NASA CR-132477

CALCULATION OF UNSTEADY TRANSONIC AERODYNAMICS FOR OSCILLATING WINGS WITH THICKNESS (COMPUTER PROGRAM)

By S. Y. Ruo

September 1974

Prepared under Contract No. NAS1-11156 by
LOCKHEED-GEORGIA COMPANY
Marietta, Georgia

for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

CONTENTS

Page Intentionally Left Blank

CALCULATION OF UNSTEADY TRANSONIC AERODYNAMICS FOR OSCILLATING WINGS WITH THICKNESS (COMPUTER PROGRAM)

By S. Y. Ruo

Lockheed-Georgia Company

SUMMARY

A computer program has been developed to account approximately for the effects of finite wing thickness in the transonic potential flow over an oscillating wing of finite span. The program is based on the original sonic-box program of Rodemich and Andrew, and accounts for the nonuniform flow caused by finite thickness by application of the local linearization concept. A brief description of each subroutine is given, and the method of input is shown in detail. A sample problem as well as a complete listing of the computer program are presented.

INTRODUCTION

This report is prepared as a supplement to NASA Contractor Report CR-2259 entitled "Calculation of Unsteady Transonic Aerodynamics for Oscillating Wings with Thickness." The purpose of this supplement is to aid those who are interested in using the computer program.

A major part of the computer program described herein is adapted from the original sonic-box computer program developed by Rodemich and Andrew.* In addition to its original capability, the program has been extended to include the thickness effect of a finite wing in unsteady sonic flow. This is accomplished in an approximate manner by employing the local linearization concept in conjunction with the known steady Mach number distribution or pressure distribution of the wing at its mean position. As the program is presently formulated, Mach number is assumed to depend only on the wing thickness; therefore, the local Mach number is the same at corresponding points on upper and lower wing surfaces. The flow is assumed to remain attached and shock-free. The program is limited to wings with unswept trailing edge and without control surfaces. The plane perpendicular to the wing planform and passing through the centerline chord is the plane of symmetry for both geometry and motion.

The wing with thickness is converted into a wing without thickness in a transformed space in which the flow properties over the wing are computed. These transformed flow properties are then converted into the corresponding properties in the physical space and the generalized aerodynamic force coefficients are computed.

---

* Rodemich, E. R.; and Andrew, L. V.: Unsteady Aerodynamics for Advanced Configurations, Part II - A Transonic Box Method for Planar Lifting Surfaces. FDL-TDR-64-152, Part II, May 1965, Air Force Flight Dynamics Lab., Wright-Patterson Air Force Base, Ohio.

SYMBOLS

a_{nm}	coefficients in doublet potential polynomial, see equation (2)
AXY(I,J)	integral defined in equation (5)
AY(J)	integral defined in equation (5)
B_{ij}	area of ij-th box on the wing (i-th spanwise column and j-th chordwise row)
b	reference length - wing centerline chord (dimension = L)
$d_{n'm'}$	coefficients in wing deflection polynomial, see equation (3)
$F(x,y)$	leading edge correction term, see equation (4)
$f(x,y)$	mode shape polynomial, see equation (3)
$G_1(x,y)$, $G_2(y)$	leading edge correction terms, see equation (4)
i	$\sqrt{-1}$
k	reduced frequency, $\omega b/U_\infty$
L	unit of length
L_{ij}	generalized aerodynamic force coefficient
S	dimensionless wing planform area of full wing (reference area = b^2)
T	unit of time
U_∞	freestream velocity (dimension = L/T)
$w(x,y)$	dimensionless downwash (reference velocity = U_∞)
x,y	dimensionless cartesian coordinates (reference length = b)
$x_\ell(y)$	x-coordinate of wing leading edge
y_{max}	dimensionless maximum semi-span (reference length = b)
$\varphi_o(x,y)$	dimensionless velocity potential of doublet
$\psi(x,y)$	downwash at point (x,y) due to doublet of unit strength at origin, see equation (8)
ω	angular velocity (dimension = radian/T)

VARIABLES IN COMMON STATEMENT

The variable names listed in the general COMMON statement in the MAIN program are described briefly in this section. The subroutine name in the parentheses following the description of a variable indicates where its value is generated or defined. The number appearing in the parentheses indicates the value of the variable for the corresponding condition described. In the following, N=1 and N=2 indicate, respectively, the real and the imaginary parts of a variable; i and n are related to the chordwise coordinate while j and m are related to the spanwise coordinate; * represents multiplication.

A(N,j,i)	influence coefficient, the integral in equation (6); the upwash at the center of a box caused by a unit doublet distribution over another box separated from the former by j number of boxes in spanwise direction and i number of boxes in chordwise direction, (POT2)
W(N,j,i)	downwash at ij-th box, equation (7), (WVAL)
S(N,j,i)	velocity potential at ij-th box, (BOXPO)
DA(k)	input data, (DATRD)
PS(N,n,m*M)	inner summation of equation (1) including the integrals for M-th mode, (BOXP)
DF(NEW,n,m*M)	coefficients in wing deflection polynomial at M-th mode, (DRED)
ML(NEW,i)	number of boxes in i-th spanwise column, (SHAPE)
AXY(n,m)	integral defined in equation (5), (SECT)
AY(m)	integral defined in equation (5), (SECT)
XEDGI(k)	x-coordinate of input leading edges of physical wing, (SHAPE)
YEDGI(k)	y-coordinate of input leading edges of physical wing, (SHAPE)
IEDG(NEW)	flag to identify the last section of the leading edge is parallel (1) or not parallel (0) to the freestream, (SHAPE)
NS(NEW)	number of leading edge segments, (SHAPE)
EM(j,i)	Mach number at the center of the ij-th box calculated from the fitted surface, (SHAPE)
XEDG(k)	x-coordinate of leading edges used in computation, (SHAPE)
YEDG(k)	y-coordinate of leading edges used in computation, (SHAPE)

HM(n,m)	coefficients in Mach number polynomial, (MRED)
AREA	area of the physical, full wing planform, (SHAPE)
D	length of box side, (MAIN)
DI	number (real) of boxes along centerline chord, (MAIN)
DH	one-half of the length of box side, (MAIN)
CK	reduced frequency
L	number (integer) of boxes along centerline chord, (MAIN)
M	deflection mode number, (MAIN)
NEW	index for physical (1) or transformed (2) wings, (MAIN)
IR	computer read-unit number, (MAIN)
IW	computer write-unit number, (MAIN)

FUNCTION OF SUBROUTINES

A brief description of the function of each subroutine is given in this section. Some equations are noted here for the purpose of identifying the steps in the sequence of obtaining the final results, i.e., the generalized aerodynamic force coefficients. It is suggested to refer to the report of Rodemich and Andrew* for the derivations of these equations. The freestream flow is in the positive x direction and the apex of wing is at the origin. Spanwise direction is designated by y. Since the plane passing through the centerline chord and perpendicular to the wing planform is the plane of symmetry of wing geometry and motion, the input of wing geometry, deflection, and steady-flow Mach number distribution needs to be made only for the portion of the wing where $x \geq 0$ and $y \geq 0$. It is assumed that the camber, twist, and mean angle of attack of the wing are all small; and the difference in local Mach number at the corresponding points of the upper and lower wing surfaces can be neglected. As the computer program is presently formulated, the motion is limited to the symmetric mode. This limitation is imposed by the expression used in the present program to evaluate the influence coefficient of upwash due to a unit doublet distribution.

In the following, the variable name used in the computer program to represent a certain quantity mentioned in the description of a subroutine is indicated in the parentheses following such quantity.

MAIN - A controlling routine; also performs the final calculation of the generalized aerodynamic force coefficient which is written for full wing as

$$L_{ij} = \frac{8}{S} \sum_{n',m'} d_{n'm'} \sum_{n,m} a_{nm} \Bigg[\frac{1}{2} \int_{x=1} y^{2(m+m')} F(1,y)dy$$
$$+ik\frac{1}{2} \iint_S x^{(n+n')} y^{2(m+m')} F(x,y)dxdy$$
$$-n'\frac{1}{2} \iint_S x^{(n+n'-1)} y^{2(m+m')} F(x,y)dxdy \Bigg] , \qquad (1)$$

where S = area of the wing planform (full wing)

k = reduced frequency,

$d_{n'm'}$ and a_{nm} are respectively the coefficients of the wing deflection and velocity potential polynomials, i.e.

