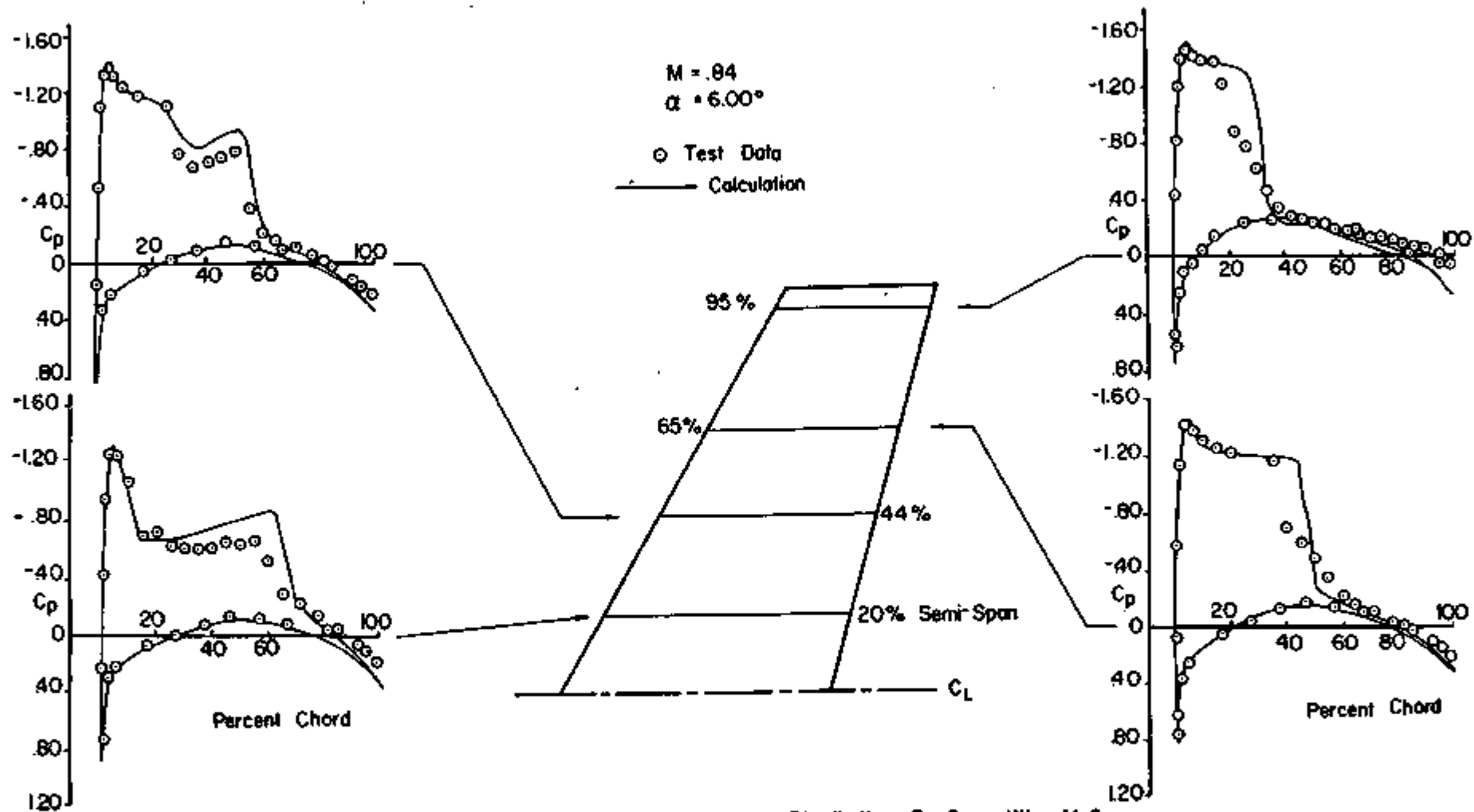


Figure 10 Comparison Of Calculated And Experimental Wing Pressure Distributions For Onera Wing M-6.



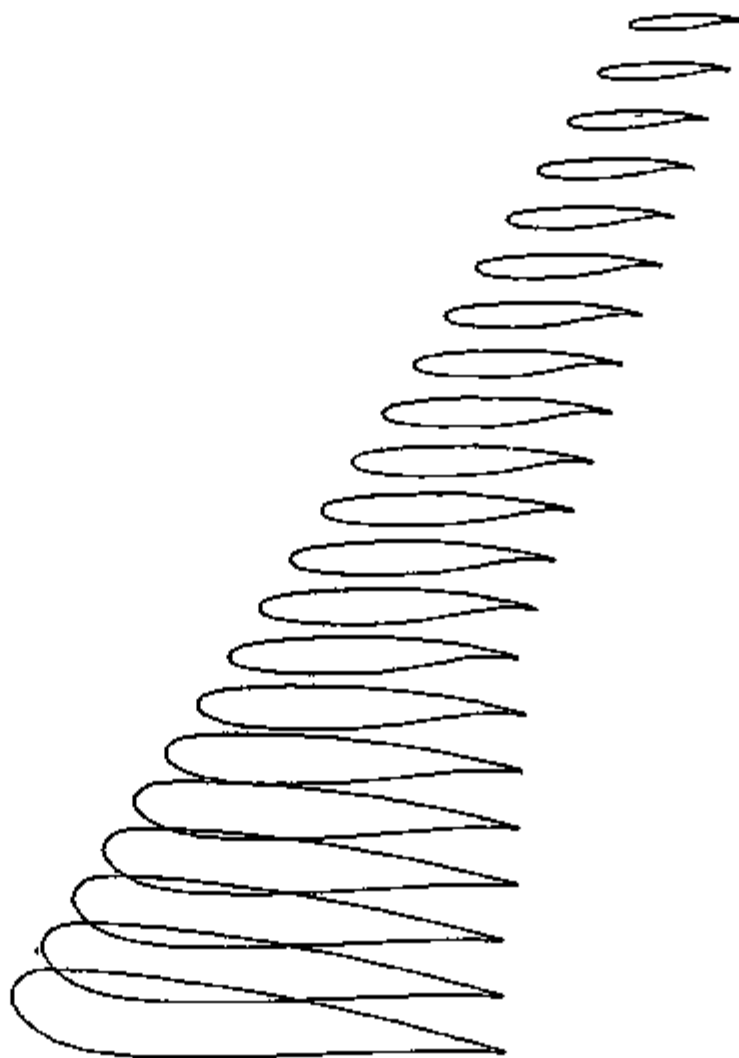


FIGURE 11. GEOMETRY OF DOUGLAS WING.

VIEW OF WING

DOUGLAS WING W2 (EXTENDED TO CENTER LINE)					
MACH	.819	YAW	0.000	ALPHA	0.000
L/D	20.09	CL	.5455	CD	.0272

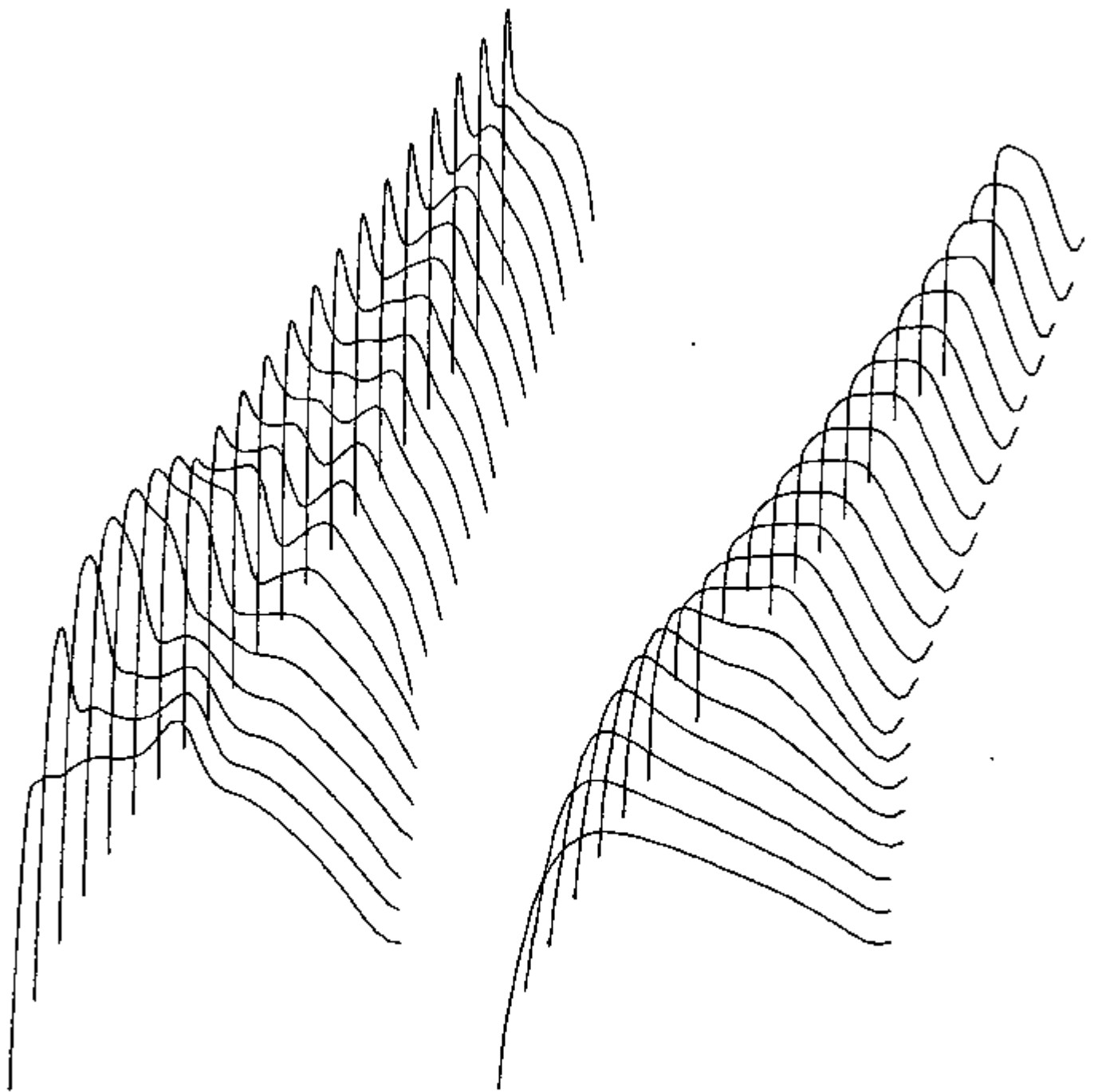


FIGURE 12. THREE-DIMENSIONAL SURFACE PRESSURE DISTRIBUTION.

UPPER SURFACE PRESSURE

LOWER SURFACE PRESSURE

DOUGLAS WING W2 (EXTENDED TO CENTER LINE)

MACH	.819	YAW	0.000	ALPHA	0.000
L/D	20.09	CL	.5455	CD	.0272

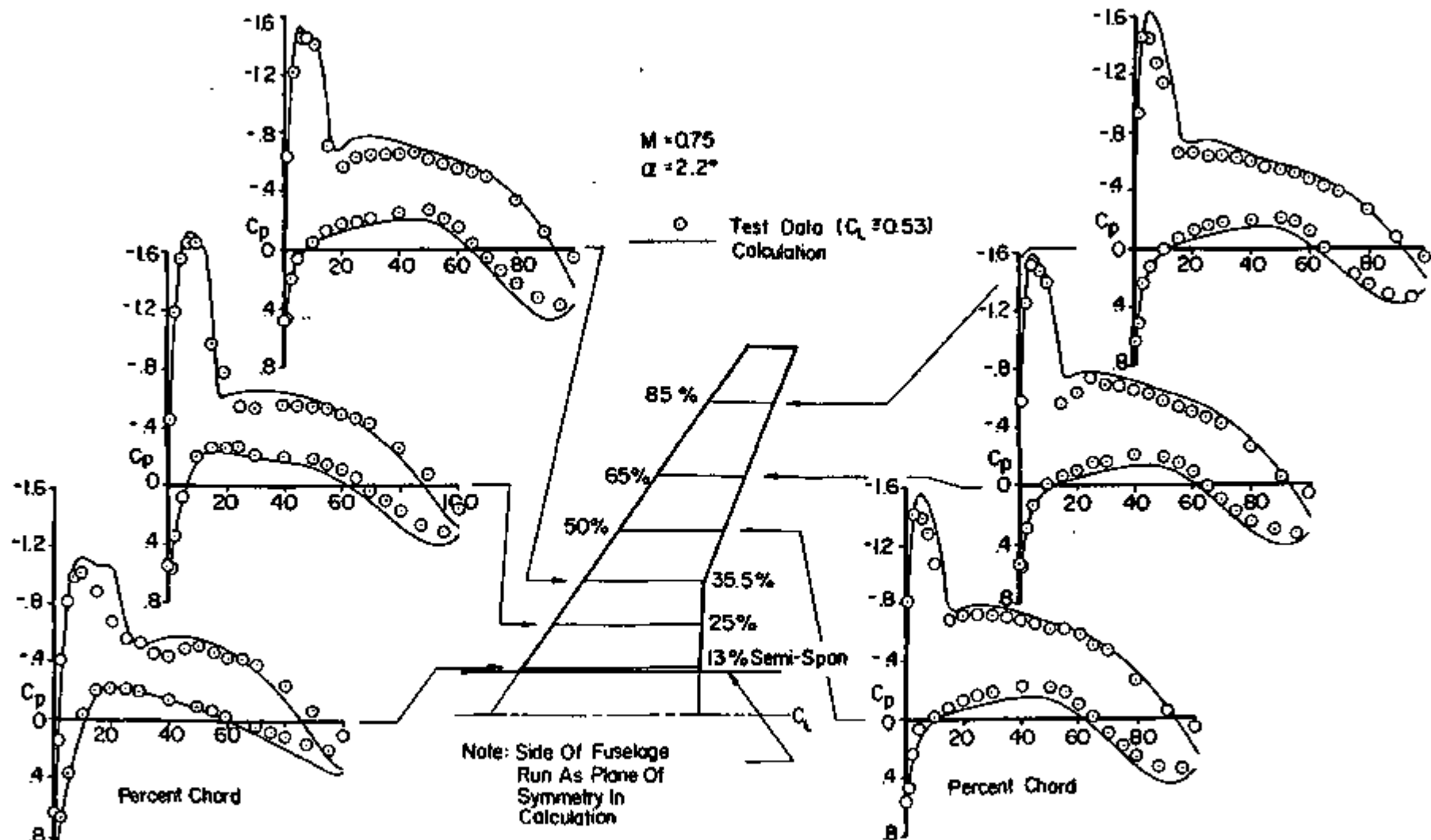


Figure 13 Comparison Of Calculated And Experimental Wing Pressure Distributions For DAC Case 5

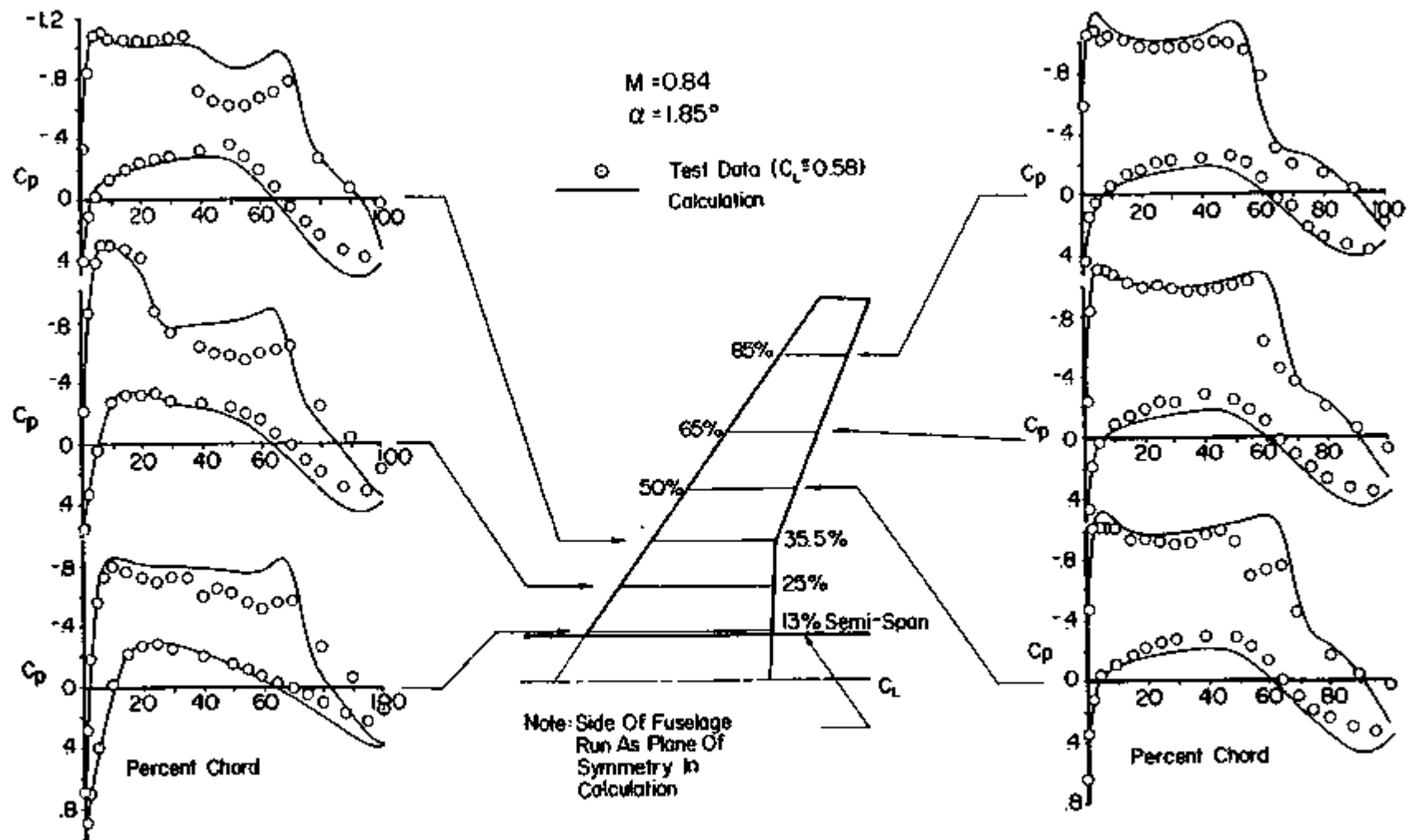


Figure 14 Comparison Of Calculated And Experimental Wing Pressure Distributions For DAC Case 5

## Appendix A. Description of the program

All the numerical results in this report were generated by the computer program FLO 22 listed in Appendix B. This program includes options to treat both a swept wing on a wall (Figure A1), and an isolated yawed wing (Figure A2). For swept wing calculations the sheared parabolic coordinates are introduced in planes parallel to the free stream. In the treatment of a yawed wing the whole coordinate system is rotated through a specified yaw angle, so that the X-Y planes are normal to the leading edge of the wing at its center line. In either case the wing section can be varied in an arbitrary manner, and the only restriction on the planform is that the leading edge may be any smooth curve, but it should not have kinks, since these would cause the second derivatives of the singular line of the coordinate system to become unbounded. Kinks are permitted in the trailing edge, on the other hand. The trailing edge defined by the input is actually replaced by a piecewise straight line connecting the nearest mesh points in the computational lattice.