*see footnote on page 2

$$\varphi_o(x,y) = \sum_m \sum_n a_{nm}\, x^n\, y^{2m}\, F(x,y), \quad n,m = 0 \rightarrow 4 \tag{2}$$

$$f(x,y) = \sum_{m'} \sum_{n'} d_{n'm'}\, x^{n'}\, y^{2m'}, \quad n',m' = 0 \rightarrow 4 \tag{3}$$

The function $F(x,y)$ in equation (2) is defined as

$$F(x,y) = G_1(x,y) \cdot G_2(y) \tag{4}$$

where the expressions for G_1 and G_2, depending solely on the shape of the leading edge, are

$$G_1(x,y) = \begin{cases} \sqrt{x^2 - x_\ell^2(y)}, & \text{if } y = 0 \text{ at } x = 0 \\ \text{or} & \\ \sqrt{x - x_\ell(y)}, & \text{if } y \neq 0 \text{ at } x = 0 \end{cases}$$

$$G_2(y) = \begin{cases} \sqrt{1 - \left(\dfrac{y}{y_{max}}\right)^2}, & \text{if } \dfrac{dy}{dx} = 0 \text{ at } x = 1 \\ \text{or} & \\ 1 & \text{, if } \dfrac{dy}{dx} \neq 0 \text{ at } x = 1 \end{cases}$$

These relationships are depicted in figure 1, and x (y) is the leading edge of the wing (see fig. 1).

DATRD - data input routine; for a new wing, the data array (DA(k)) is cleared every time but for the same wing, the old value is used if a new value is not entered (blank datum will not affect the old value).

SHAPE - wing geometry routine; approximates the wing planform with a grid of square boxes and also calculates the number of boxes (ML(NEW,i)) along each spanwise column (i-th) on both physical and transformed wings; computes the area of the full wing planform (AREA) and the surface-fitted (HM(n,m)) Mach number (EM(j,i)) at the center of each box (ij-th) on the physical wing; increases the number of leading edge segments (XEDG(i), YEDG(i)) by subdividing the input leading edge segments (XEDGI(i), YEDGI(i)) for the physical wing so that the transformed wing can be better approximated;

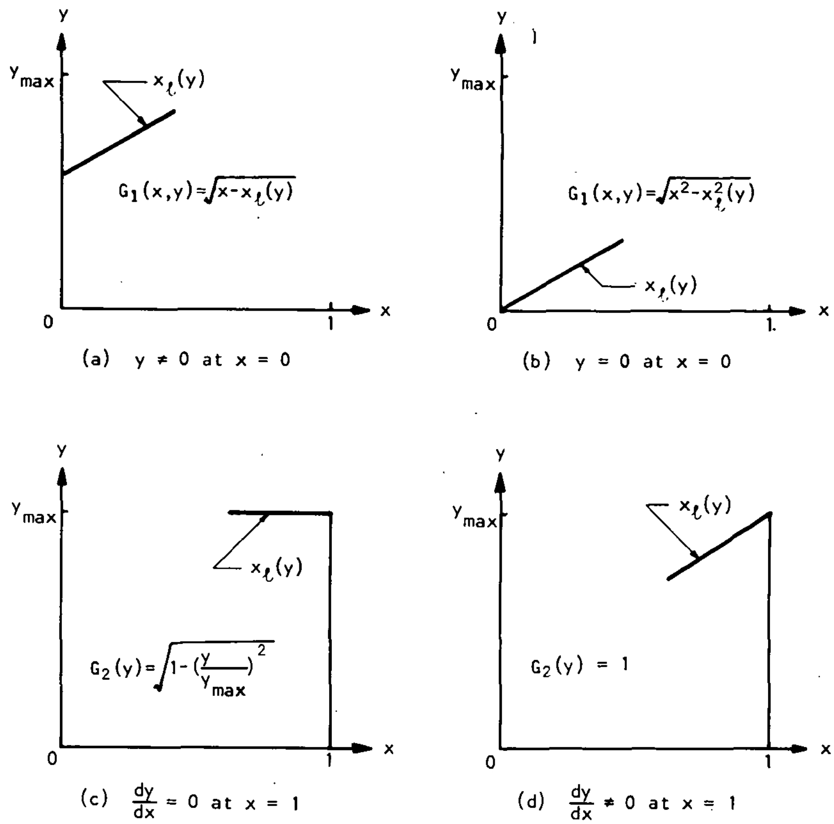

Figure 1. - Shapes of leading edge.

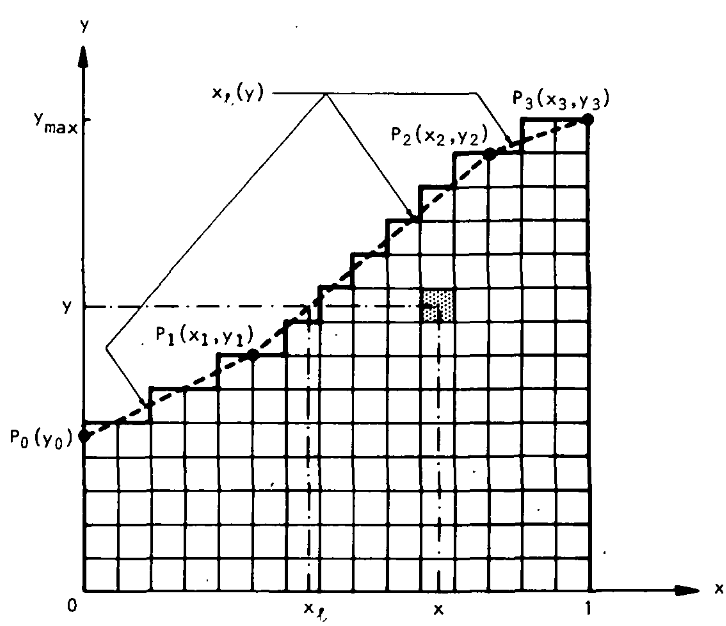

Figure 2. - Half wing geometry.

performs the coordinate transformation of the wing with the aid of the known Mach number distribution (EM(j,i)) on the physical wing and also checks the possible creation of an artificial wake in the transformed wing resulting from an uneven distortion of the wing tip.

MACH - Mach number subroutine; computes the mean local Mach number at a given point (x,y) on the physical wing from the known polynomial (HM(n,m)) which was fitted to the input data for Mach number distribution; the Mach number at corresponding points on upper and lower surfaces is assumed to be the same.

FORCI - controlling routine for the calculation of integrals in the expression for generalized aerodynamic force coefficient (Eq.(1)); obtains the values of

$$\begin{aligned} AXY(I,J) &= \frac{1}{2} \iint_S x^{(I-1)} y^{2(J-1)} F(x,y)\,dxdy \\ AY(J) &= \frac{1}{2} \int_{x=1} y^{2(J-1)}\, F(1,y)\,dy, \end{aligned} \tag{5}$$

where the integrals are evaluated in subroutine SECT. Since these integrals depend only on the wing geometry, their values are calculated once and used for all frequencies and modes both with and/or without thickness.

SECT - carries out the calculation of the integrals in equation (5) using the six-point Gaussian quadrature

$$\int_0^1 \int_0^1 g(x,y)\ dxdy = \sum_{i=1}^{n} \sum_{j=1}^{n} h_i h_j g(x_i, y_j)$$

and

$$\int_0^1 g(x)dx = \sum_{i=1}^{n} h_i g(x_i),$$

in which n = 6 is assumed.

POT2 - evaluates the integral in the expression relating the downwash and the velocity potential

$$\sum_{i',j'} \varphi_{o_{i'j'}} \iint_{B_{i'j'}} \psi(x_i-\xi, y_j-\eta)\,d\xi d\eta = w(x_i, y_j) \tag{6}$$

where $\varphi_{o_{i'j'}}$ = magnitude of velocity potential at the center of box $B_{i'j'}$

$w(x_i,y_j)$ = downwash at the center of box B_{ij}

$$= \left(\frac{\partial f}{\partial x} + ikf\right) \quad \text{at } (x_i,y_j) \tag{7}$$

$\psi(x_i-\xi,y_j-\eta)$ = downwash at (x_i,y_j) due to doublet of unit strength at (ξ,η).

$$= \frac{ik}{2\pi}\,\frac{1}{(x_i-\xi)^2}\,\exp\left\{-\frac{1}{2}ik\left[(x_i-\xi)+\frac{(y_j-\eta)^2}{(x_i-\xi)}\right]\right\}; \tag{8}$$

the integral in equation (6) is integrated around the edge of each box $B_{i'j'}$ centered at $(x_{i'},y_{j'})$ once for every frequency and the result (A(N,j,i)) is used in the calculation of the velocity potential (S(N,j,i)) in both physical and transformed spaces for all modes under consideration at the same frequency.

CIN - calculates sine and cosine integrals.