The geometry is defined by giving the wing sections at successive span stations from the wing root to the tip, or in the case of a yawed wing, from the leading to the trailing tip. Up to 11 span stations may be used for this purpose, and the planform and dihedral are determined by specifying the chord and the x and y coordinates of the leading edge at these span stations. The wing section at each station is then determined by scaling and rotating a prescribed profile, given by a table of x and y coordinates. If the wing sections are similar, only the profile for the first station need be read in. The coordinates for the other stations are obtained by scaling the original profile to the proper chord, and rotating it to obtain the appropriate twist. If, on the other hand, the sections are not similar, the program permits the coordinates of new profiles to be read in at each span station. The wing section between stations is generated by interpolation. The location of the singular line about which the wing is unwrapped by the square root transformation is determined by the parameters XSING and YSING, which must be specified at each span station. It is important to choose these so that the mapped profile does not have any sharp bumps.

The main input to the program is read from Tape 5, and the output is written on Tape 6. Tapes 1, 2 and 3 are disk files used for internal storage in order to reduce the requirements for high speed memory. Tape 4 is a permanent storage device such as

a magnetic tape on which an intermediate result can be saved. The computation can then be continued for more iterations, starting from the values saved on Tape 4. The disk instructions in the version of the code listed in Appendix B are specialized to the CDC 6600 using the FTN compiler. Otherwise the code should be readily adaptable to other computers.

The data deck for a run is arranged to include title cards listing the required data items. The complete set of title cards provides a list of all the data which must be supplied, and can be used as a guide in setting up a data deck. Each title card is followed by one or more cards supplying the numerical values of the parameters listed on the title card. All data items are read as floating point numbers in fields of 10 columns, and values representing integer parameters are converted inside the program. A glossary of the input parameters is given in Table 1, and a typical data deck is shown in Table 2.

## Table 1. Glossary of input parameters

(Listed in order of their occurrence on the data title cards)

### TITLE CARD 1

NX	The number of mesh cells in the direction of the chord used at the start of the calculation. NX = 0 causes termination of the program.
NY	The number of mesh cells in the direction normal to the chord and span.
NZ	The number of mesh cells in the span direction.
FPLOT	Controls generation of plots. FPLOT=0. for a print plot but no Calcomp plot at each span station. FPLOT=1. for both a print plot and a Calcomp plot at each span station. FPLOT=2. for a Calcomp plot but no print plot at each span station. FPLOT=3. for a three dimensional Calcomp plot only.
XSCAL, PSCAL	Control the scales of the Calcomp plots. XSCAL>0. scales each section plot to XSCAL XSCAL=0. scales each section plot to 5.0 XSCAL<0. scales the maximum chord to XSCAL, and each section plot proportionately to the local chord. PSCAL≠0. sets the pressure scale to PSCAL per inch in each section plot. PSCAL=0. sets the pressure scale to 0.4 per inch in each section plot. Also, PSCAL>0. scales the three dimensional plot so that the span or semispan is 5. If PSCAL=0. and XSCAL≠0. then the three dimensional plot is scaled so that the maximum chord is 1/2 XSCAL.
FCONT	Indicator which determines the manner of starting the program. FCONT=0. indicates the calculation begins at iteration zero. FCONT=1. indicates the computation is to be continued from a previous calculation. In this case the values of the velocity potential and the circulation are read from a magnetic tape where they were previously stored (Tape 4). It is still necessary to provide the complete data deck to redefine the geometry. The count of the iteration cycles is continued from the final count of the previous calculation and the maximum number of additional iterations to be performed is defined by MIT.



TITLE CARD 2

- MIT The maximum number of iteration cycles which will be computed.
- COV The desired accuracy. If the maximum correction is less than COV the calculation terminates or proceeds to a finer mesh, otherwise the number of cycles set by MIT are completed.
- P1 The subsonic relaxation factor for the velocity potential. It is between 1. and 2. and should be increased towards 2. as the mesh is refined.
- P2 The supersonic relaxation factor for the velocity potential. It is not greater than 1. and is normally set to 1.
- P3 The relaxation factor for the circulation. It is usually set to 1., but can be increased.
- BETA The damping parameter controlling the amount of added  $\phi_{st}$  (see equation (2.6), page 13). It is normally set between 0. and 0.25.
- STRIP Determines the split between horizontal and vertical line relaxation and is the proportion of the total mesh in which horizontal line relaxation is used. Fastest convergence is usually obtained by setting STRIP = 1. so that horizontal line relaxation is used for the entire mesh. If convergence difficulties are encountered STRIP may be reduced to some fraction between 0. and 1.
- FHALF Determines whether the mesh will be refined.  
FHALF=0.: the computation terminates after completing the prescribed number of iteration cycles or after convergence.  
FHALF $\neq$ 0.: the mesh spacing will be halved after MIT cycles have been run on the crude mesh size. An additional data card must be provided for the refined mesh giving the numerical values requested by Title Card 2. If  
FHALF<0 the interpolated potential will be smoothed |FHALF| times.

TITLE CARD 3

FMACH The free stream Mach number.

YAW The yaw angle of the wing in degrees.

ALPHA The angle of attack in degrees. When the wing is yawed, ALPHA is measured in the plane normal to the leading edge, not in the free stream direction.

CD0 The estimated parasite drag due to skin friction and separation. It is added to the pressure drag (sum of vortex drag plus wave drag) calculated by the program to give the total drag.

TITLE CARD 4

ZSYM Determines whether to treat a wing on a wall or an isolated wing.  
ZSYM=1.: the wing is on a wall  
ZSYM=0.: the wing is an isolated wing at a yaw angle given by YAW.

NC The number of span stations at which the wing section is defined on subsequent data cards from the wing root to the tip if ZSYM=1., or from the leading to the trailing tip if ZSYM=0. If NC<3 it is assumed that the wing geometry is the same as for the last case calculated and the computation for new values of FMACH, YAW, ALPHA and CD0 begins without further data items being read.

SWEEP1 Sweep of singular line at the wing root if ZSYM=1., or at the leading tip if ZSYM=0.

SWEEP2 Sweep of singular line at the tip.  
(SWEEP1 and SWEEP2 are used as end conditions for a spline fitting the x coordinates of the singular line.)

SWEEP Sweep of singular line in the far field.

DIHED1 Dihedral of singular line at the wing root if ZSYM=1., or at the leading tip if ZSYM=0.

DIHED2 Dihedral of singular line at the tip.  
(DIHED1 and DIHED2 are used as end conditions for a spline fitting the y coordinates of the singular line.)

DIHED Dihedral of singular line in the far field.

**TITLE CARD 5** (The geometry at the first span station)

**Z** Span location of the section.

**XLE, YLE** x and y coordinates of the leading edge.

**CHORD** The local chord value by which the profile coordinates are scaled.

**THICK** Modifies the section thickness. The y coordinates are multiplied by THICK.

**ALPHA** The angle through which the section is rotated to introduce twist. In the case of a yawed wing, this angle is measured in the axis system attached to the wing, not in the direction of the free stream.

**FSEC** Indicates whether or not the geometry for a new profile is supplied.  
**FSEC=0.:** the section is obtained by scaling the profile used at the previous span section according to the parameters CHORD, THICK, ALPHA. No further cards are read for this span station, and the next card should be the title card for the next span station, if any.  
**FSEC=1.:** the coordinates for a new profile are read from the data cards which follow.

**TITLE CARD 6** (Profile Geometry Supplied if FSEC=1.)

**YSYM** Indicates the type of profile.  
**YSYM=0.** denotes a cambered profile. Coordinates are supplied for upper and lower surfaces, each ordered from nose to tail with the leading edge included in both surfaces.  
**YSYM=1.** denotes a symmetric profile. A table of coordinates is read for the upper surface only.

**NU** The number of upper surface coordinates.

**NL** The number of lower surface coordinates.  
For YSYM=1., NL=NU even though no lower surface coordinates are given.

**TITLE CARD 7** (Additional Profile Geometry Supplied if FSEC=1.)

**TRAIL** The included angle at the trailing edge in degrees. The profile may be open, in which case it is the difference in angle between the upper and lower surfaces.

**SLOPT** The slope of the mean camber line at the trailing edge. This is used to continue the coordinate

surface, assumed to contain the vortex sheet, smoothly off the trailing edge. For heavily aft loaded airfoils, the lift is sensitive to the value of this parameter, which should be adjusted by comparing two dimensional calculations using parabolic coordinates with two dimensional calculations in the circle plane.

XSING, YSING      The coordinates of the singular point inside the nose about which the square root transformation is applied to generate parabolic coordinates. This point should be located as symmetrically as possible between the upper and lower surfaces at a distance from the nose roughly proportional to the leading edge radius. It can be seen whether the location has been correctly chosen by inspecting the coordinates of the mapped profile printed in the output. If the mapped profile has a bump at the center, the singular point should be moved closer to the leading edge. If the mapped profile is not symmetric near the center, with a step increase in  $y$ , say, as  $x$  increases through 0, the singular point should be moved closer to the upper surface. The coordinates of the singular point are chosen relative to the profile coordinates supplied on the cards which follow.

TITLE CARD 8      (Upper Surface Coordinates)

X,Y                The coordinates of the upper surface. These are read on the data cards which follow, one pair of coordinates per card in the first two fields of 10, from leading to trailing edge inclusive.

TITLE CARD 9      (Lower Surface Coordinates, Read if ISYM = 0.)

X,Y                The coordinates of the lower surface, read from leading edge to trailing edge. The leading edge point is the same as the upper surface leading edge point. The trailing edge point may be different if the profile has an open tail.

TITLE CARD 10,11... (Geometry at the Other Span Stations)

These title cards are the same as Title Card 5 (geometry for the first span station). The number of such cards depends on the number of input span stations NC. If the profiles are similar at each station except for scaling, thickness to chord ratio and rotation to introduce twist, FSEC=0. and no new profile coordinates are needed.

TABLE 2. DATA DECK FOR ONERA M6 WING

Columns Cards	1-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80	
Title of case	ONERA M6 WING (copied onto output and Calcomp plots)								
Title Card	NY 48.	NY 6.	NZ 8.	FPLOTT 1.	XSCAL 0.	PSCAL 0.	FCONT 0.		
Title Card	MIT 100. 100. 100.	COV 1.E-6 1.E-6 1.E-6	P1 1.6 1.6 1.6	P2 1. 1. 1.	P3 1. 1. 1.	BETA .10 .10 .10	STRIP 1. 1. 1.	FHALF 1. 1.	
Title Card	MACH .840	YAW 0.	ALPHA 3.06	CDO .010					
Title Card	ZSYM 1.	NC 6.	SWEEP1 29.9	SWEEP2 29.9	SWEEP 29.9	DIHED1 0.	DIHED2 0.	DIHED 0.	
Title Card	Z 0.	XLE 0.	YLE 0.	CHORD .6737	THICK 1.	ALPHA 0.	FSEC 1.		
Title Card	YSYM 1.	NU 72.	NL 72.						
Title Card	TRAIL 7.06	SLOFT 0	XSING .00725	YSING 0.					
Title Card (72 cards)	X (Coordinates of profile)	Y	(Upper Surface)						
Title Card	Z .2	XLE .1150	YLE 0.	CHORD .6147	THICK 1.	ALPHA 0.	FSEC 0.		
Title Card	Z .4	XLE .2300	YLE 0.	CHORD .5558	THICK 1.	ALPHA 0.	FSEC 0.		
Title Card	Z .6	XLE .3450	YLE 0.	CHORD .4968	THICK 1.	ALPHA 0.	FSEC 0.		
Title Card	Z .8	XLE .4600	YLE 0.	CHORD .4379	THICK 1.	ALPHA 0.	FSEC 0.		
Title Card	Z 1.0	XLE .5750	YLE 0.	CHORD .3789	THICK 1.	ALPHA 0.	FSEC 0.		

Both graphical and printed output are provided. The wing sections defining the geometric configurations are printed for each span station, if they are different, or for the first span station only if the sections are all similar. The program next prints the coordinates of the unfolded sections produced by the square root transformations at the root and the tip. These should be inspected to see that they are reasonably smooth. The program also prints a chart of an indicator IV showing the configuration of the wing in the coordinate surface to which it has been mapped. The values of IV are as follows:

- IV = 2 indicates a point on the wing
- 1 indicates a point on the trailing vortex sheet
- 0 indicates a point on the singular line
- 1 indicates a point adjacent to the edge of the wing  
or vortex sheet
- 2 indicates an ordinary point not in contact with the wing or vortex sheet.

The program next displays the iteration history. The maximum correction to the velocity potential and the maximum residual of the difference equations are printed at each cycle, together with the locations of the points where these occur in the computational lattice, and also the relaxation factors, the circulation at the wing center line, and the number of supersonic points.

After a specified maximum number of cycles has been completed, or a convergence criterion has been satisfied, the section lift, drag and moment coefficients are printed for each span station, and the pressure distribution is printed or displayed in a Calcomp plot as desired. Finally the characteristics of the complete wing are printed. These include the coefficients of lift and form drag computed by integrating the surface pressure, and the ratio of lift to form drag. An estimate of the friction drag coefficient may be supplied in the input, and this will be included to provide an estimate of the total drag coefficient of the ratio of lift to total drag. The pitching, rolling and yawing moments are also computed and printed. In the case of a yawed wing these are in an axis system normal to the wing leading edge at its center line. In the case of a wing on a wall the rolling moment is the root bending moment.

Finally additional Calcomp plots are generated if they are desired. These show the convergence history, and also a view of the complete wing and the three dimensional pressure distribution over the upper and lower surfaces separately, with the wing root or the leading tip at the bottom of the picture. If the mesh is to be refined the program then completes the same sequence of calculations and output for the new mesh.

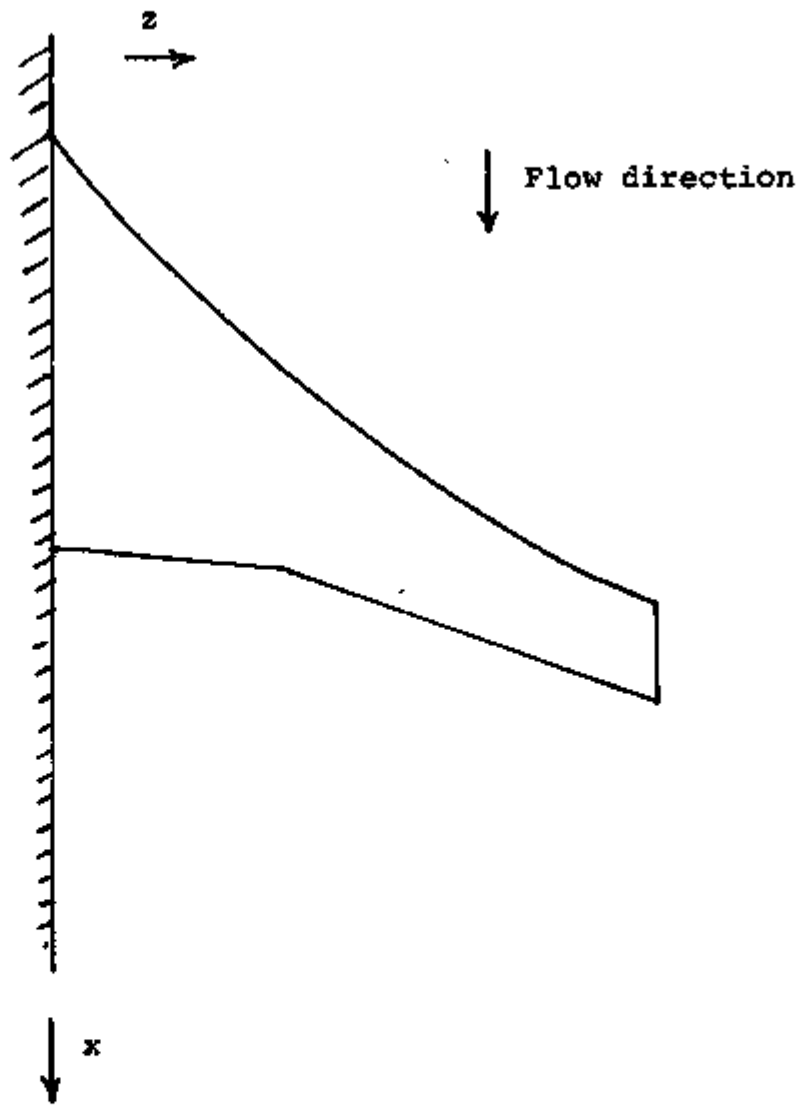


Figure A1. Swept wing on a wall.