DRED - fits the wing deflection for each mode (M) of motion in both physical and transformed spaces with a polynomial of the form shown in equation (3) expressed in terms of the dimensionless coordinates and stores the coefficients (DF(NEW,n',m'*M)); if the data of wing deflection are given at a number of points on either the physical or the transformed wing, these data are fitted into the desired polynomial by the least-square surface fitting method; for the physical wing, the wing deflection in each mode can be given as input either in the form of the desired polynomial coefficients or deflection at a number of selected points; for the transformed wing, the wing deflection in a particular mode is obtained from a number of selected points on the physical wing of known deflection (DF(1,n',m'*M)), the deflection at a point on the transformed wing is unaltered from that of the corresponding point on the physical wing but its spanwise coordinate in the transformed space is modified by a factor of the Mach number from that of the physical wing.

WVAL - calculates the complex (real-N=1, imaginary-N=2) downwash (W(N,j,i)) at the center of each box (ij-th) for a particular mode (M) and reduced frequency (CK) using the coefficients of wing deflection polynomial (DF(NEW,n',m'*M)) for the physical or the transformed wing.

BOXPO - solves a set of simultaneous equations, equation (6), relating the complex (real-N=1, imaginary-N=2) velocity potential (S(N,j,i)) and the known complex downwash (W(N,j,i)) of a particular mode (M) at the center of each box (ij-th) to obtain the velocity potential of the physical or the transformed wing.

BOXP - fits the complex (real-N=1, imaginary-N=2) velocity potential distribution (S(N,j,i)) calculated in BOXPO for physical or transformed wing with a pre-determined polynomial surface (C(k,N)), as shown in equation (2), for both real and imaginary parts; for the case with thickness effect, the complex velocity potential on the physical wing is obtained from the fitted complex velocity potential (C(k,N)) for the transformed wing by selecting a number of points on the physical wing, computing the complex velocity potential of the corresponding points on the transformed wing with the known polynomial and transforming the potential to that of those points on the physical wing, then fitting this indirectly obtained potential with the pre-determined polynomial surface (C(k,N)); also calculates the values of the inner summation (PS(N,n,m*M)) of equation (1); prints out the downwash (W(N,j,i)) calculated from the deflection polynomial (DF(NEW,n,m*M)), the computed potential (S(N,j,i)), the fitted potential (PSI(N,j)) or the lifting pressure (PR(N,j)) calculated from the fitted potential, etc., at the box centers, if the controls (DA(1003) to DA(1010)) are flagged.

MRED - fits the input steady-state Mach number distribution or pressure coefficient distribution on the physical wing with a polynomial surface of the form shown in equation (3) expressed in terms of the dimensionless coordinates for Mach number; the input for Mach number can be either in the form of the desired polynomial coefficients or the Mach number or the pressure coefficient at a number of selected points on the physical wing; for the latter case, a least-square surface-fitting method is used to obtain the desired polynomial (HM(n',m')).

LSQUA - method of the least-square surface-fitting routine.

MSIMER - finds solutions of the simultaneous real number equations.

MSIMEC - finds solutions of the simultaneous complex number equations.

FLOW CHART

The flow chart of the computer program is presented in figure 3. The function of those controlling variables appeared in the flow chart is as follows.

VARIABLE	VALUE	FUNCTION
NEW	1	Indicates case without thickness effect
	2	Indicates case with thickness effect
DA(26)	0	Indicates first frequency for a wing
	1	Indicates additional frequency for the same wing
DA(44)	0	Calculates cases with and without thickness effects
	1	Calculates case without thickness effects only
	2	Calculates case with thickness effects only
M	$\leq$ DA(28)	Counter of the modes of deflection
DA(28)	$\leq$ 10	Total number of modes to be considered
IND		Index of the storage location in data array DA

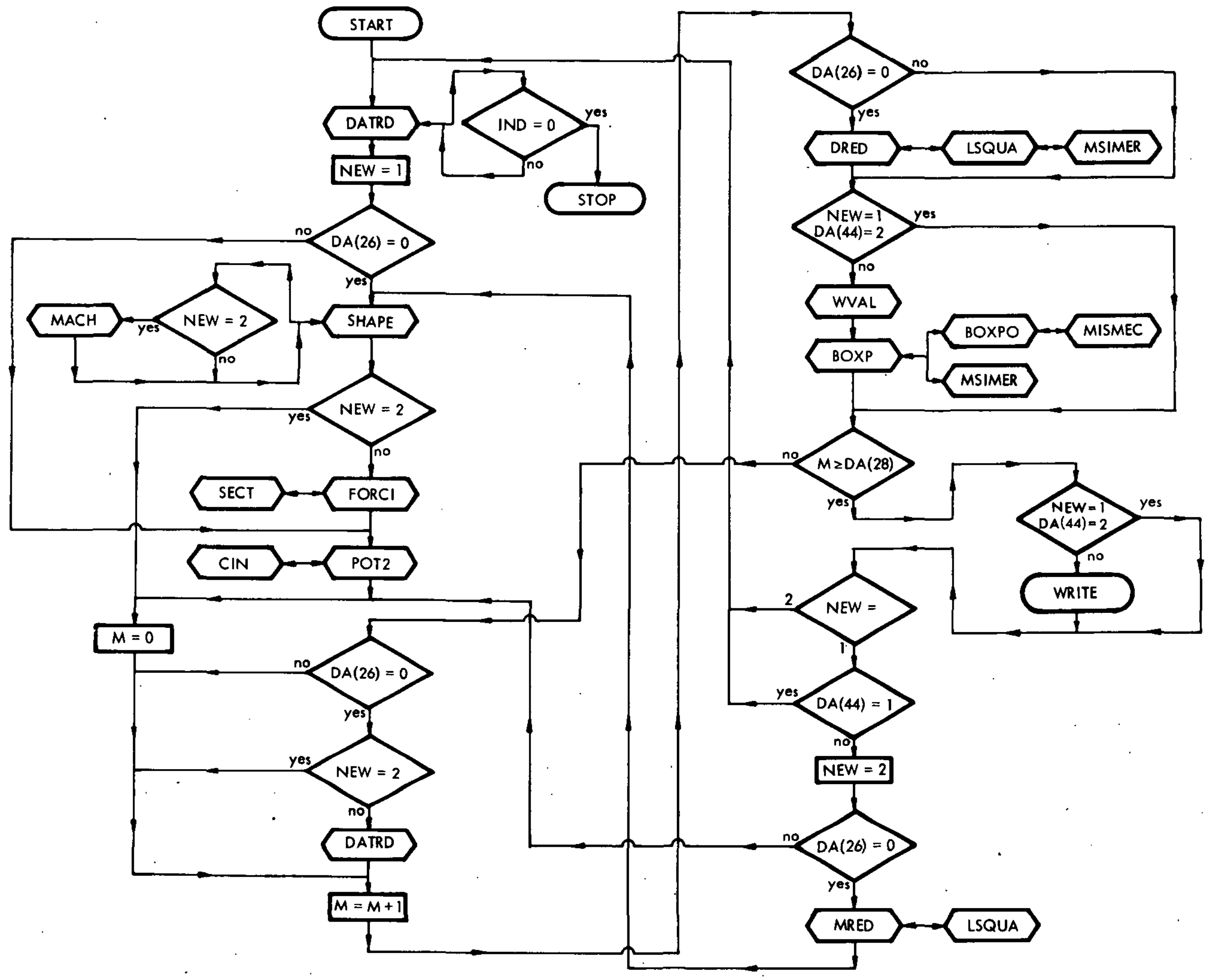

Figure 3. - Flow chart.

At the start of execution, as it calls DATRD, the data array, DA is cleared by a "+" sign entered in the first column of the first data card. The wing geometry, reference velocity, first frequency, data for the first mode of wing motion and the optional set of controls (DA(1001)-DA(1014)) are entered with a "-" sign at the first column of the last card of this data set.

The physical wing planform is approximated by a grid of square boxes in SHAPE and those integrals depending only on the wing geometry are evaluated in FORCI and SECT. Those integrals involving the frequency and wing geometry are calculated in POT2. The wing deflection of this particular mode is rearranged into a pre-determined polynomial surface in DRED. Subroutine LSQUA is not called when the wing deflection is entered as the desired polynomial coefficients. With the known downwash, computed in WVAL with the aid of the polynomial for deflection, the velocity potential at the center of each box is calculated in BOXPO. This potential is fitted into a polynomial in BOXP. The unsteady lifting pressure may be calculated and printed out from BOXP if desired. This ends the calculation of one mode of wing motion at one frequency. If there are more modes to be treated, a new set of wing deflections for each mode is read in separately each time and the corresponding velocity potential is computed until all modes are completed. Then the generalized aerodynamic force coefficients are calculated and printed out. This completes the computation for the case without thickness effect.

If the case with thickness effect of the same wing at the same frequency is desired, the steady local Mach number data are read in, either in a polynomial form or the actual values (steady pressure coefficient may be used) at a set of selected points on the wing, and they are fitted with a pre-determined polynomial surface in MRED. In subroutine SHAPE the Mach number at the center of each box over the physical wing and at the end points of each leading edge segment is computed using MACH with the aid of the polynomial for Mach number and the wing planform is then transformed; also the number of the straight line segments representing the transformed wing leading edge is increased so that the distorted wing can be better approximated. Since the frequency and the original wing have not changed, subroutine FORCI and POT2 are skipped. There is no need to read in the wing deflection again. Each mode is treated as before except the wing is now considered to be in the transformed space. The wing deflection is redistributed on the transformed wing. This is accomplished by computing the wing deflection at a set of selected points on the physical wing, transforming the coordinates of those selected points and then fitting the wing deflection with a polynomial in terms of the transformed coordinates in DRED. The velocity potential computed in BOXPO for the transformed wing is converted and surface-fitted into that of the physical wing in BOXP as each mode is treated. The calculation of generalized aerodynamic force coefficients is performed in the physical space. This completes the calculation for a wing oscillating at different modes at one frequency without and with thickness effect.

The whole process is repeated for each additional frequency except the subroutines SHAPE, FORCI, DRED and MRED are skipped. A (DA(44)) flag may be used to control the calculation for the case either with or without thickness effect or for both.

There is no limit to the numbers of different frequencies or different wings that can be handled in a single execution. A blank card placed behind the last card of the data sets will automatically terminate the execution.

INPUT GUIDE

Data input is made through the subroutine DATRD using the one dimensional array DA with a size of 1015. The allowable maximum number for some of the input data as indicated below may be changed if the dimension of the corresponding storage array and computational operations are also changed accordingly. The layout of the array DA(k) as it is presently used is as follows:

1-12: Title

13-22: Mode title

23: Frequency, (cycle/sec)

24: Centerline chord length, (ft or meter)

25: Speed of sound of the freestream, (ft/sec or meter/sec)

26: (0) - indicates the frequency is the first one for a new wing

(1) - indicates the frequency is the additional one for the same wing

27: Number of boxes along centerline chord (maximum 50)

28: Number of deflection modes (maximum 10)

29: Number of sections of leading edge to be given (maximum 6)

30-42: Coordinates of points on the leading edge, (ft or meter) (in the sequence of y_0, x_1, y_1, x_2, y_2, ..., x_6, y_6)

43: Number of boxes allowed for upstream influence (if this location is left blank or assigned a zero, it will assume DA(43)=DA(27) and in no case DA(43) DA(27) is allowed).

44: (0) - indicates to calculate cases with and without thickness effect
(1) - indicates to calculate case without thickness effect only
(2) - indicates to calculate case with thickness effect only

45: Indicator to suppress calculation of potential for a mode
(0) - no suppression
(1) - suppression

46-70: Coefficients of the deflection polynomial (in the sequence of d_{00}, d_{10}, ..., d_{40}, d_{01}, d_{11}, ..., d_{41}, ..., d_{04}, d_{14}, ..., d_{44})

71-95: Coefficients of the Mach number distribution polynomial (in the sequence of d_{00}, d_{10}, ..., d_{40}, d_{01}, d_{11}, ..., d_{41}, ..., d_{04}, d_{14}, ..., d_{44})

96: Indicator of the type of wing thickness effect input
(1) - pressure coefficient
(2) - Mach number

97: Number of points at which pressure coefficient or Mach number to be given

98: Number of points on which deflections to be given

99: 1 plus the maximum power of x in surface fit of deflection (maximum 5)

100: 1 plus the maximum power of y^2 in surface fit of deflection (maximum 5)

101-700: Deflection data for a maximum of 150 points (in the sequence of x, y, deflection and weighting factor)

701-1000: Pressure coefficient or Mach number data for a maximum of 100 points (in the sequence of x, y and pressure coefficient or Mach number)

The remaining part of DA array is used for the control of intermediate results print out. When the latter is desired, a non-zero number should be entered at a location in the DA array corresponding to the information wanted.

1001: Coefficients of surface-fitted deflection polynomial (no thickness effect)

1002: Coefficients of surface-fitted deflection polynomial (with thickness effect)

1003: Value of upwash calculated (no thickness effect)

1004: Value of upwash calculated (with thickness effect)

1005: Value of potential calculated (no thickness effect)

1006: Value of potential calculated (with thickness effect)

1007: Coefficients of surface-fitted potential polynomial (no thickness effect)

1008: Coefficients of surface-fitted potential polynomial (with thickness effect)

1009: Surface-fitted values of potential and lifting pressure (no thickness effect)

1010: Surface-fitted values of potential and lifting pressure (with thickness effect)

1011: Coefficients of surface-fitted Mach number distribution polynomial

1012: Surface-fitted value of Mach number distribution

1013: Coordinates of sections of the leading edge of the transformed wing

1014: Value of deflection on transformed wing

1015: Not used

The format of the input data card is (A1, A5, I6, 11A6, A2). The first field is for the control of clearing the data array, DA, for a new wing (+) and the control to indicate the end of the set of data (-). The second field is the indicator for the type of data, either numeric (blank) or alphameric (ALPHA). The third field is the designator for the relative location in the data array of the first number to follow in the fourth field. If this field is left blank, or a zero is entered, the execution will be terminated. The fourth field is for five consecutive input data each occupying 12 columns plus 8 blank columns at the end. All the fixed point numbers are right-adjusted and the decimal point for the floating point number must be included. If an input datum is left blank, no change at the storage location for that particular datum in the data array will occur unless the set of the input data is for a new wing.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

SAMPLE CASE

A typical data cards set-up and output for an aspect ratio 1.5 delta wing having an elliptic lateral cross-section with 10% thickness ratio performing pitch about its apex and plunge are presented.

Input

The input format is (A1, A5, I6, 11A6, A2).

```
         1         2         3         4         5         6         7
123456789012345678901234567890123456789012345678901234567890123456789012 IDENTIFICATION

+ALPHA     1ASPECT RATIO 1.5 DELTA WING (TAU=0.10)                             1
          23 1.591549407        10.0      1000.0                               3
 ALPHA    13PLUNGE                                                             2
          27          40           2           1                               4
          30         0.0        10.0        3.75                               5
-         46         1.0                                                       6
 ALPHA    13PITCH ABOUT ROOT LEADING EDGE X=0.0                                7
          46         0.0         0.1                                           8
          96           1          26                                           9
         701         0.1                0.140232                              10
         704         0.3                0.135125                              11
         707         0.5                0.132175                              12
         710         0.7                0.129943                              13
         713         0.9                0.128084                              14
         716         1.3                0.124980                              15
         719         1.7                0.122340                              16
         722         2.1                0.119963                              17
         725         2.5                0.117746                              18
         728         2.9                0.115627                              19
         731         3.3                0.113564                              20
         734         3.9                0.110510                              21
         737         4.5                0.107428                              22
         740         5.1                0.104243                              23
         743         5.7                0.100872                              24
         746         6.3                0.097215                              25
         749         6.9                0.093129                              26
         752         7.5                0.088395                              27
         755         7.9                0.084697                              28
         758         8.3                0.080342                              29
         761         8.7                0.074970                              30
         764         9.1                0.067825                              31
         767         9.3                0.063062                              32
         770         9.5                0.056823                              33
         773         9.7                0.047661                              34
-        776         9.9                0.029651                              35
-         23 3.183098814                                       1              36
-         23 6.366197628                                       1              37
                                                                              38
```

Card 1: title of the case under consideration.

Card 2: title of the first mode of deflection.

Card 3: first frequency (cycle/sec), centerline chord length (ft), reference velocity (ft/sec).

Card 4: number of boxes along the centerline chord, number of deflection modes, number of total leading edge segments of the wing.

Card 5: spanwise coordinate (ft) of the first section of the leading edge, chordwise and spanwise coordinates (ft) of the next section (the sequence is y_0, x_1, y_1 -- e.g., see figure 2).

Card 6: first mode of deflection f = 1.0,
the " - " sign indicates the end of the group of data cards to be read at this stage.

Card 7: title of the second mode.

Card 8: second mode of deflection f = 0.1x.

Card 9: identification of the type of input regarding the wing thickness effect (pressure coefficient for this case), number of points on the wing this information to be given.