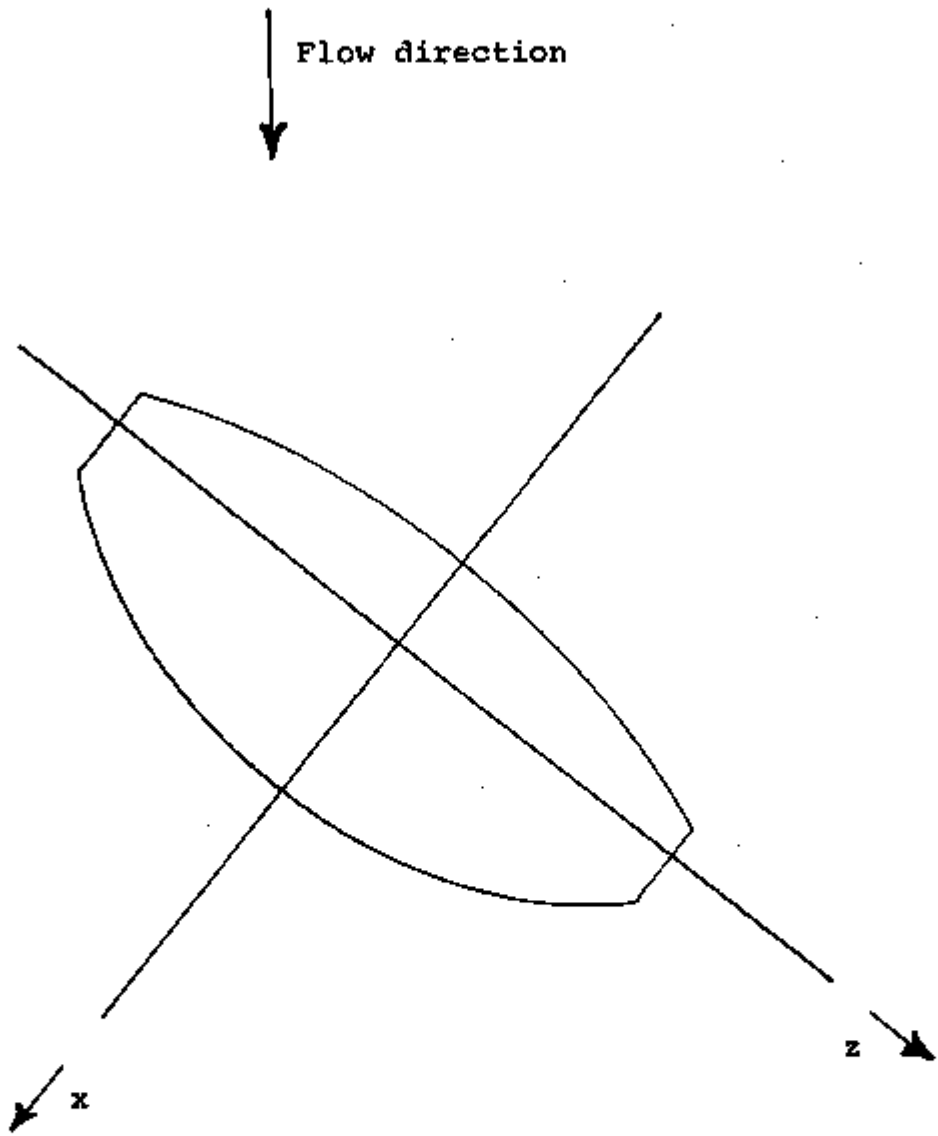


Figure A2. Yawed wing.

APPENDIX B. LISTING OF THE PROGRAM

```

PROGRAM FLO22(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,
1      TAPE5=INPUT,TAPE6=OUTPUT)
C      THREE DIMENSIONAL WING ANALYSIS IN TRANSONIC FLOW
C      USING SHEARED PARABOLIC COORDINATES
C      WITH STORAGE ON THE DISC
C      PROGRAMMED BY ANTONY JAMESON, MARCH 1974
C      REVISIONS BY D. A. CAUGHEY AND ANTONY JAMESON, DEC 1975-DEC 1976
C      G IS REDUCED VELOCITY POTENTIAL
COMMON      G(193,26,4),SO(193,35),EO(131),ZO(131),
1          IV(193,35),ITC1(35),ITC2(35),
2          AU(193),A1(193),A2(193),A3(193),
3          B0(26),B1(26),B2(26),B3(26),
4          Z(35),C1(35),C2(35),C3(35),
5          XC(35),XZ(35),XZZ(35),YC(35),YZ(35),YZZ(35),
6          NX,FY,FZ,KTE1,KTE2,SYN,KSYM,SCAL,SCALZ,
7          YAW,CYAW,RYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IO
COMMON/FLG/ STRIP,P1,P2,P3,BETA,FR,[R,JF,KR,DG,IG,JG,KG,NS
DIMENSION  XS(241,11),YS(241,11),
1          ZS(11),XLC(11),YLE(11),SLOPT(11),TRAIL(11),NP(11),
2          E1(11),E2(11),E3(11),F4(11),E5(11),
3          XP(241),YP(241),D1(241),D2(241),D3(241),
4          X(193),Y(193),SV(193),SM(193),CP(193),
5          CHOPD(35),SCL(35),SCO(35),SCM(35),TITLE(20),
6          FIT(3),CLVL(3),P10(3),P20(3),P30(3),BETA0(3),
7          STRIPQ(3),FHALF(3),RES(501),COUNT(501)
NO          = 241
NE          = 193
IREAD      = 5
IWRIT     = 6
KPLUT     = 0
IPLOT     = 1
ISTOP     = 2
N1        = 1
N2        = 2
N3        = 3
REWIND 1
REWIND 2
REWIND 3
REWIND 4
JC        = 0
RAO      = 57.2957745136823
1 WRITE (IWRIT,600)
  WRITE (IWRIT,2)
2 FORMAT(14HOPPGPAM FLO22,7GX,32HANTONY JAMESON,COURANT INSTITUTE/
1      50HOTHREE DIMENSIONAL WING ANALYSIS IN TRANSONIC FLOW,
2      36H USING SHEARED PARABOLIC COORDINATES)
READ (IREAD,530) TITLE
WRITE (IWRIT,630) TITLE
READ (IREAD,500)
READ (IREAD,510) FNX,FNY,FNZ,FPLOT,XSCAL,PSCAL,FCONT,FA1
NX          = FNX

```

```

NY          = FNY
NZ          = FNZ
IF (NX.LT.1) GO TO 201
KPL0T      = ABS(FPLOT)
READ (IREAD,500)
NM          = 0
11 NM        = NM + 1
READ (IREAD,510) FIT(NM),CJVO(NM),P10(NM),P20(NM),P30(NM),
1          BETA0(NM),STRIP0(NM),FHALF(NM)
IF (FHALF(NM).NE.0..AND.NM.LI.3) GO TO 11
FHALF(3)   = 0.
READ (IREAD,500)
READ (IREAD,510) FMACH,YA,AL,COC
YAW        = YA/PAD
ALPHA      = AL/PAD
CALL GEDM (ND,NC,NP,ZS,XS,YS,XLE,YLE,SLOPT,TRAIL,XP,YP,
1          SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED,
2          XTEC,CHORDC,ZTIP,ISYMO,KSVM)
ISYM       = ISYMO
IF (ALPHA.NE.0.) ISYM = 0
IF (KSVM.NE.0) YAW = 0.
CYAW       = COS(YAW)
SYAW       = SIN(YAW)
CA         = CYAW*COS(ALPHA)
SA         = CYAW*SIN(ALPHA)
IF (FCONT.LT.1.) GO TO 41
READ (4)   NX,NY,NZ,NM,K1,K2,NIT
MX         = NX + 1
MY         = NY + 2
MZ         = NZ + 3
DO 62 K=1,MZ
READ (4)   ((G(I,J,1),I=1,MX),J=1,MY)
BUFFER OUT(N3,1) (G(1,1,1),G(MX,MY,1))
IF (UNIT(N3).GT.0.) GO TO 1
BUFFER OUT(N1,1) (G(1,1,1),G(MX,MY,1))
IF (UNIT(N1).GT.0.) GO TO 1
62 CONTINUE
READ (4)   (EO(K),K=K1,K2)
REWIND N3
REWIND N1
REWIND 4
91 CALL COORD (NX,NY,NZ,KSVM,XTEC,ZTIP,XMAX,ZMAX,
1          SY,SCAL,SCALZ,AX,AY,AZ,
2          AC,A1,A2,A3,B0,B1,B2,B3,Z,C1,C2,C3)
CALL SINGL (NC,N2,KSVM,KTE1,KTE2,CHORDC,
1          SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED,
2          ZS,XLE,YLE,XC,XZ,XZZ,YC,YZ,YZZ,
3          Z,C1,C2,C3,c1,e2,e3,e4,e5,IND)
CALL SURF (ND,NE,NC,NX,NZ,ISYM,KSVM,KTE1,KTE2,SCAL,
1          YAW,AC,Z,ZS,XC,YC,SLOPT,TRAIL,XS,YS,NP,
2          ITE1,ITE2,IV,SO,ZO,AP,YP,D1,D2,D3,X,Y,IND)
IF (IND.EQ.0) GO TO 291
IF (FCONT.GE.1.) GO TO 101
NM          = 1

```

```

NIT      * 0
CALL ESTIM
IF (10.EQ.0) GO TO 1
REWIND N3
REWIND N1
101 WRITE (IWRIT,600)
FCOEF    = 0.
MIT      = FIT(NM) +NIT
KIT      = MIT
IF (NM.GT.1.AND.FHALF(NM).EQ.0.) KIT = 10
JIT      = NIT
KRES     = (MIT -MIT -2)/500 +2
JPES     = 0
NRES     = 0
COV      = COVC(NM)
STRIP    = STRIP(NM)
BETA     = BETA(NM)
MX       = NX +1
MY       = NY +2
MZ       = NZ +3
KY       = NY +1
K1       = 2
K2       = NZ
IF (KSYM.EQ.0) GO TO 103
K1       = 3
K2       = NZ +2
103 LZ    = NZ/2 +1
IF (KSYM.NE.0) LZ = 3
WRITE (IWRIT,104)
104 FORMAT(48H0INDICATION OF LOCATION OF WING AND VORTEX SHEET,
1        27H IN COORDINATE PLANE Y = 0./
2        27H0((IV(I,K),K=K1,K2),I=2,NX))
DO 106 I=2,NX
106 WRITE (IWRIT,650) (IV(I,K),K=K1,K2)
WRITE (IWRIT,600)
WRITE (IWRIT,112)
112 FORMAT(49H0CIRCUMFERENTIAL CELL DISTRIBUTION IN SQUARE ROOT PLANE,
1        54H AND MAPPED SURFACE COORDINATES AT CENTER LINE AND TIP/
2        15H0      X      ,15H ROOT PROFILE,15H TIP PROFILE )
DO 114 I=2,NX
114 WRITE (IWRIT,610) A0(I),S0(I,LZ),S0(I,KTEZ)
WRITE (IWRIT,116)
116 FORMAT(15H0 TE LOCATION ,15H POWER LAW )
WRITE (IWRIT,610) XMAX,AX
WRITE (IWRIT,600)
WRITE (IWRIT,116)
116 FORMAT(46H0NORMAL CELL DISTRIBUTION IN SQUARE ROOT PLANE/
1        15H0      Y      )
DO 120 J=2,KY
120 WRITE (IWRIT,610) B0(J)
WRITE (IWRIT,122)
122 FORMAT(15H0 SCALE FACTOR,15H POWER LAW )
WRITE (IWRIT,610) SY,AY
WRITE (IWRIT,600)

```

```

WRITE (IWRIT,124)
124 FORMAT(45HOSPANWISE CELL DISTRIBUTION AND SINGULAR LINE/
1      15H0      Z      ,15H      X SING      ,15H      Y SING      ,
2      15H      XZ      ,15H      YZ      ,15H      XZZ      ,
3      15H      YZZ      )
DO 126 K=K1,K2
126 WRITE (IWRIT,610) Z(K),XC(K),YC(K),XZ(K),YZ(K),XZZ(K),YZZ(K)
WRITE (IWRIT,126)
128 FORMAT(15H0 TIP LOCATION,15H POWER LAW )
WRITE (IWRIT,610) ZMAX,AZ
WRITE (IWRIT,600)
WRITE (IWRIT,132)
132 FORMAT(19H0ITERATIVE SOLUTION/
1      43HQSTRIP WIDTH FOR HORIZONTAL LINE RELAXATION)
WRITE (IWRIT,610) STRIP
WRITE (IWRIT,134)
134 FORMAT(15H0      NX      ,15H      NY      ,15H      NZ      )
WRITE (IWRIT,640) NX,NY,NZ
CALL SECOND(T)
WRITE (IWRIT,700) T
WRITE (IWRIT,136)
136 FORMAT(15H0 MACH NO      ,15H      YAW      ,15H      ANG OF ATTACK)
WRITE (IWRIT,610) FMACH,YA,AL
WRITE (IWRIT,136)
138 FORMAT(10H0ITEPATION,15H CORRECTION ,4H I ,4H J ,4H K ,
1      15H RESIDUAL ,4H 1 ,4H J ,4H K ,
2      10H CIRCULATN,10H REL FCT 1,10H REL FCT 2,10H REL FCT 3,
3      10H BETA ,10H SUBIC PTS)
141 NIT      = NIT +1
JIT      = JIT +1
P1      = P10(NM)
P2      = P20(NM)
P3      = P30(NM)
IF (NIT.LE.10) P1 = 1.
IF (NIT.LE.10) P3 = 1.
CALL MIXFLO
IF (JG.EQ.0) GO TO 151
J0      = 0
REWIND N1
REWIND N2
N      = N1
N1      = N2
N2      = N3
N3      = N
WRITE (IWRIT,660) NIT,DG,IG,JG,KG,FR,IR,JR,KR,EU(L2),
1      P1,P2,P3,BETA,NS
JRES      = JRES +1
IF (JRES.EQ.KRES) JRES = 1
IF (JRES.NE.1) GO TO 143
NRES      = NRES +1
COUNT(NRES) = NIT -1
RES(NRES) = FR
143 IF (JIT.EQ.KIT) GO TO 251
IF (NIT.LT.MIT.AND.ABS(DG).GT.COV.AND.ABS(DG).LT.20.) GO TO 141

```

```

GO TO 161
151 IF (JQ.EQ.1) GO TO 1
REWIND N1
REWIND N2
JQ      = 1
N       = N3
N3      = N2
N2      = N1
N1      = N
GO TO 141
161 RATE = 0.
IF (NRES.GT.1) RATE = (ABS(RES(NRES))/RES(1))
1          *(1./((COUNT(NRES) - COUNT(1)))
WRITE (IWRIT,162)
162 FORMAT(15H MAX RESIDUAL 1,15H MAX RESIDUAL 2,15H      WORK
1          15H RECOLTN/CYCLE)
WRITE (IWRIT,67) RES(1),RES(NRES),COUNT(NRES),RATE
CALL SECOND(T)
WRITE (IWRIT,700) T
WRITE (IWRIT,600)
DO 164 L=1,3
BUFFER IN (N1,1) (G(1,1,L),G(MX,MY,L))
IF (UNIT(N1).GT.0.) GO TO 151
164 CONTINUE
LX      = NX/2 + 1
K       = 2
171 K    = K + 1
IF (K.EQ.MZ) GO TO 151
DO 172 J=1,MY
DO 172 I=1,MX
G(I,J,1) = G(I,J,2)
172 G(I,J,2) = S(I,J,2)
BUFFER IN (N1,1) (G(1,1,3),G(MX,MY,3))
IF (UNIT(N1).GT.0.) GO TO 151
IF (K.LT.KTE1.OR.K.GT.KTE2) GO TO 171
I1      = ITE1(K)
I2      = ITE2(K)
CALL VELO (K,2,SV,SM,CP,X,Y)
CHORD(K) = X(I1) - X(I2)
CALL FCPLF (I1,I2,X,Y,CP,AL,CHORD(K),XC(K),SCL(K),SCD(K),SCH(K))
IF (KPLOT.GT.1.AND.K.GT.KTE1) GO TO 185
WRITE (IWRIT,600)
WRITE (IWRIT,182)
182 FORMAT(24HSECTION CHARACTERISTICS/
1          15H RACH NO      ,15H      YA      ,15H      ANG OF ATTACK)
WRITE (IWRIT,610) FMACH,YA,AL
WRITE (IWRIT,184)
184 FORMAT(15H SPAN STATION,15H      CL      ,15H      CO
1          15H      CM      1
185 WRITE (IWRIT,610) Z(K),SCL(K),SCD(K),SCH(K)
IF (KPLOT.LE.1) CALL CPLDT (I1,I2,FMACH,X,Y,CP)
IF (KPLOT.LT.1.OR.KPLOT.GT.2) GO TO 171
CALL GRAPH (IPLCT,I1,I2,X,Y,CP,TITLE,FMACH,YA,AL,
1          Z(K),SCL(K),SCD(K),CHORDO,XSCAL,PSCAL)

```

```

GO TO 171
191 CALL TDTFOR(KTE1,KTE2,CHORD,SCL,SCD,SCM,Z,XC,
1          CL,CD1,CMP,CMR,CMY)
  CD1      = CYAW*CD1
  CD       = CDO +CD1
  VLD1     = 0.
  IF (ABS(CD1).GT.1.E-6) VLD1 = CL/CD1
  VLD      = 0.
  IF (ABS(CD).GT.1.E-6) VLD = CL/CD
  WRITE (IWRIT,600)
  WRITE (IWRIT,192)
192 FORMAT(21HOWING CHARACTERISTICS/
1          15H0      FMACH FJ      15H      YAW      15H      ANG LF ATTACK)
  WRITE (IWRIT,610) FMACH,YA,AL
  WRITE (IWRIT,194)
194 FORMAT(15H0      CL      15H      CD FORM      15H      CD FRICTION ,
1          15H      CD      15H      L/D FORM      15H      L/D      )
  WRITE (IWRIT,611) CL,CD1,CDO,CD,VLD1,VLD
  WRITE (IWRIT,196)
196 FORMAT(15H0      CM FITCH      15H      CM ROLL      15H      CM YAW      )
  WRITE (IWRIT,611) CMP,CMR,CMY
  REWIND N1
  IF (KPLOT.LT.1) GO TO 201
  CALL RPLJTI(IPLCT,NPES,KES,COUNT,TITLE,FMACH,YA,AL,NX,NY,NZ)
  CALL THREEE(IPLCT,SV,SM,CP,X,Y,TITLE,YA,AL,
1          VLD,CL,CL,CHORD,XSCAL,PSCAL)
  IF (IO.EQ.0) GO TO 151
201 IF (ISTOP.EQ.1) GO TO 301
  IF (FHALF(NM).EQ.0.) GO TO 1
  NX      = NX +NX
  NY      = NY +NY
  NZ      = NZ +NZ
  CALL COOKO (NX,NY,NZ,KSVM,XTEG,ZTIP,XMAX,ZMAX,
1          SY,SCAL,SCALZ,AX,AY,AZ,
2          AO,A1,A2,A3,BO,B1,P2,B3,Z,C1,C2,C3)
  CALL SINGL (NC,NZ,KSVM,KTE1,KTE2,CHORD,
1          SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED,
2          ZS,XLE,YLE,XC,XZ,XZZ,YC,YZ,YZZ,
3          Z,C1,C2,C3,E1,E2,E3,E4,E5,IMO)
  CALL SURF (ND,NE,NC,NA,NZ,ISYM,KSVM,KTE1,KTE2,SCAL,
1          YAW,AU,Z,ZS,XC,YC,SLOPT,TRAIL,XS,YS,NP,
2          JTE1,JTE2,IV,SD,ZC,XP,YP,D1,D2,D3,X,Y,INC)
  IF (IND.EQ.0) GO TO 191
  CALL REFIN
  IF (IO.EQ.0) GO TO 221
  REWIND N1
  REWIND N2
  NSMOD   = -FHALF(NM)
  IF (NSMOD.LT.1) GO TO 211
  DD 202 N=1,NSMOD
  CALL SMOO
  IF (IO.EQ.0) GO TO 221
  REWIND N1
202 REWIND N2