Cards 10 to 35:
chordwise and spanwise coordinates (ft) of a point on the wing, and the pressure coefficient* at this point.
the " - " sign on the last card indicates the end of the group of data cards to be read at this stage.

Cards 36 and 37:
additional frequencies for the same wing, one card is read in at one time.

Card 38: blank card to make an exit from the computer.

---

*Alksne, A. Y.; and Spreiter, J. R.: Theoretical Pressure Distributions on Wings of Finite Span at Zero Incidence for Mach Number Near 1. NASA TR R-88, 1960.

Output

```
ASPECT RATIO 1.5 DELTA WING (TAU=0.10)
40 BOXES ALONG ROOT CHORD                    ROOT CHORD LENGTH    =   10.00 FT
REDUCED FREQUENCY =  .100                    FREE STREAM VELOCITY = 1000.00 FT/SEC
FREQUENCY =   1.592+00 CYCLE/SEC
          MODE NO. 1    PLUNGE
          MODE NO. 2    PITCH ABOUT ROOT LEADING EDGE X=0.0
```

GENERALIZED FORCES (NO THICKNESS EFFECT)

MODES PRES.	DEFL.	REAL PART	IMAG PART	ABS. VALUE	PHASE ANGLE
1	1	-6.47765-05	-2.40475-01	2.40475-01	-90.0154
1	2	-1.95306-04	-1.58666-01	1.58666-01	-90.0706
2	1	-2.40959+00	-2.39303-01	2.42144+00	-174.3284
2	2	-1.59065+00	-1.77008-01	1.60047+00	-173.6502

```
          MODE NO. 1
          MODE NO. 2
```

GENERALIZED FORCES (WITH THICKNESS EFFECT)

MODES PRES.	DEFL.	REAL PART	IMAG PART	ABS. VALUE	PHASE ANGLE
1	1	1.55911-03	-2.50292-01	2.50297-01	-89.6431
1	2	1.22643-03	-1.63389-01	1.63394-01	-89.5699
2	1	-2.49934+00	-2.66717-01	2.51353+00	-173.9087
2	2	-1.63035+00	-1.98794-01	1.64212+00	-173.0468

ASPECT RATIO 1.5 DELTA WING (TAU=0.10)

40 BOXES ALONG ROOT CHORD — ROOT CHORD LENGTH = 10.00 FT

REDUCED FREQUENCY = .200 — FREE STREAM VELOCITY = 1000.00 FT/SEC

FREQUENCY = 3.183+00 CYCLE/SEC

MODE NO. 1

MODE NO. 2

GENERALIZED FORCES (NO THICKNESS EFFECT)

MODES PRES.	DEFL.	REAL PART	IMAG PART	ABS. VALUE	PHASE ANGLE
1	1	4.33190-03	-4.77153-01	4.77173-01	-89.4798
1	2	3.41201-03	-3.14166-01	3.14184-01	-89.3778
2	1	-2.39958+00	-4.95443-01	2.45023+00	-168.3340
2	2	-1.58175+00	-3.69772-01	1.62439+00	-166.8420

MODE NO. 1

MODE NO. 2

GENERALIZED FORCES (WITH THICKNESS EFFECT)

MODES PRES.	DEFL.	REAL PART	IMAG PART	ABS. VALUE	PHASE ANGLE
1	1	6.30539-03	-4.98918-01	4.98958-01	-89.2759
1	2	4.77937-03	-3.25637-01	3.25672-01	-89.1592
2	1	-2.50045+00	-5.29752-01	2.55595+00	-168.0380
2	2	-1.63191+00	-3.93733-01	1.67873+00	-166.4354

ASPECT RATIO 1.5 DELTA WING (TAU=0.10)

40 BOXES ALONG ROOT CHORD ROOT CHORD LENGTH = 10.00 FT

REDUCED FREQUENCY = .400 FREE STREAM VELOCITY = 1000.00 FT/SEC

FREQUENCY = 6.366+00 CYCLE/SEC

MODE NO. 1

MODE NO. 2

GENERALIZED FORCES (NO THICKNESS EFFECT)

MODES PRES.	DEFL.	REAL PART	IMAG PART	ABS. VALUE	PHASE ANGLE
1	1	3.08793-02	-9.48383-01	9.48885-01	-88.1351
1	2	2.46913-02	-6.24848-01	6.25336-01	-87.7371
2	1	-2.40445+00	-1.00687+00	2.60675+00	-157.2784
2	2	-1.58734+00	-7.53332-01	1.75703+00	-154.6115

MODE NO. 1

MODE NO. 2

GENERALIZED FORCES (WITH THICKNESS EFFECT)