```

```

211 N          * N1
    N1        * N2
    N2        * N3
    N3        * N
    NM        * NM +1
    MIT       * 0
    GO TO 101
221 NX        * NX/2
    NY        * NY/2
    NZ        * NZ/2
    CALL COORD (NX,NY,NZ,KSYM,XTFO,ZTIP,XMAX,ZMAX,
1          SY,SCAL,SCALZ,AX,AY,AZ,
2          AO,A1,A2,A3,BJ,B1,B2,B3,Z,C1,C2,C3)
    CALL SINGL (NC,NZ,KSYM,KTE1,KTE2,CHORD,
1          SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED,
2          ZS,XLE,YLE,XC,XZ,XZZ,YC,YZ,YZZ,
3          Z,C1,C2,C3,E1,E2,E3,E4,E5,IND)
    CALL SURF (NO,NE,NC,NX,NZ,ISYM,KSYM,KTE1,KTE2,SCAL,
1          YA,AG,Z,ZO,AC,YC,SLOPI,TRAIL,XS,YS,NP,
2          ITE1,ITE2,IV,SV,ZC,XP,YP,DI,D2,D3,X,Y,IND)
    IF (IND.EQ.0) GO TO 291
    GO TO 151
251 K1        * KTE1 -1
    K2        * KTE2 +ITE2(KTE2) -NX/2
    DO 252 M=1,3
    WRITE (4)  NX,NY,NZ,NM,K1,K2,M-1
    DO 262 K=1,MZ
    BUFFER IN (M,1) (G(I,J,1),G(MX,MY,1))
    IF (UNIT(M),GT.0.) GO TO 281
262 WRITE (4)  ((G(I,J,1),I=1,MX),J=1,MY)
    REWIND 4
    WRITE (4)  (FO(K),K=K1,K2)
    ENDFILE 4
252 CONTINUE
    REWIND 4
    CALL SSWTCH(1,JSTOP)
    IF (ISTOP.EQ.1) GO TO 161
    JIT       * 0
    IF (MIT.LT.MIT.AND.ABS(DG).GT.CDV.AND.ABS(DG).LT.10.) GO TO 141
    GO TO 161
281 REWIND 4
    GO TO 151
291 WRITE (IWRIT,600)
    WRITE (IWRIT,292)
292 FORMAT(24HOBAD DATA,SPLINE FAILURE)
    GO TO 1
301 IF (KPLOT.GT.0) CALL PLOT(0.,0.,999)
    STOP
500 FORMAT(1X)
510 FORMAT(8F10.6)
530 FORMAT(20A4)
600 FORMAT(1H1)
610 FORMAT(F12.4,7F15.4)
620 FORMAT(8E15.5)

```



```

630 FORMAT(1H0,20A4)
640 FORMAT(I8,7I15)
650 FORMAT(1X,32I4)
660 FORMAT(I10,E15.5,3I4,E15.5,3I4,>F10.5,I1)
670 FORMAT(2E15.4,2F15.4)
700 FORMAT(15H0COMPUTING TIME,F10.3,10H SECONDS)
END

```

```

SUBROUTINE GEOM (NU,NC,NP,ZS,XS,YS,XLE,YLE,SLOPT,TRAIL,XP,YP,
1 SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED,
2 XTED,CHORD,2TIP,ISYM,KSYM)
C GEOMETRIC DEFINITION OF WING
DIMENSION XS(ND,1),YS(ND,1),ZS(1),XLE(1),YLE(1),
1 SLOPT(1),TRAIL(1),XP(1),YP(1),NP(1)
IREAD = 5
IWRIT = 6
KAD = 57.2957795130523
READ (IREAD,500)
READ (IREAD,510) ZSYM,FNC,SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED
IF (FNC.LT.3.) RETURN
KSYM = ZSYM
NC = FNC
WRITE (IWRIT,2)
2 FORMAT(15H0 SWEEP(1) ,15H SWEEP(2) ,15H FINAL SWEEP ,
1 15H DIHED(1) ,15H DIHED(2) ,15H FINAL DIHED )
WRITE (IWRIT,610) XL,YL,CHORD,THICK,AL
WRITE (IWRIT,610) SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED
SWEEP1 = SWEEP1/RAD
SWEEP2 = SWEEP2/RAD
SWEEP = SWEEP/RAD
DIHED1 = DIHED1/RAD
DIHED2 = DIHED2/RAD
DIHED = DIHED/RAD
ISYM = 1
XTED = 0.
CHORD = 0.
K = 1
11 READ (IREAD,500)
READ (IREAD,510) ZS(K),XL,YL,CHORD,THICK,AL,FSEC
ALPHA = AL/RAD
IF (K.GT.1.AND.FSEC.EQ.0.) GO TO 31
READ (IREAD,500)
READ (IREAD,510) YSYM,FNU,FNL
NU = FNU
NL = FNL
N = NU +NL -1
READ (IREAD,500)
READ (IREAD,510) TPL,SL1,XSING,YSING
READ (IREAD,500)
DO 12 I=NL,N
12 READ (IREAD,510) XP(I),YP(I)

```

```

L      * NL +1
IF (YSYM.GT.0.) GO TO 15
READ  (IRFAD,500)
DO 14 I=1,NL
READ  (IRFAD,510) VAL,DUM
J      = L -I
XP(J)  = VAL
14  YP(J)  = DUM
GO TO 21
15  J      = L
DO 16 I=NL,N
J      = J -1
XP(J)  = XP(I)
16  YP(J)  = -YP(I)
21  WRITE (IWRIT,600)
WRITE (IWRIT,22) ZS(K)
22  FORMAT(16HPROFILE AT Z = ,F10.5/
1     15H0     TE ANGLE ,15H     TE SLOPE ,15H     X SING ,
2     15H     Y SING )
WRITE (IWRIT,610) TRL,SLT,XSING,YSING
WRITE (IWRIT,24)
24  FORMAT(15H0     X     ,15H     Y     )
DO 26 I=1,N
26  WRITE (IWRIT,610) XP(I),YP(I)
31  SCALE  = CHORD/(XP(1) -XP(NL))
XLE(K)   = XL + (XSING -XP(NL))*THICK*SCALE
YLE(K)   = YL + (YSING -YP(NL))*THICK*SCALE
XX       = XP(NL) + (XSING -XP(NL))*THICK
YY       = YP(NL) + (YSING -YP(NL))*THICK
CA       = COS(ALPHA)
SA       = SIN(ALPHA)
DO 32 I=1,N
XS(I,K)  = SCALE*((XP(I) -XX)*CA +THICK*(YP(I) -YY)*SA)
32  YS(I,K)  = SCALE*(THICK*(YP(I) -YY)*CA -(XP(I) -XX)*SA)
SLOPT(K) = THICK*SLT -TAN(ALPHA)
TRAIL(K) = THICK*TRL/RAD
NP(K)    = N
KTEO     = AMAX1(XTEO,XS(1,K))
CHORDO   = AMAX1(CHORDO,CHORD)
IF (YSYM.LE.0..OR.ALPHA.NE.0.) ISYM0 = 0
WRITE (IWRIT,52) ZS(K)
52  FORMAT(27HSECTION DEFINITION AT Z = ,F10.5/
1     15H0     XLE     ,15H     YLE     ,15H     CHORD
2     15H THICKNESS RATIO,15H     ALPHA )
WRITE (IWRIT,610) XL,YL,CHORD,THICK,AL
K      = K +1
IF (K.LE.NC) GO TO 11
ZO     = .5*(ZS(1) +ZS(NC))
IF (KSYM.NE.0) ZO = ZS(1)
DO 62 K=1,NC
62  ZS(K)   = ZS(K) -ZO
ZTIP   = ZS(NC)
RETURN
500  FORMAT(1X)

```

```

510 FORMAT(8F10.6)
600 FORMAT(1H1)
610 FORMAT(F12.4,7F15.4)
  ENO

```

```

      SUBROUTINE COCFD (NX,NY,NZ,KSYM,XTEO,ZTIP,XMAX,ZMAX,
1          SY,SCAL,SCALZ,AX,AY,AZ,
2          AO,A1,A2,A3,BO,B1,B2,B3,Z,C1,C2,C3)
C     SETS UP STRETCHED PARABOLIC AND SPANWISE COORDINATES
      DIMENSION AO(1),A1(1),A2(1),A3(1),BO(1),B1(1),B2(1),B3(1),
1          Z(1),C1(1),C2(1),C3(1)
      DX      = 2./NX
      DY      = 1./NY
      KY      = NY + 1
      OZ      = 2./NZ
      ZC      = 1. - OZ
      K1      = 2
      K2      = NZ
      IF (KSYM.EQ.0) GO TO 1
      DZ      = 1./NZ
      ZC      = 0.
      K1      = 3
      K2      = NZ + 2
1     AX      = .5
      AY      = .5
      AZ      = .5
      BX      = 0.
      BZ      = 0.
      XMAX    = .625
      ZMAX    = .625
      SY      = .5
      SCAL    = XTEO/(.50001+XMAX*XMAX)
      SCALZ   = ZTIP/(1.00001+ZMAX)
      V2      = (DX/DY)**2
      W1      = SCAL/SCALZ
      W2      = (W1*OZ/OZ)**2
      S73     = SQRT(73.)
      BBX     = -BX*SQRT(3.*(7. + S73))/((1. + S73)*XMAX**2)
      ABX     = 1. - BBX*SQRT((7. + S73)/12.)*XMAX**3
      CBX     = (19. + S73)*XMAX*XMAX/12.
      ABBX    = ABX + BBX*(3.*CBX - 4.*XMAX*XMAX)+XMAX*XMAX/
1          SQRT(CBX - XMAX*XMAX)
      DO 12 I=2,NX
      DD      = (I - 1)*DX - 1.
      B       = 1.
      IF (ABS(DD).GT.XMAX) GO TO 13
      A       = CBX - DD*DD
      AS      = SQRT(A)
      C       = ABX*AS + BBX*(3.*CBX - 4.*DD*DD)+DD*DD
      DO      = ABX*DD + BBX*AS*DD**3
      DI      = AS/C

```

```

DZ      = BFX*(CBX*(-6.*CBX + 19.*LD*DD) - 12.*DD**4)*GD/(A*C)
GO TO 14
13 IF (DD.LT.0.) B = -1.
A      = 1. - ((DC - B*XMAX)/(1. - XMAX))**2
C      = A**AX
D      = (AY + AX - 1.)*(1. - A)
DD     = B*XMAX + A*BX*(DC - P*XMAX)/C
D1     = A*C/((1. + D)*ABBX)
D2     = -(AX + AX)*(DD - B*XMAX)
1      = (3. + D)/((1. + D)*A*(1. - XMAX)**2)
14 A0(I) = DD
A1(I)  = .5*(1/OX)
A2(I)  = D1*D1
12 A3(I) = .5*OX*D2
DC 22 J=2,KY
DD     = (KY - J)*DY
A      = 1. - DD*LD
C      = A**AY
D      = (AY + AY - 1.)*(1. - A)
D1     = A*C/((1. + D)*SY)
D0(J)  = SY*DD/C
B1(J)  = .5*D1/DY
B2(J)  = D1*D1*V2
22 B3(J) = -AY*DD*DY*(3. + D)/((1. + D)*A)
BBZ    = -B2*SQRT(3.*(7. + S73))/((1. + S73)*ZMAX**3)
ABZ    = 1. - BBZ*SQRT((7. + S73)/12.)*ZMAX**3
CBZ    = (19. + S73)*ZMAX*ZMAX/12.
ABBZ   = ABZ + BBZ*(3.*CBZ - 4.*ZMAX*ZMAX)*ZMAX*ZMAX/
1      = SQRT(CBZ - ZMAX*ZMAX)
DU 32 K=2,K2
DD     = (K - K1)*DZ - LD
B      = 1.
IF (ABS(DD).GT.ZMAX) GO TO 33
A      = CBZ - DD*DU
AS     = SQRT(A)
C      = ABZ*AS + BBZ*(3.*CBZ - 4.*DD*DD)*DD*DD
DD     = ABZ*DD + BBZ*AS*DD**3
D1     = AS/C
D2     = BBZ*(CBZ*(-6.*CBZ + 19.*DD*DD) - 12.*DD**4)*DU/(A*C)
GO TO 34
33 IF (DD.LT.0.) B = -1.
A      = 1. - ((DC - B*ZMAX)/(1. - ZMAX))**2
C      = A**AZ
D      = (AZ + AZ - 1.)*(1. - A)
DD     = B*ZMAX + A*BZ*(DC - H*ZMAX)/C
D1     = A*C/((1. + D)*ABBZ)
D2     = -(AZ + AZ)*(DD - B*ZMAX)
1      = (3. + D)/((1. + D)*A*(1. - ZMAX)**2)
34 Z(K) = SCALZ*DD
C1(K)  = .5*C1*W1/DZ
C2(K)  = D1*D1*W2
32 C3(K) = .5*DZ*DZ
RETURN
END

```

```

SUBROUTINE  SINGL (NC,NZ,KSYP,KTE1,KTE2,CHORDO,
1           SWEEP1,SWEEP2,SWEEP,DIHED1,DIHED2,DIHED,
2           ZS,XLE,YLE,XC,XZ,XZZ,YC,YZ,YZZ,
3           Z,C1,C2,C3,E1,E2,E3,E4,E5,IND)
C GENERATES SINGULAR LINE FOR SQUARE ROOT TRANSFORMATION
DIMENSION  ZS(1),XLE(1),YLE(1),XC(1),XZ(1),XZZ(1),
1           YC(1),YZ(1),YZZ(1),Z(1),C1(1),C2(1),C3(1),
2           E1(1),E2(1),E3(1),E4(1),E5(1)
DO 2 K=1,NC
E4(K)      = 0.
2 E5(K)      = 0.
K1         = 2
K2         = NZ
IF (KSYP.EQ.0) GO TO 11
K1         = 3
K2         = NZ +2
KTE1       = 3
11 DO 12 K=K1,K2
IF (Z(K).LT.ZS(1)) KTe1 = K +1
IF (Z(K).LE.ZS(NC)) KTe2 = K
12 CONTINUE
B          = CHORDO
S1         = TAN(SWEEP1)
S2         = TAN(SWEEP2)
T1         = TAN(DIHED1)
T2         = TAN(DIHED2)
CALL SPLIF (1,NC,ZS,XLE,E1,E2,E3,1,S1,1,S2,0,0,IND)
CALL INTPL (KTE1,KTE2,Z,XC,1,NC,ZS,XLE,E1,E2,E3,0)
CALL INTPL (KTE1,KTE2,Z,XZ,1,NC,ZS,E1,E2,E3,E4,0)
CALL INTPL (KTE1,KTE2,Z,XZZ,1,NC,ZS,E2,E3,E4,E5,0)
CALL SPLIF (1,NC,ZS,YLE,E1,E2,E3,1,T1,1,T2,0,0,IND)
CALL INTPL (KTE1,KTE2,Z,YC,1,NC,ZS,YLE,E1,E2,E3,0)
CALL INTPL (KTE1,KTE2,Z,YZ,1,NC,ZS,E1,E2,E3,E4,0)
CALL INTPL (KTE1,KTE2,Z,YZZ,1,NC,ZS,E2,E3,E4,E5,0)
S          = B*TAN(SWEEP)
S1         = B*S1
S2         = B*S2
T          = B*TAN(DIHED)
T1         = B*T1
T2         = B*T2
XC(2)      = 3.*(XC(3) -XC(4)) +XC(5)
YC(2)      = 3.*(YC(3) -YC(4)) +YC(5)
IF (KSYP.NE.0) GO TO 31
N          = KTE1 -1
DO 22 K=K1,N
ZZ         = (Z(K) -Z(KTE1))/B
A          = EXP(ZZ)
XC(K)      = XC(KTE1) +S*ZZ -(S1 -S)*(1. -A)
YC(K)      = YC(KTE1) +T*ZZ -(T1 -T)*(1. -A)
XZ(K)      = (S +(S1 -S)*A)/B
YZ(K)      = (T +(T1 -T)*A)/B
XZZ(K)     = (S1 -S)*A/(B*B)
22 YZZ(K)   = (T1 -T)*A/(B*B)
31 N       = KTE2 +1

```

```

DC 32 K=N,K2
ZZ      = (Z(K) -Z(KTE2))/B
A       = EXP(-ZZ)
XC(K)  = AC(KTE2) +S*ZZ + (S2 -S)*(1. -A)
YC(K)  = YC(KTE2) +T*ZZ + (T2 -T)*(1. -A)
XZ(K)  = (S + (S2 -S)*A)/B
YZ(K)  = (T + (T2 -T)*A)/B
XZZ(K) = -(S2 -S)*A/(B+B)
32 YZZ(K) = -(T2 -T)*A/(B+B)
RETURN
END

```

```

SUBROUTINE SURF (ND,NE,NC,NX,NZ,ISYM,KSYM,KTE1,KTE2,SCAL,
1             YAW,AO,Z,ZS,AC,YC,SLOPT,TRAIL,XS,YS,NP,
2             ITE1,ITE2,IV,SO,ZO,XP,YP,D1,D2,D3,X,Y,IND)
C INTERPOLATES MAPPED WING SURFACE AT MESH POINTS
C INTERPOLATION IS LINEAR IN PHYSICAL PLANE
DIMENSION SO(NE,1),XS(ND,1),YS(ND,1),ZS(1),SLOPT(1),TRAIL(1),
1         XC(1),YC(1),AO(1),Z(1),ZC(1),X(1),Y(1),
2         XP(1),YP(1),D1(1),D2(1),D3(1),
3         IV(NE,1),NP(1),ITE1(1),ITE2(1)
PI      = 3.14159265358979
IYAW   = TAN(YAW)
S1     = .5*SCAL
DX     = 2./NX
LX     = NX/2 +1
MX     = NX +1
MZ     = NZ +3
IVO    = 1 -ISYM -ISYM -ISYM
IV1    = -1 -ISYM
DO 2 K=1,MZ
ITE1(K) = MX
ITE2(K) = MX
DO 2 I=1,MX
IV(I,K) = -2
2 SO(I,K) = 0.
K      = KTE1
K2     = 1
21 K2  = K2 +1
K1     = K2 -1
R2     = 1.
IF (ZS(K2) -Z(K)) 21,25,25
23 R2  = (Z(K) -ZS(K1))/(ZS(K2) -ZS(K1))
25 R1  = 1. -R2
C      = R1*XS(1,K1) +R2*XS(1,K2)
CC     = SQRT((C +C)/SCAL)
DO 32 I=2,NX
IF ((AO(I) +.5*DX).LT.-CC) I1 = I +1
IF ((AO(I) -.5*DX).LT.CC) I2 = I
32 CONTINUE
ITE1(K) = I1

```

```

I1E2(K)  * I2
CC        * AO(12)/CC
Z0(K)    * Z(K) -TYAW*(XC(K)  +S1*AO(12)*AO(12))
KK        * K1
P         * R1
41 N      * NP(KK)
U         * SQRT(XS(1,KK)/C)/CC
DO 42 I=2,NX
42 X(I)   * J*AO(I)
ANGL     * PI +PI
U        * 1.
V        * 0.
DO 44 I=1,N
R        * SQRT(XS(I,KK)**2 +YS(I,KK)**2)
IF (F.EQ.0.) GO TO 45
ANGL     * ANGL +ATAN2((U*YS(I,KK) -V*XS(I,KK)),
1          (U*YS(I,KK) +V*YS(I,KK)))
U        * XS(I,KK)
V        * YS(I,KK)
R        * SQRT((R +R)/SCAL)
XP(I)    * R*COS(.5*ANGL)
YP(I)    * R*SIN(.5*ANGL)
GO TO 44
45 ANGL   * PI
U        * -1.
V        * 0.
XP(I)    * 0.
YP(I)    * 0.
44 CONTINUE
ANGL     * ATAN(SLOPT(KK))
ANGL1    * ATAN(YS(1,KK)/XS(1,KK))
ANGL2    * ATAN(YS(N,KK)/XS(N,KK))
ANGL1    * ANGL -.5*(ANGL1 -TRAIL(KK))
ANGL2    * ANGL -.5*(ANGL2 +TRAIL(KK))
T1       * TAN(ANGL1)
T2       * TAN(ANGL2)
CALL SPLIF (1,N,XP,YP,D1,D2,D3,1,T1,1,T2,0,0.,1ND)
CALL INTPL (I1,I2,X,Y,1,0,XP,YP,D1,D2,D3,0)
X1       * .25*XS(1,KK)
A        * SLOPT(KK)*(XS(1,KK) -X1)
B        * 1./(XS(1,KK) -X1)
ANGL     * PI +PI
U        * 1.
V        * 0.
M        * I1 -1
DO 52 I=2,M
XX       * .5*SCAL*X(I)**2
L        * B*(XX -A1)
YY       * YS(1,KK) +A*ALOE(D)/D
R        * SQRT(XX**2 +YY**2)
ANGL     * ANGL +ATAN2((U*YY -V*XX),(U*XX +V*YY))
U        * XX
V        * YY
R        * SQRT((R +R)/SCAL)

```

```

57 Y(I)      = R*SIN(.7*ANGL)
   A        = SUBPT(KK)*(XS(N,PK) -X1)
   B        = 1./(XS(N,PK) -X1)
   ANGL     = 0.
   U        = 1.
   V        = 0.
   M        = IZ +1
   DO 54 I=M,NX
   XX       = .5*SCAL*(I)**2
   D        = B*(XX -X1)
   YY       = YS(N,KK) +A*ALOG(D)/D
   R        = SQRT(XX**2 +YY**2)
   ANGL     = ANGL +ATAN2((U*YY -V*XX),(U*XX +V*YY))
   U        = XX
   V        = YY
   P        = SQRT((U +R)/SCAL)
54 Y(I)     = R*SIN(.5*ANGL)
   Q        = P*CC*CC
   DO 62 I=2,NX
62 SD(I,K) = SO(I,K) +Q*Y(I)
   IF (KK.EQ.K2) GO TO 71
   KK       = K2
   P        = #2
   GO TO 41
71 DO 72 I=11,12
72 IV(I,K) = 2
   M        = I1 -1
   DO 74 I=2,M
   ZZ       = Z(K) -TYAW*(XC(K) +S1*AO(I)*AO(I))
   IF (ZZ.GE.ZO(KTE1)) IV(I,K) = IV0
74 CONTINUE
   M        = IZ +1
   DO 76 I=M,NX
   ZZ       = Z(K) -TYAW*(XC(K) +S1*AO(I)*AO(I))
   IF (ZZ.GE.ZO(KTE1)) IV(I,K) = IV0
76 CONTINUE
   K2       = K2 -1
   K        = K +1
   IF (K.LE.KTE2) GO TO 21
   K1       = 2
   K2       = NZ
   IF (KSYM.EQ.0) GO TO 81
   K1       = 3
   K2       = NZ +2
81 DO 82 I=2,NX
   ZZ       = Z(K) -TYAW*(XC(K) +S1*AO(I)*AO(I))
   IF (ZZ.LE.ZS(NC).AND.ZZ.GE.ZO(KTE1)) IV(I,K) = IV0
82 CONTINUE
   K        = K +1
   IF (K.LE.K2) GO TO 81
   N        = KTE2
   IF (YAW.LE.0.) GO TO 93
   IO       = ITEL(KTE2) +1
   GO 92 I=IC,LX

```



```

      N          = N + 1
92  ZQ(N)       = Z(KTE2) -TYAW*(XC(KTE2) +S1*AO(I)*AO(I))
93  I          = ITE1(KTE1)
      ZQ(KTE1-1) = Z(KTE1-1) -TYAW*(XC(KTE1-1) +S1*AO(I)*AO(I))
      ZQ(N+1)    = Z(KTE2+1)
      DO 102 K=K1,K2
      DO 104 I=2,NX
      IF (IV(I,K).GT.C) GO TO 104
      IF (IV(I+1,K+1).GT.O.OR.IV(I-1,K+1).GT.O) IV(I,K) = IV1
      IF (IV(I+1,K-1).GT.O.OR.IV(I-1,K-1).GT.O) IV(I,K) = IV1
104 CONTINUE
102 IF (SO(LX,K).LT.1.E-05) IV(LX,K) = 0
      IF (KSYM.EQ.O) RETURN
      DO 112 I=2,NX
112 SO(I,2)    = S.*(SO(I,3) -SO(I,4)) +SO(I,5)
      RETURN
      END

```

```

C      SUBROUTINE ESTIM
      INITIAL ESTIMATE OF REDUCED POTENTIAL
      COMMON      G(193,26,4),SO(193,35),EU(131),ZO(131),
1      IV(193,35),ITE1(35),ITE2(35),
2      AO(193),A1(193),A2(193),A3(193),
3      BO(26),P1(26),B2(26),B3(26),
4      Z(35),C1(35),C2(35),C3(35),
5      XC(35),XZ(35),XZZ(35),YC(35),YZ(35),YZZ(35),
6      NX,NY,NZ,KTE1,KTE2,ISYM,KSYM,SCAL,SCALZ,
7      YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IG
      MX          = NX + 1
      KY          = NY + 1
      MY          = NY + 2
      MZ          = NZ + 3
      DO 12 I=1,193
      DO 12 J=1,26
      DO 12 K=1,4
12  G(I,J,K)     = 0.
      K          = 1
21  DO 22 I=2,NX
      G(I,KY+1,1) = 0.
      IF (IV(I,K).LT.2) GO TO 22
      DSI        = SO(I+1,K) -SO(I-1,K)
      DSK        = SO(I,K+1) -SO(I,K-1)
      SX         = A1(I)*DSI
      SZ         = C1(K)*DSK
      FH         = AO(I)*AC(I) +SO(I,K)*SO(I,K)
      H          = 1./FH
      AZ         = -AO(I)*XZ(K) -SO(I,K)*YZ(K)
      BZ         = -AO(I)*YZ(K) +SC(I,K)*XZ(K)
      HZ         = AZ*SX -BZ +FH*SZ
      FYY        = 1. +SX*SX +H*HZ*HZ
      FXY        = SX +H*AZ*HZ

```

```

V      = SA*AO(1) -CA*SO(1,K)
U      = CA*AO(1) +SA*SO(1,K)
W      = SYAW +CA*XZ(K) +SA*YZ(K)
G(I,KY+1,1) = G(I,KY-1,1)
1      +(V*(1. -M*BZ*MZ) -U*FY -W*MZ)/(FYY*B1(KY))
22 CONTINUE
BUFFER OUT(N3,1) (G(I,1,1),G(MX,MY,1))
IF (UNIT(N3).GT.0.) GO TO 41
BUFFER OUT(N1,1) (G(1,1,1),G(MX,MY,1))
IF (UNIT(N1).GT.0.) GO TO 41
K      = K +1
IF (K.LE.MZ) GO TO 21
K1     = KTE1 -1
K2     = KTE2 +1ITE2(KTE2) -NX/2
DO 32 K=K1,K2
32 EG(K) = 0.
IC     = 1
RETJPN
41 IC   = 0
KRTJPN
END

```

C  
C  
SUBROUTINE MIXFLU  
SOLUTION OF EQUATIONS FOR MIXED SUBSONIC AND SUPERSONIC FLOW  
USING ROTATED DIFFERENCE SCHEME

```

COMMON G(193,26,4),SO(193,35),EO(131),ZO(131),
1      IV(193,35),ITE1(35),ITE2(35),
2      A0(193),A1(193),A2(193),A3(193),
3      B0(26),B1(26),B2(26),B3(26),
4      Z(35),C1(35),C2(35),C3(35),
5      XC(35),XZ(35),XZZ(35),YC(35),YZ(35),YZZ(35),
6      NX,NY,NZ,KTE1,KTE2,ISYM,KSYM,SCAL,SCALZ,
7      YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IO
COMMON/FLU/ STRIP,P1,P2,P3,BETA,FR,IR,JR,KR,DG,IG,JG,KG,NS
COMMON/SWF/ GK1(193,26),GK2(193,26),
1      SX(193),SZ(193),SXX(193),SXZ(193),SZZ(193),
2      R0(193),F1(193),C(193),D(193),
3      G10(26),G20(26),G30(26),G40(26),G1(26),G2(26),
4      I1,I2,K,L,N0,LX,MX,KY,MY,T1,AA0,Q1,Q2,TYAW,S1
LX     = NX/2 +1
MX     = NX +1
KY     = NY +1
MY     = NY +2
TYAW  = SYAW/CYAW
S1     = .5*SCAL
DX     = 2./NX
T1     = DX*(X
AA0    = 1./FMACH**2 +.2
Q1     = 2./P1
Q2     = 1./P2
FR     = 0.

```

```

IR          = 0
JR          = 0
KR          = 0
DG          = 0.
IG          = 0
JG          = 0
KG          = 0
NS          = 0
K1          = 2
IF (FMACH.GE.1.) K1 = 3
K2          = NZ
IF (KSYM.EQ.0) GO TO 1
K1          = 3
K2          = NZ +2
1 F          = ABS(.5*STRIP*NX)
L           = F
IF (L.EQ.NX/2) L = L -1
I1          = LX -L
I2          = LX +L
IF (L.EQ.0) I2 = LX -1
DO 2 L=1,3
BUFFER IN (N1,1) (G(1,1,L),G(MX,MY,L))
IF (UNIT(N1).GT.0.) GO TO 101
2 CONTINUE
DO 4 J=1,MY
DO 4 I=1,MX
G(I,J,4) = G(I,J,1)
GK1(I,J) = G(I,J,1)
4 GK2(I,J) = G(I,J,1)
K           = 2
L           = 2
NO          = KTE1 -1
IF (K.EQ.K1) GO TO 21
BUFFER OUT(N2,1) (G(1,1,4),G(MX,MY,4))
IF (UNIT(N2).GT.0.) GO TO 101
BUFFER IN (N1,1) (G(1,1,4),G(MX,MY,4))
IF (UNIT(N1).GT.0.) GO TO 101
IF (KSYM.EQ.0) GO TO 51
I           = LX
DSI         = SO(I+1,3) -SO(I-1,3)
DSK         = SO(I,4) -SO(I,2)
SX(I)      = A1(1)*DSI
SZ(I)      = C1(3)*OSK
R           = AMING(1,IV(I,K))
J           = KY
DO 12 M=2,KY
YP         = B0(J) +SO(I,3)
H          = R/(1. -R +YP*YP)
AZ         = -YP*YZ(3)
BZ         = YP*AZ(3)
A          = H*AZ*A1(1)
B          = (H*(BZ -AZ*SX(I)) -SZ(I))*B1(J)
DGI        = G(I+1,J,3) -G(I-1,J,3)
DGJ        = G(I,J+1,3) -G(I,J-1,3)

```

```

G(I,J,2) = G(I,J,4) + (A*DG1 - B*DGJ)/C1(3)
GK1(I,J) = G(I,J,2)
G(I,J,1) = 3.*(G(I,J,2) - G(I,J,3)) + G(I,J,4)
GK2(I,J) = G(I,J,1)
P = 1.
12 J = J - 1
J = KY + 1
G(I,J,2) = G(I,J,4) + (A*DG1 - B*DGJ)/C1(3)
GK1(I,J) = G(I,J,2)
G(I,J,1) = 3.*(G(I,J,2) - G(I,J,3)) + G(I,J,4)
GK2(I,J) = G(I,J,1)
M = MX/2 - 1
DG 14 II=1,M
I = LX - II
GO TO 16
15 I = LX + II
16 DSI = SO(I+1,3) - SO(I-1,3)
DSK = SO(I,4) - SO(I,2)
SX(I) = A1(I)*DSI
SZ(I) = C1(3)*DSK
DO 18 J=2,KY
YP = HG(J) + SO(I,3)
H = 1./(AO(I)*AO(I) + YP*YP)
AZ = -AO(I)*XZ(3) - YP*YZ(3)
BZ = -AO(I)*YZ(3) + YP*XZ(3)
S = SIGN(1.,AZ)
A = H*ABS(AZ)*A1(I)
B = (H*(BZ - AZ*SX(I)) - SZ(I))*B1(J)
IP = I + IFIX(S)
IM = I - IFIX(S)
DGI = G(I,J,4) - G(IM,J,4)
DGJ = G(I,J+1,3) - G(I,J-1,3)
G(I,J,2) = (C1(3)*G(I,J,4) + A*(G(IP,J,2) + DGI) - B*DGJ)/
1 (C1(3) + A)
GK1(I,J) = G(I,J,2)
G(I,J,1) = 3.*(G(I,J,2) - G(I,J,3)) + G(I,J,4)
18 GK2(I,J) = G(I,J,1)
J = KY + 1
G(I,J,2) = (C1(3)*G(I,J,4) + A*(G(IP,J,2) + DGI) - B*DGJ)/
1 (C1(3) + A)
GK1(I,J) = G(I,J,2)
IF (1.LT.LX) GL TO 15
14 CONTINUE
GO TO 51
21 BUFFER OUT(NZ,1) (G(1,1,4),G(MX,MY,4))
DO 22 J=1,MY
G10(J) = G(I2,J,2)
G20(J) = G(I2-1,J,2)
G30(J) = G(I1,J,2)
22 G40(J) = G(I1+1,J,2)
DO 32 I=2,NX
DSI = SO(I+1,K) - SO(I-1,K)
DSK = SO(I,K+1) - SO(I,K-1)
DSII = SO(I+1,K) - SO(I,K) - SO(I,K) + SO(I-1,K)

```

```

1      +A3(I)*DS1
DSKK   = SO(I,K+1) -SO(I,K) -SO(I,K) +SC(I,K-1)
1      +C3(K)*DSK
DSIK   = SO(I+1,K+1) -SO(I-1,K+1) -SO(I+1,K-1) +SO(I-1,K-1)
SX(I)  = A1(I)*DS1
SZ(I)  = C1(K)*DSK
SXX(I) = A2(I)*DS1
SZZ(I) = C2(K)*DSKK
32 SXZ(I) = T1*A1(I)*C1(K)*DSIK
IF (I2.GT.11) CALL YSWEEP
IF (UNIT(N2).GT.0.) GO TO 101
IF (*.LT.K2) BUFFER IN (N1,1) (G(1,1,4),G(MX,MY,4))
IF (I1.GT.2) CALL XSWEEP
IF (UNIT(N1).GT.0.) GO TO 101
IF (K.NE.KTE2.CP.YAW.LE.C.) GO TO 51
10     * ITE1(K) +1
DG 42 I=IC,LX
M      = NX +2 -I
E      = G(M,KY,2) -G(I,MY,2)
NO     = NO +1
42 EQ(NO) = EC(NO) +P3*(E -EQ(NO))
51 IF (K.EQ.K2) GO TO 61
DO 52 J=1,MY
DO 52 I=1,MX
G(I,J,1) = G(I,J,2)
G(I,J,2) = G(I,J,3)
G(I,J,3) = G(I,J,4)
52 G(I,J,4) = G(I,J,1)
K      = K +1
GO TO 21
61 DO 62 L=2,3
BUFFER OUT(N2,1) (G(1,1,L),G(MX,MY,L))
IF (UNIT(N2).GT.0.) GO TO 101
62 CONTINUE
FR     = 1.2*FR/AA0
10     = 1
RETURN
101 10     = 0
RETURN
END

```

```

C SUBROUTINE YSWEEP
ROW RELAXATION
COMMON G(193,26,4),SO(193,35),EC(131),ZO(131),
1 IV(193,35),ITE1(35),ITE2(35),
2 AV(193),A1(193),A2(193),A3(193),
3 B0(26),B1(26),B2(26),B3(26),
4 Z(35),C1(35),C2(35),C3(35),
5 XC(35),X2(35),XZZ(35),YC(35),YZ(35),YZZ(35),
6 NX,NY,NZ,KTE1,KTE2,ISYM,KSVM,SCAL,SCALZ,
7 YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IO