MODES PRES.	DEFL.	REAL PART	IMAG PART	ABS. VALUE	PHASE ANGLE
1	1	3.30904-02	-9.87578-01	9.88132-01	-88.0809
1	2	2.47565-02	-6.43576-01	6.44052-01	-87.7971
2	1	-2.49756+00	-1.05658+00	2.71186+00	-157.0694
2	2	-1.63073+00	-7.83174-01	1.80904+00	-154.3468

```
C      MAIN PROGRAM
       COMMON     A(2,100,50),W(2,50,50),DA(1015),PS(2,5,50),DF(2,5,50)
      1           ,ML(2,50),AXY(9,9),AY(9),XEDGI(8),YEDGI(8)
      2           ,IEDG(2),NS(2),EM(50,50),XEDG(32),YEDG(32),HM(5,5)
      3           ,AREA,D,DI,DH,CK,L,M,NEW,IR,IW
       DIMENSION G(10)
       DATA Z/1H /
       IR=5
       IW=6
       WRITE(IW,55)
C      READ DATA FOR THE ACTUAL WING
   100 CALL DATRD(DA)
       NEW=1
       TEST1=DA(1001)+DA(1003)+DA(1005)+DA(1007)+DA(1009)
       CK=DA(23)*DA(24)/DA(25)*6.28318531
       DI=DA(27)
       L=DI
       D=1.0/DI
       DH=0.5*D
       WRITE
      0  (IW,60)(DA(I),I=1,12)
   110 IF (L) 600,600,120
   120 IF (50-L) 600,130,130
   130 WRITE
      0    (IW,65) L,DA(24),CK,DA(25),DA(23)
       IF (DA(26)) 160,150,160
   150 CALL SHAPE
       IF(NEW.EQ.2) GO TO 180
       CALL FORCI
   160 LIM=ML(NEW,L)
       IF (LIM-50) 170,170,650
   170 LIM2=2*LIM
       NFLNS = DA(43)
       IF(NFLNS.EQ.0) NFLNS = L
       LPOT = MIN0(L,NFLNS)
       CALL POT2(100,LIM2,LPOT,CK,D)
   180 CONTINUE
       M=0
       K=DA(28)
       GO TO 230
C      PRELIMINARY CALCULATIONS ARE FINISHED.
C      THE NEXT SECTION IS GONE THROUGH FOR EACH MODE.
   200 IF (DA(26)) 230,210,230
   210 IF(NEW.EQ.2) GO TO 230
       CALL DATRD(DA)
```

```
230 K=K-1
    M=M+1
    IF(TEST1.LT.1.0) GO TO 250
    WRITE
   0  (IW,10)M
 10 FORMAT (1H115X,8HMODE NO.I3)
    GO TO 270
250 WRITE
   0   (IW,15 ) M
 15 FORMAT(1H0,15X,8HMODE NO.+I3)
270 CONTINUE
    WRITE
   0  (IW,16)(DA(I),I= 13,22)
 16 FORMAT(1H+,30X,12A6)
    IF (DA(26)) 310,290,310
290 CALL DRED
300 G(M)=DA(45)
310 IF (G(M)) 380,320,380
320 IF(IFIX(DA(44)).EQ.2.AND.NEW.EQ.1) GO TO 380
    CALL WVAL
    IF (ML(NEW,1)) 330,370,330
330 LIM2=2*ML(NEW,1)
    DO 350 J=1,LIM2
350 W(J,1,1)=W(J,1,1)*2.0/3.14159265
C     LEADING EDGE CORRECTION
370 CONTINUE
    CALL BOXP
380 IF (K) 400,400,390
390 IF (M-10) 200,400,400
C     FINAL SECTION OF PROGRAM - COMPUTATION OF GENERALIZED FORCES
400 CONTINUE
    IF(IFIX(DA(44)).EQ.2.AND.NEW.EQ.1) GO TO 480
    IF(TEST1.LT.1.0) GO TO 410
    WRITE(IW,55)
    WRITE
   0   (IW,60) (DA(I),I=1,12)
    WRITE
   0   (IW,65) L,DA(24),CK,DA(25),DA(23)
410 WRITE (IW,20)
    IF(NEW.EQ.1) WRITE(IW,22)
    IF(NEW.EQ.2) WRITE(IW,24)
    WRITE(IW,25)
 20 FORMAT(/1H010X,18HGENERALIZED FORCES)
 22 FORMAT(1H+,28X,22H (NO THICKNESS EFFECT))
 24 FORMAT(1H+,28X,24H (WITH THICKNESS EFFECT))
 25 FORMAT(1H0,5X,5HMODES/4X,11HPRES. DEFL.,8X,9HREAL PART,10X,9HIMAG
   $PART,10X,10HABS. VALUE,6X+11HPHASE ANGLE)
    AC=8.0/AREA
    DO 470 M1=1,M
    IF (G(M1)) 470,430,470
430 DO 450 M2=1,M
    S1=0.0
    S2=0.0
    N1=5*(M1-1)
```

```
      N2=5*(M2-1)
      DO 440 J=1,5
      J1=J+N1
      J2=J+N2
      DO 440 I=1,5
      S1=S1+PS(1,I,J1)*DF(1,I,J2)
  440 S2=S2+PS(2,I,J1)*DF(1,I,J2)
      S1=AC*S1
      S2=AC*S2
      S3= SQRT(S1**2+S2**2)
      S4=57.29578*ATAN2(S2,S1)
  450 WRITE
     O  (IW,30)M1,M2,S1,S2,S3,S4
   30 FORMAT (1H02I6,1P3E19.5,0P1F16.4)
  470 CONTINUE
  480 WRITE  (IW,55)
      IF(DA(26).GT.0.) GO TO 520
      DO 500  I=13,22
  500 DA(I)=Z
C     CHECK IF TRANSFORMED WING HAS BEEN CALCULATED
C     IF NOT, GO TO  SHAPE  TO TRANSFORM THE WING
  520 IF(NEW.EQ.2.OR.IFIX(DA(44)).EQ.1) GO TO 100
      NEW=2
      IF(DA(26)) 180,530,180
  530 CALL MRED
      GO TO 150
C     ERROR EXITS
  600 IPR=27
  610 WRITE
     O  (IW,35)IPR
   35 FORMAT(1H010X,8HBAD DATAI4)
      GO TO 700
  650 WRITE   (IW,40)
   40 FORMAT(1H010X,42HLATERAL LIMIT ON NUMBER OF BOXES EXCEEDED.)
  700 STOP
   55 FORMAT(1H1)
   60 FORMAT (1H010X,12A6)
   65 FORMAT(1H0,10X,I2,23H BOXES ALONG ROOT CHORD,
     1             15X,22HROOT CHORD LENGTH     =,F8.2,3H FT/
     2         1H0,10X,19HREDUCED FREQUENCY =,F6.3,
     3             15X,22HFREE STREAM VELOCITY =,F8.2,7H FT/SEC/
     4         1H0,10X,11HFREQUENCY =,1PE11.3,10H CYCLE/SEC)
      END
```

```
      SUBROUTINE DATRD(DATA)
C                  CARD-READ SUBROUTINE 'DATRD(DATA(I))'
      DIMENSION DRBU(12),DATA(1),DDRBU(10)
      DATA ATEST/5HALPHA/,DTEST/1H /,ETEST/1H-/,STEST/1H+/
      DATA Z/1H /
      IR=5
      IW=6
    1 READ   (IR,2) EMIN,ALP,IND,(DRBU(I),I=1,12)
    2 FORMAT(A1,A5,I6,11A6,A2)
      IF(IND.EQ.0) GO TO 20
      IF (EMIN.NE.STEST) GO TO 105
C     NEW WING IF COLUMN 1 CONTAINS A PLUS SIGN
C     INITIALIZATION OF DATA ARRAY
      DO 101 I=1,1015
  101 DATA(I)=0.0
      DO 102 I=1,22
  102 DATA(I)=Z
      DO 103 I=104,700,4
  103 DATA(I)=1.0
  105 CONTINUE
      IF (ALP.EQ.ATEST) GO TO 9
      IF (ALP.NE.DTEST) GO TO 8
C     NUMERIC CARD
      DO 3 I=1,10
      DDRBU(I)=DRBU(I)
    3 CONTINUE
      DECODE(DDRBU,990)(DRBU(I),I=1,5)
      DO 5 I=1,5
      IF(DRBU(I))4,6,4
C     TEST FOR BLANK FIELD
    6 IF(SIGN (1.0,DRBU(I)))5,5,4
    4 DATA(IND)=DRBU(I)
    5 IND=IND+1
      GOTO 11
C     ALPHA CARD
    9 DO 10 I=1,10
      DATA(IND)=DRBU(I)
   10 IND=IND + 1
   11 IF (EMIN.NE.ETEST) GO TO 1
C     RETURN IF COLUMN 1 CONTAINS A MINUS SIGN
   13 RETURN
C     END OF DATA CARDS
   20 WRITE(IW,995)
      STOP
C     BAD CARD
    8 CONTINUE
      WRITE
     0   (IW,993) EMIN,ALP,IND,(DRBU(I),I=1,12)
      WRITE   (IW,991)
      STOP
  990 FORMAT(5E12.0)
  991 FORMAT(38H BAD DATA ON THIS CARD. JOB TERMINATED   )
  993 FORMAT(12H0CARD IMAGE=A1,A5,I6,12A6)
  995 FORMAT(///1H ,10X,17HNO MORE DATA CARD/)
      END
```

```
      SUBROUTINE SHAPE
      COMMON     A(2,100,50),W(2,50,50),DA(1015),PS(2,5,50),DF(2,5,50)
     1          ,ML(2,50),AXY(9,9),AY(9),XEDGI(8),YEDGI(8)
     2          ,IEDG(2),NS(2),EM(50,50),XEDG(32),YEDG(32),HM(5,5)