```

```

COMMON/FLU/ STRIP,P1,P2,P3,BETA,FR,IR,JR,KR,DG,IG,JG,KG,NS
COMMON/SWP/ GK1(193,26),GK2(193,26),
1          SX(193),SZ(193),SXX(193),SYZ(193),SZZ(193),
2          RG(193),K1(193),C(193),D(193),
3          G10(26),G20(26),G30(26),G40(26),G1(26),G2(26),
4          I1,I2,K,L,N0,LX,MX,KY,MY,T1,AA0,Q1,Q2,TYAW,S1
  J1          = 2
  IF (FMACH.GE.1.) J1 = 3
  C(I1-1)    = 0.
  D(I1-1)    = 0.
  DO 12 I=I1,12
  RO(I)      = 1.
  RI(I)      = 1.
  GK1(I,1)   = G(I,1,L)
12 GK1(I,J1=1) = G(I,J1-1,L)
  J          = J1
  I3        = I2
31 BC       = -T1*B1(J)*C1(K)
  DO 32 I=I1,I3
  AB        = -T1*A1(I)*B1(J)
  AC        = T1*A1(I)*C1(K)
  YP        = S0(I,K) +B0(J)
  A         = 1. -RO(I) +AC(I)*AO(I) +YP*YP
  H         = RO(I)/A
  FH        = RO(I)*A
  P         = AO(I)*(4.*YP*YP -FH)
  Q         = YP*(4.*AO(I)*AO(I) -FH)
  A         = XZ(K)*XZ(K) -YZ(K)*YZ(K)
  B         = (XZ(K) +XZ(K))*YZ(K)
  AZ        = -AO(I)*XZ(K) -YP*YZ(K)
  BZ        = -AO(I)*YZ(K) +YP*XZ(K)
  CZ        = H*H*(P*A -Q*B) -AO(I)*XZZ(K) -YP*YZZ(K)
  DZ        = H*H*(Q*A +P*B) -AO(I)*YZZ(K) +YP*XZZ(K)
  DGI       = G(I+1,J,L) -G(I-1,J,L)
  DGJ       = G(I,J+1,L) -G(I,J-1,L)
  DGK       = G(I,J,L+1) -G(I,J,L)
  DGIJ      = G(I+1,J,L) -G(I,J,L) -G(I,J,L) +G(I-1,J,L)
  I         +A3(I)*DGI
  DGJJ      = G(I,J+1,L) -G(I,J,L) -G(I,J,L) +G(I,J-1,L)
  J         -B3(J)*DGJ
  DGKK      = G(I,J,L+1) -G(I,J,L) -G(I,J,L) +G(I,J,L-1)
  I         +C3(K)*DGK
  DGIJ      = G(I+1,J+1,L) -G(I-1,J+1,L)
  I         -G(I+1,J-1,L) +G(I-1,J-1,L)
  DGIK      = G(I+1,J,L+1) -G(I+1,J,L-1)
  I         -G(I-1,J,L+1) +G(I-1,J,L-1)
  DGJK      = G(I,J+1,L+1) -G(I,J-1,L+1)
  I         -G(I,J+1,L-1) +G(I,J-1,L-1)
  GX        = A1(I)*DGI
  GY        = -B1(J)*DGJ
  U         = GX -SX(I)*GY +CA*AO(I) +SA*YP
  V         = GY +SA*AO(I) -CA*YP
  W         = RG(I)*(C1(K)*DGK -SZ(I)*GY +SYAW
  I         +CA*XZ(K) +SA*YZ(K) +H*(U*AZ +V*BZ))

```

```

AU      = U +W*AZ
AV      = V +W*BZ
QXY     = H*(U+U +V*V)
QQ      = QXY +W*K
AA      = DIM(AAQ,2*QQ)
HZ      = AZ*SX(I) -BZ +FH*SZ(I)
FXX     = 1. +H*AZ*AZ
FTY     = 1. +SX(I)*SX(I) +H*HZ*HZ
FXY     = SX(I) +H*AZ*HZ
BV      = AV -AU*SX(I) -FH*W*SZ(I)
UU      = H*AI*AU
VV      = H*BV*BV
WW      = FH*W*W
UV      = H*AU*BV
Uw      = AU*W
Vw      = BV*W
AXX     = R1(I)*(FXX*AA -LU)
AZZ     = FH*AA -Ww
AXZ     = (R1(I) +R1(I))*(AZ*AA -Uw)
R       = -(AXX*SX(I) +AZZ*SZ(I) +AXZ*SXZ(I))*GY
1       +T1*(AA*(CZ*GX +LZ -SX(I)*CZ)*GY)
2       -H*(CA*(AU*AU -AV*AV) +(SA +SA)*AU*AV)
3       -QXY*(U*AC(I) +V*TP)
4       +(W +W)*(AC(I)*AZ +YP*BZ))
5       -Ww*(CA*XZZ(K) +SA*YZZ(K)) -W*W*(U*CZ +V*UZ))
AXT     = ABS(AU*AI(I))
AYT     = ABS(BV*BI(J))
AZT     = ABS(FH*W*CI(K))
A       = RO(I)*BETA*AA/*MAX1(AXT,AYT,AZT,(1. -RO(I)))
AXT     = A*AXT
AYT     = A*AYT
AZT     = A*AZT
IF (QQ,GE,AA) GO TO 33
AXX     = AXX*AZ(I)
Ayy     = (FYY*AA -VV)*BZ(J)
AZZ     = AZZ*CZ(K)
AXY     = -R1(I)*(FXY*AA +UV)*(AB +AB)
AXZ     = AXZ*AC
AYZ     = -R1(I)*(HZ*AA +Vw)*(BC +BC)
BP      = AXX
BM      = AXX
B       = -AXX -AXX -Q1*(Ayy +AZZ)
R       = AXX*DGII +Ayy*DGJJ +AZZ*DGKK
1       +AXY*DGII +AYZ*DGJK +AXZ*DGJK +R
GO TO 35
33 NS   = NS +1
S       = SIGN(1.,U)
IM      = I -IFIX(S)
IMM     = IM -IFIX(S)
AXX     = UU*AZ(I)
Ayy     = VV*BZ(J)
AZZ     = WW*CZ(K)
AXY     = S.*S*UV*AB
AXZ     = S.*S*Uw*AC

```

```

AYZ      = B.*Vx+BC
BXX      = (FXX+QQ -UU)*A2(I)
BYY      = (FYY+QQ -VV)*B2(J)
BZZ      = (FH+QQ -WW)*C2(K)
BXY      = -(FXY+QQ +UV)*(AB +AB)
BAZ      = (AZ+QQ -UW)*(AC +AC)
BYZ      = -(HZ+QQ +VW)*(BC +BC)
AG       = AA/QQ
DELTA G  = BXX+DGII +BYY+DGJJ +BZZ+DGKK
1        +BXY+DGIJ +BYZ+DGJK +BAZ+DGIK
DGII     = G(I,J,L) -G(IM,J,L) -G(IM,J,L) +G(IMM,J,L)
1        +A3(I)*DGI
DGJJ     = G(I,J,L) -G(I,J-1,L) -G(I,J-1,L) +GK1(I,J-2)
1        -B2(J)*DGJ
DGKK     = G(I,J,L) -G(I,J,L-1) -G(I,J,L-1) +GK2(I,J)
1        +C3(K)*DGK
DGIJ     = G(I,J,L) -G(IM,J,L)
1        -G(I,J-1,L) +G(IM,J-1,L)
DGIK     = G(I,J,L) -G(I,J,L-1)
1        -G(IM,J,L) +G(IM,J,L-1)
DGJK     = G(I,J,L) -G(I,J,L-1)
1        -G(I,J-1,L) +G(I,J-1,L-1)
GSS      = AXX*DGII +AYY*DGJJ +AZZ+DGKK
1        +AXY*DGIJ +AYZ+DGJK +AXZ+DGIK
B        = .5*(AG -1.)*(AXX +AXX +AXY +AXZ)
BP       = AG*BXX -(1. -S)*B
BM       = AG*BXX -(1. +S)*B
S        = -AC*(BXY +BXX +C2*(BYY +BZZ))
1        +(AG -1.)*(2.*(AXX +AYY +AZZ) +AXY +AYZ +AXZ)
R        = (AG -1.)*GSS +AQ*DELTA G +R
35 IF (ABS(R).LE.ABS(FR)) GO TO 37
FR       = R
IR       = I
JR       = J
KR       = K
37 K     = R -AYT*(GK1(I,J-1) -G(I,J-1,L))
1        -AZT*(GK1(I,J) -G(I,J,L-1))
B        = B -AXT -AYT -AZT
BM       = BM +AXT
S        = 1./(B -BM*C(I-1))
C(I)     = S*BP
32 D(I)  = S*(R -BM*O(I-1))
CG       = C.
I        = I3
DU 42 M=I1,I3
CG       = D(I) -C(I)*CG
IF (ABS(CG).LE.ABS(DG)) GO TO 43
DG       = CG
IG       = I
JG       = J
KG       = K
43 GK2(I,J) = GK1(I,J)
GK1(I,J) = G(I,J,L)
G(I,J,L) = G(I,J,L) -CG

```



```

42 I      = I  -1
   J      = J  +1
   IF (J .GT. 31,51,61)
51 IF (I2.GT.ITE2(K)) I3 = ITE2(K)
   IF (ITE2(K).EQ.MX) I3 = LX
   DO 52 I=I1,I3
   LV      = IABS(1 -IABS(IV(I,K)))
   RO(I)   = AMINO(LV,IABS(IV(I,K)))
52 R1(I)   = LV
   GO TO 31
61 N      = NO
   I      = LX +1
   IF (K.LT.KTE1.OR.K.GT.KTE2) GO TO 71
   IO     = NX +2 -I3
   DO 52 I=IO,I3
   A      = 1. -RO(I) +AO(I)*AO(I) +SO(I,K)*SO(I,K)
   H      = RO(I)/A
   FH     = RO(I)*A
   AZ     = -AO(I)*XZ(K) -SO(I,K)*YZ(K)
   BZ     = -AO(I)*YZ(K) +SO(I,K)*XZ(K)
   HZ     = AZ*SX(I) -BZ +FH*SZ(I)
   FYY    = 1. +SX(I)*SX(I) +H*HZ*HZ
   FXY    = SX(I) +H*AZ*HZ
   DGI    = G(I+1,KY,L) -G(I-1,KY,L)
   DGK    = G(I,KY,L+1) -GK2(I,KY)
   V      = SA*AG(I) -CA*SO(I,K)
   U      = A1(I)*DGI +CA*AO(I) +SA*SO(I,K)
   W      = C1(K)*DGK +SYAW +CA*XZ(K) +SA*YZ(K)
62 G(I,KY+1,L) = G(I,KY-1,L)
   I      = IO
   IF (IO.NE.ITE1(K)) GO TO 71
   E      = G(I3,KY,L) -G(IO,KY,L)
   NO     = NO +1
   EO(NO) = EO(NO) +P3*(E -EO(NO))
   N      = NO
71 IF (I.LE.I1) RETURN
   I      = I  -1
   E      = 0.
   IF (IV(I,K).NE.1) GO TO 77
   ZZ     = Z(K) -TYAW*(XC(K) +S1*AO(I)*AO(I))
73 IF (ZZ.GE.ZO(N-1)) GO TO 75
   N      = N  -1
   GO TO 73
75 R      = (ZZ -ZO(N-1))/(ZO(N) -ZO(N-1))
   E      = R*EO(N) +(1. -R)*EO(N-1)
77 M      = NX +2 -I
   G(I,KY+1,L) = G(M,KY-1,L) -E
   G(M,KY+1,L) = G(I,KY-1,L) +E
   GK2(M,KY)  = GK1(M,KY)
   GK1(M,KY)  = G(M,KY,L)
   G(M,KY,L)  = G(I,KY,L) +E
   GO TO 71
END

```

```

SUBROUTINE XSWEEP
COLUMN RELAXATION
COMMON      G(193,26,4),SO(193,35),EO(131),ZO(131),
1          IV(193,35),ITE1(35),ITE2(35),
2          AO(193),A1(193),A2(193),A3(193),
3          BC(25),B1(26),B2(26),B3(26),
4          Z(35),C1(35),C2(35),C3(35),
5          XC(35),XZ(35),XZZ(35),YC(35),YZ(35),YZZ(35),
6          NX,NY,NZ,KTE1,KTE2,LSYM,KSYM,SCAL,SCALZ,
7          YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IG
COMMON/FLO/ STRIP,P1,P2,P3,BETA,FR,IR,JR,KR,OG,IG,JG,KG,NS
COMMON/SWP/ GK1(193,26),GK2(193,26),
1          SX(193),SZ(143),SXX(193),SYZ(193),SZZ(193),
2          RO(193),R1(193),C(193),D(143),
3          G10(26),G20(26),G30(26),G40(26),G1(26),G2(26),
4          I1,I2,K,L,N),LX,MX,KY,MY,T1,AAO,Q1,Q2,TYAW,S1
N          = ND
J1         = 2
IF (FMACH.GE.1.) J1 = 3
C(J1-1)    = 0.
D(J1-1)    = C.
S          = 1.
I1         = 1
I          = I2 + 1
DO 12 J=2,KY
RC(J)      = 1.
R1(J)      = 1.
G1(J)      = G10(J)
12 G2(J)    = G20(J)
21 IP      = I + I1
IM         = I - I1
J2         = KY
IF (IV(I,K).LT.2.AND.I.GT.LX) J2 = NY
LV         = IABS(I) - IABS(IV(I,K))
RC(KY)     = AMIN0(LV,IABS(IV(I,K)))
R1(KY)     = LV
AC         = T1*A1(I)*C1(K)
DO 32 J=J1,J2
AB         = -T1*A1(I)*B1(J)
BC         = -T1*B1(J)*C1(K)
YP         = SO(I,K) +dO(J)
A          = 1. -RC(J) +AC(I)*AO(I) +YP*YP
H          = RC(J)/A
FH         = R1(J)*A
P          = AO(I)*(4.*YP+YP -FH)
C          = YP*(4.*AC(I)*AO(I) -FH)
A          = XZ(K)*XZ(K) -YZ(K)*YZ(K)
B          = (XZ(K) +XZ(K))*YZ(K)
AZ         = -AO(I)+XZ(K) -YP*YZ(K)
BZ         = -AC(I)*YZ(K) +YP*XZ(K)
CZ         = H*H*(P*A -Q*B) -AL(1)*XZZ(K) -YP*YZZ(K)
DZ         = H*H*(Q*A +P*B) -AO(I)*YZZ(K) +YP*XZZ(K)
DGI        = S*(G(IP,J,L) -G1(J))
DGI        = G(I,J+1,L) -G(I,J-1,L)

```

```

DGK      = G(I,J,L+1) -GK1(I,J)
DGII     = G(I+1,J,L) -G(I,J,L) -G(I,J,L) +G(I-1,J,L)
1        +A3(I)*DGI
DGJJ     = G(I,J+1,L) -G(I,J,L) -G(I,J,L) +G(I,J-1,L)
1        -B3(J)*DGJ
DGKK     = G(I,J,L+1) -G(I,J,L) -G(I,J,L) +G(I,J,L-1)
1        +C3(K)*DGK
DGIJ     = G(I+1,J+1,L) -G(I-1,J+1,L)
1        -G(I+1,J-1,L) +G(I-1,J-1,L)
DGIK     = G(I+1,J,L+1) -G(I+1,J,L-1)
1        -G(I-1,J,L+1) +G(I-1,J,L-1)
DGJK     = G(I,J+1,L+1) -G(I,J-1,L+1)
1        -G(I,J+1,L-1) +G(I,J-1,L-1)
GX       = A1(I)*DGI
GY       = -B1(J)*DGJ
U        = GX -SX(I)*GY +CA*AG(I) +SA*YP
V        = GY +SA*AJ(I) -CA*YP
W        = RU(J)*(C1(K)*DGK -SZ(I)*GY +SYAW
1        +CA*XZ(K) +SA*YZ(K) +H*(U*AZ +V*BZ))
AU       = U +W*AZ
AV       = V +W*BZ
QXY      = H*(L*U +V*V)
QQ       = QXY +W*W
AA       = DIM(AAG,2*QG)
HZ       = AZ*SX(I) -BZ +FH*SZ(I)
FXX      = L. +H*AZ*AZ
FYY      = L. +SX(I)*SX(I) +H*HZ*HZ
FXY      = SX(I) +H*AZ*HZ
BV       = AV -AU*SX(I) -FH*W*SZ(I)
UU       = H*AL*AU
VV       = H*BV*BV
WW       = FH*W*W
UV       = H*AU*BV
UW       = AU*W
VW       = BV*W
AXX      = R1(J)*(FXX*AA -LU)
AZZ      = FH*AA -WW
AXZ      = (R1(J) +R1(J))*(AZ*AA -UW)
R        = -(AXX*SX(I) +AZZ*SZZ(I) +AXZ*SXZ(I))*GY
1        +T1*(AA*(CZ*GX +(CZ -SX(I)*CZ)*GY)
2        -H*(CA*(AU*AL -AV*AV) +(SA +SA)*AU*AV
3        -QXY*(U*AC(I) +V*YP
4        +(W +W)*(AO(I)*AZ +YP*BZ)))
5        -W*(CA*XZZ(K) +SA*YZZ(K)) -W*W*(U*CZ +V*DZ))
AXT      = ABS(AU*A1(I))
AYT      = ABS(BV*B1(J))
AZT      = ABS(FH*W*C1(K))
A        = RC(J)*BETA*AA/AMAX1(AX),AYT,AZT,(L. -RC(J))
AXT      = A*AXT
AYT      = A*AYT
AZT      = A*AZT
IF (DQ.GE.AA) GO TO 33
AXX      = AXX*AZ(I)
AYY      = (FYY*AA -VV)*BZ(J)

```

```

AZZ      = AZZ+C2(K)
AXY      = -R1(J)*(FXY*AA +LV)*(AB +A3)
AXZ      = AXZ+AC
AYZ      = -R1(J)*(HZ*AA +VW)*(BC +8C)
BP       = AYY
BM       = AYY
B        = -AYY -AYY -Q1*(AXX +AZZ)
R        = AXX+DGII +AYY+DGJJ +AZZ+DGKK
1        +AYY+DGJJ +AYZ+EGJK +AXZ+DGJK +R
GO FL 35
33 NS    = NS +1
AXX      = UU+A2(I)
AYY      = VV+B2(J)
AZZ      = WW+C2(K)
AXY      = U.*S+UV*AB
AXZ      = U.*S+UV*AC
AYZ      = U.*VW*BC
BXX      = (FXX+QQ -JU)*A2(I)
BYY      = (FYY+QQ -VV)*B2(J)
BZZ      = (F4+QQ -WW)*C2(K)
BXY      = -(FXY+QQ +UV)*(AB +AB)
BXZ      = (AZ+QQ -UW)*(AC +AC)
BYZ      = -(HZ+QQ +VW)*(BC +BC)
AL       = A4/CQ
PDLTAG  = BXX+DGII +BYY+DGJJ +BZZ+DGKK
1        +BXY+DGJJ +BYZ+EGJK +BXZ+DGJK
DGII     = G(I,J,L) -G(IM,J,L) -G(IP,J,L) +G2(J)
1        +A3(I)*DGI
DGJJ     = G(I,J,L) -G(I,J-1,L) -G(I,J-1,L) +G(I,J-2,L)
1        -B3(J)*DGG
DEKK     = G(I,J,L) -G(I,J,L-1) -G(I,J,L-1) +GK2(I,J)
1        +CB(K)*DGK
DGIJ     = G(I,J,L) -G(IM,J,L)
1        -G(I,J-1,L) +G(IM,J-1,L)
DGIK     = G(I,J,L) -G(I,J,L-1)
1        -G(IM,J,L) +G(IM,J,L-1)
DGJK     = G(I,J,L) -G(I,J,L-1)
1        -G(I,J-1,L) +G(I,J-1,L-1)
GSS      = AXX+DGII +AYY+DGJJ +AZZ+DGKK
1        +AAY+DGIJ +AYZ+EGJK +AXZ+DGJK
BP       = AG+BYY
BM       = BP -(AQ -1.)*(AYY +AYY +AXY +AYZ)
B        = -BP -BP -Q2*AG*(EXX +BZZ)
1        +(AC -1.)*(2.*(AXX +AYY +AZZ) +AXY +AYZ +AXZ)
K        = (AC -1.)*GSS +AQ*DELTAG +R
35 IF (ABS(R).LE.ABS(FR)) GO TO 37
FR       = R
IR       = I
JK       = J
KR       = K
37 R     = K -AXT*(G1(J) -G(IM,J,L))
1        -AZT*(GK1(I,J) -G(I,J,L-1))
B        = B -AXT -AYT -AZT
BM       = BM +AYT

```

```

      B          = 1./ (8 -BM*C(J-1))
      C(J)       = B*BP
32  D(J)        = B*(R -BM*D(J-1))
      CG        = 0.
      J         = J2
      DO 42 M=J1,J2
      CG        = D(J) -C(J)*CG
      IF (ABS(CG).LE.ABS(DG)) GO TO 43
      DG        = CG
      IG        = I
      JG        = J
      KG        = K
43  G2(J)       = G1(J)
      G1(J)      = G(I,J,L)
      GK2(I,J)   = GK1(I,J)
      GK1(I,J)   = G(I,J,L)
      G(I,J,L)   = G(I,J,L) -CG
42  J           = J -1
      IF (IV(I,K).LT.2) GO TO 51
      A         = 1. -R0(KY) +A0(I)*A0(I) +SC(I,K)*SO(I,K)
      H         = R0(KY)/A
      FH        = R0(KY)*A
      AZ        = -A0(I)*XZ(K) -SC(I,K)*YZ(K)
      BZ        = -A0(I)*YZ(K) +SO(I,K)*XZ(K)
      HZ        = AZ*SX(I) -BZ +FH*SZ(I)
      FYY       = 1. +SX(I)*SX(I) +H*HZ*HZ
      FXY       = SX(I) +H*AZ*HZ
      DGI       = S*(G(IP,KY,L) -G2(KY))
      DGK       = G(I,KY,L+1) -GK2(I,KY)
      V         = SA*A0(I) -CA*SO(I,K)
      U         = A1(I)*DGI +CA*A0(I) +SA*SO(I,K)
      W         = CI(K)*DGK +SYAW +CA*KZ(K) +SA*YZ(K)
      G(I,KY+1,L) = G(I,KY-1,L)
1      +IV*(1. -H*BZ*HZ) -U*FXY -W*HZ)/(FYY*B1(KY))
      IF (I.NE.ITE1(K)) GO TO 61
      M         = NX +2 -1
      E         = G(M,KY,L) -G(I,KY,L)
      NO        = NO +1
      EO(NO)    = EO(NO) +P3*(E -EO(NO))
      N         = NO
      GO TO 61
51  IF (1.GT.LX) GO TO 61
      E         = 0.
      IF (IV(I,K).NE.1) GO TO 57
      ZZ        = Z(K) -TYAW*(XC(K) +S1*A0(I)*A0(I))
53  IF (ZZ.GE.ZO(N-1)) GO TO 53
      N         = N -1
      GO TO 53
55  R          = (ZZ -ZO(N-1))/(ZO(N) -ZO(N-1))
      E         = R*EO(N) +(1. -R)*EO(N-1)
57  M         = NX +2 -1
      G(I,KY+1,L) = G(M,KY-1,L) -E
      G(M,KY+1,L) = G(I,KY-1,L) +E
      GK2(M,KY)  = GK1(M,KY)

```

```

GK1(M,KY) = G(M,KY,L)
G(M,KY,L) = G(I,KY;L) +E
61 IF (I.EQ.NX) GO TO 71
IF (I.EQ.2) RETURN
I = I +11
GO TO 21
71 S = -1.
I1 = -1
I = I1 -1
DO 72 J=2,KY
G1(J) = G30(J)
72 G2(J) = G40(J)
GL TO 21
END

```

```

C SUBROUTINE VELO (K,L,SV,SM,CP,X,Y)
CALCULATES SURFACE VELOCITY
COMMON G(193,26,4),SO(193,35),EO(131),ZO(131),
1 IV(193,35),ITE1(35),ITE2(35),
2 AO(193),A1(193),A2(193),A3(193),
3 B0(26),B1(26),B2(26),B3(26),
4 Z(35),C1(35),C2(35),C3(35),
5 XC(35),XZ(35),XZZ(35),YC(35),YZ(35),YZZ(35),
6 NX,NY,NZ,KTE1,KTE2,SYM,KSYM,SCAL,SCALZ,
7 YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IO
DIMENSION SV(1),SM(1),CP(1),X(1),Y(1)
I1 = ITE1(K)
I2 = ITE2(K)
J = NY +1
Q1 = .2*FMACH**2
T1 = 1./(.7*FMACH**2)
DO 12 I=I1,I2
FH = AC(I)*AO(I) +SO(I,K)*SO(I,K)
H = 0.
IF (IV(I,K).NE.0) H = 1./FH
AZ = -AC(I)*XZ(K) -SO(I,K)*YZ(K)
BZ = -AC(I)*YZ(K) +SO(I,K)*XZ(K)
DSI = SO(I+1,K) -SO(I-1,K)
DSK = SO(I,K+1) -SO(I,K-1)
SX = A1(I)*DSI
SZ = C1(K)*DSK
DGI = G(I+1,J,L) -G(I-1,J,L)
DGJ = G(I,J+1,L) -G(I,J-1,L)
DGK = G(I,J,L+1) -G(I,J,L-1)
U = A1(I)*DGJ +SX*B1(J)*DGJ +CA*AO(I) +SA*SO(I,K)
V = -B1(J)*DGJ +SA*AO(I) -CA*SO(I,K)
W = C1(K)*DGK +SZ*B1(J)*DGJ +SYAW
1 +CA*XZ(K) +SA*YZ(K) +H*(L*AZ +V*BZ)
QQ = H*(U*U +V*V) +W*W
SV(I) = SIGN(SORT(QQ),U)
IF (IV(I,K).EQ.0) SV(I) = SV(I-1) +SV(I-1) -SV(I-2)

```

```

      QQ      = 1. +Q1*(1. -QQ)
      SM(I)   = FMACH*SV(I)/SQRT(CQ)
      CP(I)   = T1*(QQ**3.5 -1.)
      X(I)    = XC(K) +.5*SCAL*(AO(I)+AO(I) -SJ(I,K)*SO(I,K))
12  Y(I)     = YC(K) +SCAL*AO(I)*SO(I,K)
      RETURN
      END

```

```

C      SUBROUTINE CPLCT (I1,I2,FMACH,X,Y,CP)
      PLOTS CP AT EQUAL INTERVALS IN THE MAPPED PLANE
      DIMENSION  KODE(2),LINE(100),X(1),Y(1),CP(1)
      DATA      KODE/1H ,1H+/
      IWRIT      = 6
      WRITE (IWRIT,2)
2  FORMAT(50#OPLGT OF CP AT EQUAL INTERVALS IN THE MAPPED PLANE/
1  10HO      X      ,10H      Y      ,10H      CP )
      CPO      = ((1. +.2*FMACH**2)**3.5 -1.)/(1.7*FMACH**2)
      DO 12 I=1,100
12  LINE(I)   = KODE(1)
      DO 22 I=1,12
      K        = 30.*(CPO -CP(I)) +4.5
      K        = MINO(100,K)
      LINE(K)  = KODE(2)
      WRITE (IWRIT,610) X(I),Y(I),CP(I),LINE
22  LINE(K)   = KODE(1)
      RETURN
610  FORMAT(3F10.4,100A1)
      END

```

```

C      SUBROUTINE FORCF (I1,I2,X,Y,CP,AL,CHORD,XM,CL,CD,CM)
      CALCULATES SECTION FORCE COEFFICIENTS
      DIMENSION  X(1),Y(1),CP(1)
      RAD        = 57.2957795130823
      ALPHA      = AL/RAD
      CL         = C.
      CD         = 0.
      CM         = 0.
      N          = I2 -1
      DO 12 I=1,N
      DX         = (X(I+1) -X(I))/CHORD
      DY         = (Y(I+1) -Y(I))/CHORD
      XA        = (.5*(X(I+1) +X(I)) -XM)/CHORD
      YA        = .5*(Y(I+1) +Y(I))/CHORD
      CPA       = .5*(CP(I+1) +CP(I))
      DCL       = -CPA*DX
      DCD       = CPA*DY
      CL        = CL +DCL
      CD        = CD +DCD

```

```

12 CM      = CM + DCF*YA - OCL*XA
DCL       = CL*COS(ALPHA) - CD*SIN(ALPHA)
CD        = CL*SIN(ALPHA) + CD*COS(ALPHA)
CL        = OCL
RETJKN
END

```

```

SUBROUTINE TOTFOR(KTE1,KTE2,CHORD,SCL,SCD,SCM,Z,XC,
1          CL,CD,CMP,CMR,CMY)
C  CALCULATES TOTAL FORCE COEFFICIENTS
DIMENSION CHORD(1),SCL(1),SCD(1),SCM(1),Z(1),XC(1)
SPAN      = Z(KTE2) - Z(KTE1)
CL        = 0.
CD        = 0.
CMP       = 0.
CMR       = 0.
CMY       = 0.
S         = 0.
N         = KTE2 - 1
DO 12 K=KTE1,N
DZ        = .5*(Z(K+1) - Z(K))
AZ        = .5*(Z(K+1) + Z(K))
CL        = CL + DZ*(SCL(K+1)*CHORD(K+1) + SCL(K)*CHORD(K))
CD        = CD + DZ*(SCD(K+1)*CHORD(K+1) + SCD(K)*CHORD(K))
CMP       = CMP + DZ*(CHORD(K+1)*(SCM(K+1)*CHORD(K+1)
1          - SCL(K+1)*XC(K+1))
2          + CHORD(K)*(SCM(K)*CHORD(K)
3          - SCL(K)*XC(K)))
CMR       = CMR + AZ*DZ*(SCL(K+1)*CHORD(K+1) + SCL(K)*CHORD(K))
CMY       = CMY + AZ*DZ*(SCD(K+1)*CHORD(K+1) + SCD(K)*CHORD(K))
12, S     = S + DZ*(CHORD(K+1) + CHORD(K))
CL        = CL/S
CD        = CD/S
CMP       = CMP*SPAN/S**2
CMR       = (CMP + CMR)/(S*SPAN)
CMY       = (CMY + CMY)/(S*SPAN)
RETURN
END

```

```

SUBROUTINE REFIN
C  HALVES MESH SIZE
COMMON G(193,26,4),SO(193,35),E0(131),Z0(131),
1 IV(193,35),IT1(35),IT2(35),
2 A0(193),A1(193),A2(193),A3(193),
3 B0(26),B1(26),B2(26),B3(26),
4 Z(35),C1(35),C2(35),C3(35),
5 XC(35),XL(35),XZ(35),YC(35),YZ(35),YZZ(35),
6 NX,NY,NZ,KTE1,KTE2,ISYM,KSYM,SCAL,SCAL2,

```



```

7          YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IG
MX        = NX +1
KY        = NY +1
MY        = NY +2
MZ        = NZ +3
MXO       = NX/2 +1
MYO       = NY/2 +2
MZO       = NZ/2 +1
K         = 1
IF (KSYM.EQ.0) GO TO 11
MZO      = NZ/2 +3
BUFFER IN (N1,1) (G(1,1,1),G(MXO,MYO,1))
IF (UNIT(N1).GT.0.) GO TO 401
K        = 2
11 BUFFER IN (N1,1) (G(1,1,1),G(MXO,MYO,1))
IF (UNIT(N1).GT.0.) GO TO 401
J        = NY/2 +1
JJ       = KY
21 i      = MXO
II       = MX
31 G(II,JJ,1) = G(1,J,1)
I        = I -1
II       = II -2
IF (I.GT.0) GO TO 31
J        = J -1
JJ       = JJ -2
IF (J.GT.0) GO TO 21
DD 42 J=1,KY,2
DD 42 I=2,NX,2
42 G(I,J,1) = .5*(G(I+1,J,1) +G(I-1,J,1))
DD 52 I=1,MX
DD 54 J=2,NY,2
54 G(I,J,1) = .5*(G(I,J+1,1) +G(I,J-1,1))
52 G(I,NY,1) = 0.
BUFFER OUT(N2,1) (G(1,1,1),G(MX,MY,1))
IF (UNIT(N2).GT.0.) GO TO 401
K        = K +1
IF (K.LE.MZO) GO TO 11
REWIND N1
REWIND N2
BUFFER IN (N2,1) (G(1,1,1),G(MX,MY,1))
IF (UNIT(N2).GT.0.) GO TO 401
BUFFER IN (N2,1) (G(1,1,3),G(MX,MY,3))
IF (UNIT(N2).GT.0.) GO TO 401
BUFFER OUT(N1,1) (G(1,1,1),G(MX,MY,1))
IF (UNIT(N1).GT.0.) GO TO 401
K        = 1
IF (KSYM.NE.0) K = 2
111 K     = K +1
DD 112 J=1,MY
DD 112 I=1,MX
112 G(I,J,2) = .5*(G(I,J,1) +G(I,J,3))
DD 122 L=2,3
BUFFER OUT(N1,1) (G(1,1,L),G(MX,MY,L))

```

```

IF (UNIT(N1).GT.0.) GO TO 401
222 CONTINUE
IF (K.EQ.M20) GO TO 201
DO 132 J=1,MY
DO 122 I=1,MX
132 G(I,J,1) = G(I,J,3)
BUFFER IN (N2,1) (G(I,1,3),G(MX,MY,3))
IF (UNIT(N2).GT.0.) GO TO 401
GO TO 111
201 REWIND N1
REWIND N2
DO 202 L=1,3
BUFFER IN (N1,1) (G(I,1,L),G(MX,MY,L))
IF (UNIT(N1).GT.0.) GO TO 401
202 CONTINUE
BUFFER OUT(N2,1) (G(I,1,1),G(MX,MY,1))
IF (UNIT(N2).GT.0.) GO TO 401
YAW = SYAW/CYAW
SI = .5*SCAL
NU = KTE1 - 1
EG(NC) = 0.
K = 2
IF (KSYM.NE.0) GO TO 251
211 K = NO
I = MX0 + 1
IF (K.LT.KTE1.OP.K.GT.KTE2) GO TO 231
I1 = ITE1(K)
I2 = ITE2(K)
DO 212 I=I1,I2
DSI = SO(I+1,K) -SO(I-1,K)
DSK = SO(I,K+1) -SO(I,K-1)
SX = A1(I)*DSI
SZ = C1(K)*DSK
R = AMINC(1,IV(I,K))
A = 1. -R +AJ(I)*AG(I) +SO(I,K)*SO(I,K)
H = R/A
FH = R*A
AZ = -AO(I)*XZ(K) -S(I,K)*YZ(K)
BZ = -AO(I)*YZ(K) +SO(I,K)*XZ(K)
HZ = AZ*SX -BZ +FH*SZ
FYY = 1. +SX*SX +H*HZ*HZ
FXY = SX +H*BZ*HZ
DGI = G(I+1,KY,2) -G(I-1,KY,2)
DGK = G(I,KY,3) -G(I,KY,1)
V = SA*AC(I) -CA*SO(I,K)
U = A1(I)*DGI +CA*AC(I) +SA*SU(I,K)
W = C1(K)*DGK +SYAW +CA*XZ(K) +SA*YZ(K)
212 G(I,KY+1,2) = G(I,KY-1,2)
1 + (V*(1. -H*BZ*HZ) -U*FXY -W*HZ)/(FYY+B1(KY))
NO = NO + 1
EG(NC) = G(I2,KY,2) -G(I1,KY,2)
N = NO
I = I1
IF (K.NE.KTE2.OP.YAW.LE.0.) GO TO 231

```

```

221 I      = I  +1
    M      = NX +2 -I
    NC     = NO  +1
    EO(NC) = G(M,KY,2) -G(I,KY,2)
    IF (I.LT.MX0) GO TO 221
    J      = J1
231 I      = I  -1
    E      = 0.
    IF (IV(I,K).NE.1) GO TO 237
    ZZ     = Z(K) -IYAW*(XC(K) +S1*AO(I)+AO(I))
233 IF (ZZ.GE.ZO(N-1)) GO TO 235
    N      = N  -1
    GO TO 233
235 F      = (ZZ -ZO(N-1))/(ZO(N) -ZO(N-1))
    E      = R*E(I,N) +(1. -P)*EO(N-1)
237 M      = NX +2 -I
    G(I,KY+1,2) = G(M,KY-1,2) -E
    G(M,KY+1,2) = G(I,KY-1,2) +E
    IF (IV(I,K).NE.-1) GO TO 241
    G(I,KY,2)   = .5*G(I,KY,1) +.25*(G(I,KY,3) +G(M,KY,3))
    IF (IV(I,K+1).LT.1)
1G(I,KY,2)    = .5*G(I,KY,3) +.25*(G(I,KY,1) +G(M,KY,1))
    G(M,KY,2)   = G(I,KY,2)
    G(I,KY-1,2) = .5*(G(I,KY,2) +G(I,KY-2,2))
    G(M,KY-1,2) = .5*(G(M,KY,2) +G(M,KY-2,2))
241 IF (I.GT.2) GO TO 231
251 K      = K  +1
    IF (K.EQ.MZ) GO TO 261
    DO 252 J=1,MY
    DO 252 I=1,MX
    G(I,J,1)   = G(I,J,2)
252 G(I,J,2)   = G(I,J,3)
    BUFFER OUT(NZ,1) (G(I,1,1),G(MX,MY,1))
    IF (UNIT(NZ).GT.0.) GO TO 401
    BUFFER IN (N1,1) (G(I,1,3),G(MX,MY,3))
    IF (UNIT(N1).GT.0.) GO TO 401
    GO TO 211
261 EG(NC+1) = 0.
    DO 262 L=2,3
    BUFFER OUT(NZ,1) (G(I,1,L),G(MX,MY,L))
    IF (UNIT(NZ).GT.0.) GO TO 401
262 CONTINUE
    REWIND N1
    REWIND N2
    DO 302 K=1,MZ
    BUFFER IN (N2,1) (G(I,1,1),G(MX,MY,1))
    IF (UNIT(N2).GT.0.) GO TO 401
    BUFFER OUT(N1,1) (G(I,1,1),G(MX,MY,1))
    IF (UNIT(N1).GT.0.) GO TO 401
302 CONTINUE
    IC     = 1
    RETURN
401 ID    = 0
    RETURN
    END