     3          ,AREA,D,DI,DH,CK,L,M,NEW,IR,IW
C     FOLLOWING COMMON IS A TEMPORARY STORAGE IN THIS ROUTINE
      COMMON     ER(5)
      ERRR=1.E-06
      IEDG(NEW)=0
      IF(NEW.EQ.2) GO TO 200
      NS(1)=DA(29)
      IF (NS(1)) 750,750,100
  100 IF(NS(1)-6) 105,105,750
  105 NSP=NS(1)+1
      IF (DA(24)) 755,755,110
  110 DO 115 I=1,NSP
      XEDG(I)=DA(2*I+27)/DA(24)
  115 YEDG(I)=DA(2*I+28)/DA(24)
      XEDG(1)=0.0
      XEDG(NSP+1)=1.0
      YEDG(NSP+1)=YEDG(NSP)
      DO 120 I=1,8
      XEDGI(I)=XEDG(I)
  120 YEDGI(I)=YEDG(I)
      GO TO 500
C     FOR MODIFIED WING
C     COMPUTE MACH DISTRIBUTION ON ORIGINAL WING
  200 DUM=0.
      DO 220 I=1,L
      X=DH+FLOAT(I-1)*D
      JML=ML(1,I)
      DO 220 J=1,JML
      Y=DH+FLOAT(J-1)*D
      CALL MACH(X,Y,EMLOC,DUM,HM,1)
  220 EM(J,I)=EMLOC
      K=1
      NSP=1.+DA(27)/2. + 1.E-07
      D2=2.*D
C     DEFINE EDGES OF MODIFIED WING
      DO 240 I=2,NSP
      XEDG(I)=FLOAT(I-1)*D2
      IF(XEDG(I).GT.XEDGI(K)) K=K+1
      YEDG(I)=YEDGI(K-1)+(YEDGI(K)-YEDGI(K-1))*(XEDG(I)-XEDGI(K-1))/
     $               (XEDGI(K)-XEDGI(K-1))
  240 CONTINUE
      IF(IFIX(DA(27))-2*(NSP-1)) 710,255,250
  250 NSP=NSP+1
      NS12=NS(1)+2
      XEDG(NSP)=XEDGI(NS12)
      YEDG(NSP)=YEDGI(NS12)
  255 CONTINUE
      K1=2
      K2=NSP
      NS12=NS(1)+2
      DO 300 I=2,NS12
```

```
      DO 260 K=K1,K2
      K3=K
      IF(ABS(XEDGI(I)-XEDG(K)).LT.ERRR) GO TO 290
      IF(XEDG(K).GT.XEDGI(I)) GO TO 270
  260 CONTINUE
      GO TO 290
  270 K4=K2+K3
      DO 280 J=K3,K2
      K=K4-J
      XEDG(K+1)=XEDG(K)
  280 YEDG(K+1)=YEDG(K)
      NSP=NSP+1
      XEDG(K3)=XEDGI(I)
      YEDG(K3)=YEDGI(I)
      K2=K2+1
  290 K1=K1+1
  300 CONTINUE
      IF(DA(1013)) 310,320,310
  310 K1=1
      NS(NEW)=NSP
      NS12=NSP+1
      WRITE
     O    (IW,40) K1,NEW,NS(NEW),NEW,IEDG(NEW)
      WRITE
     O    (IW,50) (K,XEDG(K),YEDG(K),K=1,NS12)
  320 CONTINUE
      IF(NSP.GT.32) GO TO 720
C     CHECK IF MODIFIED WING HAS A FOLD-OVER EDGE
C     TRANSFORM Y-COORDINATE
      DO 350 I=1,NSP
      YMP=0.
      Y1=0.
      Y=DH
  330 CALL MACH(XEDG(I),Y,EMLOC,DMDY,HM,2)
      YM=Y*EMLOC
      YMD=YM-YMP
      IF(Y1.GT.Y) YMD=-YMD
      IF(YMD.GE.0.) GO TO 335
      ER(1)=XEDG(I)
      ER(2)=Y
      ER(3)=EMLOC
      ER(4)=DMDY
      ER(5)=YMP
      IER=5
C     ACCEPT A FOLD-OVER OF NOT MORE THAN 1/5 OF A BOX
      IF(ABS(YMD).GT.0.4*DH) GO TO 715
      IERROR=330
      WRITE
     O    (IW,20) IERROR,(ER(J),J=1,IER)
  335 IF(Y.EQ.YEDG(I)) GO TO 340
      Y1=Y
      Y=Y+D
      YMP=YM
      IF(Y.GT.YEDG(I)) Y=YEDG(I)
      GO TO 330
  340 YEDG(I)=YEDG(I)*EMLOC
  350 CONTINUE
      IF(DA(1013)) 360,370,360
```

```
  360 K1=2
      WRITE
     O    (IW,40) K1,NEW,NS(NEW),NEW,IEDG(NEW)
      WRITE
     O    (IW,50) (K,XEDG(K),YEDG(K),K=1,NS12)
  370 CONTINUE
C     READJUST THE POINT DISTRIBUTION TO DEFINE NS(2)
      K1=0
      ICHECK=0
      YMP=0.
      DO 450 I=2,NSP
      YMP=AMAX1(YMP,YEDG(I))
      IF(YEDG(I)-YEDG(I-1)) 400,420,440
C     ACCEPT SMALL WAKE BEHIND TRANSFORMED WING LEADING EDGE  AS PART
C        OF WING, IF NOT MORE THAN 1/5 OF A BOX
  400 ER(1)=XEDG(I)
      ER(2)=YEDG(I)
      ER(3)=XEDG(I-1)
      ER(4)=YEDG(I-1)
      ER(5)=YMP
      IER=5
      IF(YMP-YEDG(I).GT.0.4*DH) GO TO 705
      IERROR=400
      WRITE
     O    (IW,20) IERROR,(ER(J),J=1,IER)
      YEDG(I) = YMP
  420 IF(I.EQ.ICHECK+1) GO TO 430
      ICHECK=I
      GO TO 440
  430 ICHECK=I
      K1=K1+1
  440 K=I-K1
      XEDG(K)=XEDG(I)
      YEDG(K)=YEDG(I)
  450 CONTINUE
      IF(ABS(1.-XEDG(K)).GT.ERRR) GO TO 700
      IF(YEDG(K)-YEDG(K-1)) 705,460,470
  460 NS(2)=K-2
      GO TO 480
  470 NS(2)=K-1
  480 NSP=NS(2)+1
      IF(DA(1013)) 490,495,490
  490 K1=3
      WRITE
     O    (IW,40) K1,NEW,NS(NEW),NEW,IEDG(NEW)
      WRITE
     O    (IW,50) (K,XEDG(K),YEDG(K),K=1,NS12)
  495 CONTINUE
      Y1=YEDG(1)
  500 CONTINUE
      IF (Y1) 760,520,520
  520 K=0
      X=DH
      IF(NEW.EQ.2) GO TO 530
      AREA=0.0
  530 DO 600 I=1,L
```

```
540 IF (X-XEDG(K+1)) 550,550,570
550 ML(NEW,I)=0.5+DI*(YEDG(K)+F*(X-XEDG(K))/G)
    GO TO 600
570 K=K+1
    G=XEDG(K+1)-XEDG(K)
    F=YEDG(K+1)-YEDG(K)
    IF (G) 765,580,580
580 IF (F) 770,590,590
590 IF(NEW.EQ.2) GO TO 540
    AREA=AREA+G*(YEDG(K+1)+YEDG(K))
    GO TO 540
600 X=X+D
    DO 610 I=2,L
    K=I
    IF(ML(NEW,I).LT.ML(NEW,I-1)) GO TO 725
610 CONTINUE
    K=L
    IF(ML(NEW,K).GT.50) GO TO 725
    IF(NEW.EQ.1) GO TO 650
    IF(DA(1012)) 620,650,620
C   PRINT OUT MACH DISTRIBUTION
620 WRITE(IW,60)
    DO 640 I=1,L
    JL=ML(1,I)
    IF(JL.EQ.0) GO TO 640
    WRITE
   0   (IW,70)  I
    JLP=JL/6
    IF(JL-6*JLP.NE.0) JLP=JLP+1
    DO 630 J=1,JLP
630 WRITE
   0   (IW,80) ((J1,EM(J1,I)),J1=J,JL,JLP)
640 CONTINUE
650 CONTINUE
    NS12=NS(NEW)
    IF(XEDG(NSP).GT.1.) GO TO 765
    IF(XEDG(NSP)-1.0+ERRR) 680,670,670
670 IF (YEDG(NSP)-YEDG(NS12)) 680,680,690
680 IEDG(NEW)=1
690 CONTINUE
    RETURN
700 IERROR=700
    ER(1)=XEDG(K)
    ER(2)=YEDG(K)
    IER=2
    GO TO 740
705 IERROR=705
    GO TO 740
710 IERROR=710
    ER(1)=DA(27)
    ER(2)=NSP
    IER=2
    GO TO 740
715 IERROR=715
    GO TO 740
720 IERROR=720
    ER(1)=XEDG(NSP)
```

```
      ER(2)=YEDG(NSP)
      ER(3)=NSP
      IER=3
      GO TO 740
  725 IERROR=725
      ER(1)=NEW
      ER(2)=K
      ER(3)=ML(NEW,K)
      ER(4)=ML(NEW,K-1)
      IER=4
  740 WRITE
     O   (IW,20) IERROR, (ER(I),I=1,IER)
      STOP
  750 IPR=29
      GO TO 790
  755 IPR=24
      GO TO 790
  760 IPR=30
      GO TO 790
  765 K=  MINO(K,NS(1))
      IPR=2*K+29
      GO TO 790
  770 IPR=2*K+30
  790 WRITE
     O  (IW,10)IPR
   10 FORMAT(1H0,10X,17HSHAPE -- BAD DATA,I5)
   20 FORMAT(1H0,10X,38HBAD NUMBER IN SHAPE NEAR STATEMENT NO.,I5,
     1       /1H ,15X,1P5E14.6)