```

```

SUBROUTINE SMOO
SMOOTHS POTENTIAL
COMMON      G(193,26,4),SO(193,35),EO(131),ZO(131),
1          IV(193,35),ITE1(35),ITE2(35),
2          AU(193),A1(193),A2(193),A3(193),
3          EO(26),E1(26),E2(26),E3(26),
4          Z(35),C1(35),C2(35),C3(35),
5          XC(35),XZ(35),XZZ(35),YC(35),YZ(35),YZZ(35),
6          NX,NY,NZ,KTC1,KTC2,ISYM,KSYSM,SCAL,SCALZ,
7          YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IO
MX          = NX  +1
MY          = NY  +1
MZ          = NY  +2
K1          = 2
K2          = NZ
IF (KSYSM.EQ.0) GO TO 1
K1          = 3
K2          = NZ  +2
1 PX        = 1./6.
PY          = 1./6.
PZ          = 1./6.
DO 2 L=1,3
BUFFER IN (N1,1) (G(1,1,L),G(MX,MY,L))
IF (LIMIT(N1).GT.0.) GO TO 51
2 CONTINUE
BUFFER OUT(N2,1) (G(1,1,1),G(MX,MY,1))
IF (LIMIT(N2).GT.0.) GO TO 51
K          = K1
11 K        = K  +1
DO 12 J=3,MY
DO 14 I=2,NX
14 G(I,J,4) = (1. -PX -PY -PZ)*G(I,J,2)
1          +.5*PX*(G(I+1,J,2) +G(I-1,J,2))
2          +.5*PY*(G(I,J+1,2) +G(I,J-1,2))
3          +.5*PZ*(G(I,J,3) +G(I,J,1))
G(1,J,4) = G(1,J,2)
12 G(MX,J,4) = G(MX,J,2)
DO 16 I=1,MX
G(I,1,4) = G(I,1,2)
G(I,2,4) = G(I,2,2)
G(I,KY,4) = G(I,KY,2)
16 G(I,MY,4) = G(I,MY,2)
BUFFER OUT(N2,1) (G(1,1,4),G(MX,MY,4))
IF (LIMIT(N2).GT.0.) GO TO 51
IF (K.EQ.K2) GO TO 31
DO 22 J=1,MY
DO 22 I=1,MX
G(I,J,1) = G(I,J,2)
22 G(I,J,2) = G(I,J,3)
BUFFER IN (N1,1) (G(1,1,3),G(MX,MY,3))
IF (LIMIT(N1).GT.0.) GO TO 51
GO TO 11
31 BUFFER OUT(N2,1) (G(1,1,3),G(MX,MY,3))

```

```

IF (UNIT(N2).GT.0.) GO TO 51
REWIND N1
REWIND N2
DO 42 K=1,MZ
BUFFER IN (N2,1) (G(1,1,1),G(MX,MY,1))
IF (UNIT(N2).GT.0.) GO TO 51
BUFFER OUT(N1,1) (G(1,1,1),G(MX,MY,1))
IF (UNIT(N1).GT.0.) GO TO 51
42 CONTINUE
ID      = 1
RETURN
51 ID      = 0
RETURN
END

```

```

SUBROUTINE SPLIF(M,N,S,F,FP,FPP,FPPP,KM,VM,KN,VN,MODE,FQM,IND)
C SPLINE FIT - JAMESON
C INTEGRAL PLACED IN FPPP IF MODE GREATER THAN 0
C IND SET TO ZERO IF DATA ILLLEGAL
DIMENSION S(1),F(1),FP(1),FPP(1),FPPP(1)
IND      = 0
K        = 1ABS(N -M)
IF (K -1) 81,81,1
1 K      = (N -M)/K
I        = 0
J        = M +K
DS       = S(J) -S(I)
D        = DS
IF (DS) 11,81,11
11 DF    = (F(J) -F(I))/DS
IF (KM -2) 12,13,14
12 U     = 0
V       = 3.+(DF -VM)/DS
GO TO 25
13 U     = 0.
V       = VM
GO TO 25
14 U     = -1.
V       = -DS*VM
GO TO 25
21 I     = 0
J       = J +K
DS      = S(J) -S(I)
IF (D*DS) 81,81,23
23 DF    = (F(J) -F(I))/DS
B       = 1./(DS +DS +U)
U       = 2*DS
V       = B*(2.*DF -V)
25 FP(I) = U
FPP(I)  = V
U       = (2. -U)*DS

```

```

      V      = 6.*LF +DS*V
      IF (J -N) 21,31,21
31  IF (KN -2) 32,33,34
32  V      = (6.*VN -V)/U
      GO TO 35
33  V      = VN
      GO TO 35
34  V      = (DS*VN +FPP(I))/(1. +FP(I))
35  B      = V
      D      = DS
41  DS     = S(J) -S(I)
      U     = FPP(I) -FP(I)*V
      FPPP(I) = (V -U)/C3
      FPP(I)  = U
      FP(I)   = (F(J) -F(I))/DS -DS*(V +U +U)/E.
      V      = U
      J      = I
      I      = I -K
      IF (L -M) 41,51,41
51  I      = N -K
      FPPP(N) = FPPP(I)
      FPP(N)  = B
      FP(N)   = DF +D*(FPP(I) +U +B)/E.
      INO    = 1
      IF (MODE) 61,81,61
61  FPPP(J) = FOM
      V     = FPP(J)
71  I      = J
      J     = J +K
      DS    = S(J) -S(I)
      U     = FPP(J)
      FPPP(J) = FPPP(I) +.5*DS*(F(I) +F(J) -DS*DS*(U +V)/12.)
      V     = U
      IF (J -N) 71,81,71
81  RETURN
      END

```

```

C  SUBROUTINE INTPL(MI,NI,SI,FI,M,N,S,F,FP,FPP,FPPP,MODE)
C  INTERPOLATION USING TAYLOR SERIES - JAMESON
C  ADDS CORRECTION FOR PIECEWISE CONSTANT FOURTH DERIVATIVE
C  IF MODE GREATER THAN 0
      DIMENSION SI(1),FI(1),S(1),F(1),FP(1),FPP(1),FPPP(1)
      K      = IABS(N -M)
      K      = (N -M)/K
      I      = M
      MIN    = MI
      NIN    = NI
      D      = S(N) -S(M)
      IF (D*(SI(NI) -SI(MI))) 11,13,13
11  MIN    = NI
      NIN    = MI

```

```

13 KI      = IABS(NIN -MIN)
   IF (KI) 21,21,15
15 KI      = (NIN -MIN)/KI
21 II      = MIN -KI
   C      = 0.
   IF (PDE) 31,31,23
23 C      = 1.
31 II      = II +KI
   SS     = S1(II)
33 I      = I +K
   IF (I -N) 35,37,35
35 IF (C*(S(I) -SS)) 32,33,37
27 J      = I
   I      = I -K
   SS     = SS -S(I)
   FPPPF  = C*(FPPP(J) -FPPP(I))/(S(J) -S(I))
   FF     = FPPP(I) +.25*SS*FPPPF
   FF     = FPP(I) +SS*FF/3.
   FF     = FP(I) +.5*SS*FF
   FI(II) = F(I) +SS*FF
   IF (II -NIN) 31,41,21
41 RETURN
   END

```

```

SUBROUTINE RPLLT (IPLOT,NRES,RES,COUNT,TITLE,FMACH,YA,AL,
1          N1,N2,N3)
C PLOTS CONVERGENCE RATE
DIMENSION RES(1),COUNT(1),TITLE(20),R(20)
IF (NRES.LE.1) RETURN
IF (IPLOT.EQ.0) GO TO 11
CALL PLOTSBL(1000,24HANTONY JAMESON 109604R)
CALL PLOT(1.,25,1.,-3)
11 IPLOT = 0
   RATE = (ABS(RES(NRES)/RES(1)))
   1     **(.7/(COUNT(NRES) -COUNT(1)))
   ENCODE(80,12,R) TITLE
12 FORMAT(20A4)
   CALL SYMBOL(1.,.5,.14,R,0.,80)
   ENCODE(50,14,R) FMACH,YA,AL
14 FORMAT(5HMACH ,F9.3,4X,5HYA ,F9.3,4X,5HALPHA,F9.3)
   CALL SYMBOL(1.,.25,.14,R,0.,50)
   ENCODE(32,16,R) RES(1),RES(NRES)
16 FORMAT(5HRES1 ,E9.3,4X,5HRES2 ,E9.3)
   CALL SYMBOL(1.,0.,.14,R,0.,32)
   ENCODE(50,18,R) COUNT(1),COUNT(NRES),RATE
18 FORMAT(5HGRK1 ,F9.2,4X,5HGRK2 ,F9.2,4X,5HRATE ,F9.4)
   CALL SYMBOL(1.,-.25,.14,R,0.,50)
   ENCODE(24,20,R) N1,N2,N3
20 FORMAT(6HGRID ,I4,3H X ,I4,3H X ,I4)
   CALL SYMBOL(1.,-.5,.14,R,0.,24)
   RMIN = 0.

```

```

RMAX      = 0.
COUNT1   = COUNT(1)
RES1      = RES(1)
DO 22 I=1,NRES
COUNT(I) = COUNT(I) -COUNT1
RES(I)    = ALOG(ABS(RES(I)/RES1))
RMAX      = AMAX1(RMAX,RES(I))
22 RMIN    = AMIN1(RMIN,RES(I))
YSCAL     = 1./ALOG(10.)
YINT      = 1.
IF (YSCAL*RMIN.LT.-6.) YINT = 2.
YLOW      = -6.*YINT
YSCAL     = YSCAL/YINT
XINT      = 50.
IF (COUNT(NRES).GT.300.) XINT = 100.
IF (COUNT(NRES).GT.600.) XINT = 200.
IF (COUNT(NRES).GT.1200.) XINT = 500.
IF (COUNT(NRES).GT.6000.) XINT = 1000.
KSCAL     = 1./XINT
CALL PLOT(1.5,4.5,-3)
CALL AXIS(0.,-3.,10HLOG(EKKDF),10,0.,90.,YLOW,YINT,0)
CALL PLOT(3.,-3.,-3)
CALL AXIS(-3.,0.,4HNCYC,-4,0.,0.,0.,XINT,0)
DO 32 I=1,NRES
COUNT(I) = KSCAL*COUNT(I) -3.
32 RES(I)  = AMIN1(2.,YSCAL*RES(I)) +6.
CALL LINE(COUNT,RES,NRES,1,0,1,0.,1,0.,1.)
CALL PLOT(8.5,-1.5,-3)
RETURN
END

```

```

SUBROUTINE GRAPH (IPLOT,I1,I2,X,Y,CP,TITLE,FMACH,YA,AL,
1              Z,CL,CD,CHGRUO,KSCAL,PSCAL)
C GENERATES CALCOMP PLOTS
DIMENSION X(1),Y(1),CP(1),TITLE(20),R(20)
IF (IPLOT.FQ.0) GO TO 11
CALL PLOTSBL(1000,24HANTONY JAMESON 109604R)
CALL PLOT(1.25,1.,-3)
11 IPLOT = 0
ENCODE(80,12,R) TITLE
12 FORMAT(20A4)
CALL SYMBOL(.5,C.,.14,R,0.,80)
ENCODE(44,14,R) FMACH,YA,AL
14 FORMAT(5HFMACH ,F7.3,4X,5HYAW ,F7.3,4X,5HALPHA,F7.3)
CALL SYMBOL(.5,-.25,.14,R,0.,44)
ENCODE(44,16,P) Z,CL,CD
16 FORMAT(5HZ ,F7.2,4X,5HCL ,F7.4,4X,5HCD ,F7.4)
CALL SYMBOL(.5,-.5,.14,R,0.,44)
XMAX      = X(I1)
XMIN      = X(I2)
YMIN      = Y(I1)

```



```

DO 22 I=I1,I2
XMAX      = AMAX1(X(I),XMAX)
XMIN      = AMIN1(X(I),XMIN)
22 YMIN    = AMIN1(Y(I),YMIN)
SCALX     = 5./(XMAX -XMIN)
IF (XSCAL.GT.0.) SCALX = XSCAL/(XMAX -XMIN)
IF (XSCAL.LT.0.) SCALX = ABS(XSCAL)/CHGRDO
PINT      = -.4
IF (PSCAL.NE.0.) PINT = -ABS(PSCAL)
SCALP     = 1./PINT
PMIN      = -3.*PINT
PMAX      = 5.*PINT
DO 24 I=I1,I2
X(I)      = SCALX*(X(I) -XMIN) +.5
24 Y(I)    = SCALX*(Y(I) -YMIN) +.5
CPMAX     = 0.
IMAX      = (I2 +I1)/2
N         = (I2 -I1)/8
N1        = IMAX -N
N2        = IMAX +N
DO 26 I=N1,N2
IF (CP(I).LE.CPMAX) GO TO 24
CPMAX     = CP(I)
IMAX      = I
26 CONTINUE
N         = I2 -I1 +1
CALL LINE(X(I1),Y(I1),N,1,0,1,0.,1.,0.,1.)
CALL PLOT(0.,4.5,-3)
CALL AXIS(0.,-3.,2HCP,2,8.,90.,PMIN,PINT,0)
CPC       = ((15. +FMACH**2)/6.)**3.0 -1./(.7*FMACH**2)
IF (CPC.GE.PMAX) CALL SYMBOL(0.,SCALP*CPC,.40,15,0.,-1)
DO 32 I=I1,IMAX
IF (CP(I).LT.PMAX) GO TO 32
CALL SYMBOL(X(I),SCALP*CP(I),.07,3,45.,-1)
32 CONTINUE
DO 34 I=IMAX,I2
IF (CP(I).LT.PMAX) GO TO 34
CALL SYMBOL(X(I),SCALP*CP(I),.07,3,0.,-1)
34 CONTINUE
CALL PLOT(12.,-4.5,-3)
RETURN
END

```

```

SUBROUTINE THREED(IPLDT,SV,SM,CP,X,Y,TITLE,YA,AL,
1          VLD,CL,CD,CHGRDO,XSCAL,PSCAL)
C GENERATES THREE DIMENSIONAL PLOTS
COMMON    G(193,26,4),SU(193,35),E0(131),Z0(131),
1          IV(193,35),IFE1(35),ITE2(35),
2          AO(193),A1(193),A2(193),A3(193),
3          BU(26),B1(26),B2(26),B3(26),
4          Z(35),C1(35),C2(35),C3(35),

```

```

5          XC(35),XZ(35),XZ2(35),YG(35),YZ(35),YZZ(35),
6          NX,NY,NZ,KTE1,KTE2,ISYM,KSYM,SCAL,SCALZ,
7          YA,CYA,SYA,ALPHA,CA,SA,FMACH,N1,N2,N3,IO
  DIMENSION X(1),Y(1),SV(1),SM(1),CP(1),TITLE(20),R(20)
  LX      = NX/2 +1
  PX      = NX +1
  MY      = NY +2
  IF (XSCAL.NE.0.) SCALX = .5*ABS(XSCAL)/CMORDO
  IF (PSCAL.GE.0.) SCALX = 5./((Z(KTE2) -Z(KTE1)))
  SCALP   = -1.25
  IF (FSCAL.NE.0.) SCALP = -.5/ABS(PSCAL)
  SX      = 2. -SCALX*XC(KTE1)
  TX      = 3.5
  IF (IPL0T.EQ.0) GO TO 1
  CALL PLOTSBL(1000,24MANTONY JAMESON 109004R)
  CALL PLOT(1,25,)-3)
1  IPL0T   = 0
   M      = 1
   ENCODE(12,2,R)
2  FORMAT(12HVIEW OF WING)
   CALL SYMBOL(2,0,0,14,R,0,12)
11 DO I=1,3
   BUFFER IN (N1,1) (G(1,I,L),G(MX,MY,L))
   IF (UNIT(N1).GT.0.) GO TO 101
12 CONTINUE
   K      = 2
21 K      = K +1
   IF (K.GT.KTE2) GO TO 61
   DO 22 J=1,MY
   DO 22 I=1,MX
   G(I,J,1) = G(I,J,2)
22 G(I,J,2) = G(I,J,3)
   BUFFER IN (N1,1) (G(1,I,3),G(MX,MY,3))
   IF (UNIT(N1).GT.0.) GO TO 101
   IF (K.LT.KTE1) GO TO 21
   I1     = ITE1(K)
   I2     = ITE2(K)
   CALL VELD (K,2,SV,SM,CP,X,Y)
   IF (K.GT.KTE1) GO TO 41
   ENCODE(40,32,R) TITLE
32 FORMAT(20A4)
   CALL SYMBOL(.5,0,0,14,R,0,00)
   ENCODE(44,34,R) FMACH,YA,AL
34 FORMAT(5HMACH ,F7.3,4X,5HYAW ,F7.3,4X,5HALPHA,F7.3)
   CALL SYMBOL(.5,-.25,0,14,R,0,44)
   ENCODE(44,36,R) VLD,CL,CC
36 FORMAT(5HL/D ,F7.2,4X,5HCL ,F7.4,4X,5HCD ,F7.4)
   CALL SYMBOL(.5,-.5,0,14,R,0,44)
41 SY     = 5.*(Z(K) -Z(KTE1))/(Z(KTE2) -Z(KTE1)) +2.75
   DO 42 I=1,I2
   X(I)   = SCALX*X(I) +SX
   Y(I)   = SCALX*Y(I) +SY
42 CP(I)  = SCALP*CP(I) +SY
   IF (K.EQ.2) GO TO 51

```

```

      N      = I2 -I1 +1
      CALL LINE(X(I1),Y(I1),N,1,C,1,C,1,C,1,C,1,C,1,C,1)
      GO TO 21
51  N      = I2 -IX +1
      CALL LINE(X(I1),Y(I1),N,1,C,1,C,1,C,1,C,1,C,1,C,1)
      N      = IX -I1 +1
      DO 52 I=I1,IX
52  X(I)    = X(I) +TX
      CALL LINE(X(I1),Y(I1),N,1,C,1,C,1,C,1,C,1,C,1,C,1)
      GO TO 21
61  PEWIND N1
      M      = M +1
      CALL PLOT(12,,0,, -3)
      IF (M.GT.2) GO TO 71
      SX     = -SC/LX*X((KTE))
      ENCODE(24,62,R)
62  FORMAT(24HUPPER SURFACE PRESSURE )
      CALL SYMBOL(0,,.5,.14,R,C,24)
      ENCODE(24,64,R)
64  FORMAT(24HLOWER SURFACE PRESSURE )
      CALL SYMBOL(3.5,.5,.14,R,C,24)
      GO TO 11
71  IU      = 1
      RETURN
101 IU      = 0
      CALL PLOT(12,,0,, -3)
      RETURN
      END

```

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Administration, nor any person acting on behalf of the Administration:

- A. Makes any warranty or representation, express or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
- B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Administration" includes any employee or contractor of the Administration, or employee of such contractor, to the extent that such employee or contractor of the Administration, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Administration, or his employment with such contractor.