   40 FORMAT(/1H0,5X,3HNO.,I2,43H REDISTRIBUTION OF WING LEADING EDGES,
     1 NS(,I1,4H) = ,I2,8H,  IEDG(,I1,4H) = ,I1)
   50 FORMAT(4(6X,I3,1P2E11.4))
   60 FORMAT(1H1, 5X,30HLOCAL MACH NUMBER DISTRIBUTION/)
   70 FORMAT(1H0,5X,I2,6HTH ROW)
   80 FORMAT(1H ,5X,6(2X,I3,1PE$3.5))
      STOP
      END


      SUBROUTINE MACH(X,Y,EMLOC,DMDY,HM,K)
C     CALCULATE LOCAL MACH NUMBER
      DIMENSION HM(5,5)
      Y2=Y*Y
      YP=1.0
      EMLOC=0.0
      DMDY=0.0
      DO 20 J=1,5
      XYP=YP
      C=FLOAT(J)
      DO 10 I=1,5
      EMLOC=EMLOC+HM(I,J)*XYP
      IF(J.EQ.5.OR.K.EQ.1) GO TO 10
      DMDY=DMDY+C*HM(I,J+1)*XYP
   10 XYP=X*XYP
   20 YP=Y2*YP
      DMDY=2.*Y*DMDY
      RETURN
      END
```

```
      SUBROUTINE FORCI
C     CONTROL SUBROUTINE FOR CALCULATION OF INTEGRALS USED
C     TO FIND GENERALIZED FORCES
      COMMON    A(2,100,50),W(2,50,50),DA(1015),PS(2,5,50),DF(2,5,50)
     1          ,ML(2,50),AXY(9,9),AY(9),XEDGI(8),YEDGI(8)
     2          ,IEDG(2),NS(2),EM(50,50),XEDG(32),YEDG(32),HM(5,5)
     3          ,AREA,D,DI,DH,CK,L,M,NEW,IR,IW
C     FOLLOWING COMMON COMMUNICATES BETWEEN 'FORCI' AND 'SECT'
      COMMON YMAX2,N
      NSS=NS(1)+1
      YMAX2=YEDG(NSS)*YEDG(NSS)
      NSS=NS(1)
      DO 3 J=1,9
      DO 2 I=1,9
    2 AXY(I,J)=0.0
    3 AY(J)=0.0
      IF (DA(30)) 4,5,4
    4 N=0
      CALL SECT
C     SECT DOES THE CALCULATIONS FOR EACH SECTION OF THE PLANFORM
    5 DO 7 I=1,NSS
      IF (YEDG(I)-YEDG(I+1)) 6,7,7
    6 N=I
      CALL SECT
    7 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SECT
C
C     AXY(I,J) IS THE INTEGRAL OVER THE PLANFORM OF
C              X**(I-1)*Y**(2J-2)*P(X,Y)*Q(Y)
C
C     IN TERMS OF THE EQUATION OF THE LEADING EDGE X = X0(Y)
C              P(X,Y) = SQRTF(X**L-X0**L)
C     WHERE L = 1 OR 2
C              Q(Y) = 1 OR SQRTF(1-(Y/YMAX)**2)
C     DEPENDING ON THE WING SHAPE
C
C     AY(J) IS THE INTEGRAL OVER THE TRAILING EDGE OF
C              Y**(2J-2)*P(1,Y)*Q(Y)
C
      COMMON    A(2,100,50),W(2,50,50),DA(1015),PS(2,5,50),DF(2,5,50)
     1          ,ML(2,50),AXY(9,9),AY(9),XEDGI(8),YEDGI(8)
     2          ,IEDG(2),NS(2),EM(50,50),XEDG(32),YEDG(32),HM(5,5)
     3          ,AREA,D,DI,DH,CK,L,M,NEW,IR,IW
C     FOLLOWING COMMON COMMUNICATES BETWEEN 'FORCI' AND 'SECT'
      COMMON YMAX2,N
      DIMENSION U(6),H(6)
C     COMPUTES THE CONTRIBUTION TO AXY,AY OF THE SECTION OF WING
C     BOUNDED BY A SEGMENT OF THE LEADING EDGE, THE TRAILING EDGE,
C     AND TWO LINES ON WHICH Y IS CONSTANT
C     A SIX POINT GAUSSIAN FORMULA IS USED FOR THE INTEGRATION
C     OVER EACH VARIABLE
```

```
      U(1)=0.61930959
      U(2)=0.83060469
      U(3)=0.96623476
      U(4)=0.03376524
      U(5)=0.16939531
      U(6)=0.38069041
      H(1)=0.23395697
      H(2)=0.18038079
      H(3)=0.08566225
      H(4)=H(3)
      H(5)=H(2)
      H(6)=H(1)
C     GAUSSIAN POINTS AND WEIGHTS FOR THE INTERVAL (0,1)
      X2=XEDG(N+1)
      X1=XEDG(N)
      Y2=YEDG(N+1)
      Y1=YEDG(N)
      IF (N) 4,2,4
    2 X1=0.0
      Y1=0.0
    4 DY=Y2-Y1
      DO 19 J=1,6
      V=U(J)
      G=H(J)*DY
      IF (Y2**2-YMAX2) 7,6,7
    6 G=2.0*V*G
      V=V*V
    7 Y=Y2-V*DY
      X0=X2+V*(X1-X2)
      X0P=1.0-X0
      G=G* SQRT(X0P)
      YQ=Y*Y
      IF (IEDG(NEW)) 8,9,8
    8 G=G* SQRT(1.0-YQ/YMAX2)
    9 E=2.0*X0P*G
      IF (DA(30)) 10,10,11
   10 G=G* SQRT(1.0+X0)
   11 DO 17 I=1,6
      U2=U(I)**2
      X=X0+X0P*U2
      F=E*H(I)*U2
      IF (DA(30)) 12,12,13
   12 F=F* SQRT(X+X0)
   13 YP=1.0
      DO 16 M=1,9
      XP=YP
      DO 15 IL=1,9
      AXY(IL,M)=AXY(IL,M)+XP*F
   15 XP=X*XP
   16 YP=YQ*YP
   17 CONTINUE
      YP=1.0
      DO 18 M=1,9
      AY(M)=AY(M)+YP*G
   18 YP=YQ*YP
   19 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE POT2(M2,M0,N0,CK,D)
C
C     THE VELOCITY FIELD OF A UNIFORM DOUBLET DISTRIBUTION
C     OVER A BOX IS COMPUTED AT ALL POINTS AT WHICH IT WILL BE
C     NEEDED AND STORED IN THE ARRAY A IN COMMON
C
C     M0,N0 CONTROL THE NUMBER OF VALUES COMPUTED
C
C     M2 IS THE RANGE OF THE SECOND SUBSCRIPT IN THE ARRAY,
C     DIMENSIONED A(2,M2,N2), BUT TREATED HERE AS AN ARRAY
C     WITH TWO SUBSCRIPTS
C
      DIMENSION A(2,1)
      COMMON A
      M=M0
      N=N0
      DK=CK*D
      DK2=DK**2
      M1=M-1
      DK8=DK2/8.0
      DK4=2.0*DK8
      DK12=DK2/12.0
      CM=0.5
      DH=DK*0.5
      DM=0.5*DH
      DD=2.0*DK
      DDM=DD
      D1=0.25*DK2
      B5=DK2/24.0
      DO 3 I=1,M
      B1=0.0
      B4=2.0/DM
      B2=B5/B4-DH
      B3=-0.5*B5
      D3=DH*B4+B5
      D4=DK8*B4
      DD4=2.0*D4
      CN=1.0
      K=I
      C3=0.0
      C4=0.0
      C7=0.0
      C8=0.0
      DO 2 J=1,N
      A1=DM/CN
      C1=CM* COS(A1)
      C2=-CM* SIN(A1)
      C5=CM*CIN(A1,C6)
      C6=-CM*C6
      C9=C1-C3
      C10=C2-C4
      C11=C5-C7
      C12=C6-C8
      A(1,K)=B3*C9-B4*C10-B5*C3-B1*C11-B2*C12
      A(2,K)=B4*C9+B3*C10-B5*C4+B2*C11-B1*C12
   23 C3=C1
      C4=C2
```

```
      C7=C5
      C8=C6
      B1=B1-D1
      B3=B3-D3
      B4=B4-D4
      D4=D4+DD4
      CN=CN+2.0
    2 K=K+M2
      CM=CM+1.0
      DM=DM+DDM
    3 DDM=DDM+DD
      DO 5 IL=1,2
      K1=1
      DO 5 J=1,N
      DO 4 I=1,M1
      K=K1+M-I
    4 A(IL,K)=A(IL,K)-A(IL,K-1)
      A(IL,K1)=2.0*A(IL,K1)
    5 K1=K1+M2
      CM=0.0
      DM=0.0
      DDM=DK
      DO 12 I=1,M
      C7=0.0
      C8=0.0
      C9=0.0
      C10=0.0
      P1=0.0
      P2=0.0
      CN=1.0
      B6=0.5*DK12
      K=I
      DO 10 J=1,N
      A1=CM/CN
      A2=DM/CN
      IF (A1-0.2) 7,7,8
    7 B1=2.0-A1**2/3.0
      B2=-DK/(6.0*CN)
      GO TO 9
    8 B3= SIN(A1)/A1
      B1=2.0*B3
      B2=(B3- COS(A1))/A2-DH/CN*B3
    9 B3= COS(A2)/CN
      B4= SIN(A2)/CN
      C3=B1*B3+B2*B4
      C4=B2*B3-B1*B4
      B5=DH*CN
      C1=B5*C4-2.0*C3
      C2=-2.0*C4-B5*C3
      C5=C1-C7
      C6=C2-C8
      P3=P2-B6*CN
      P4=P3+2.0*DK12*(CN-1.0)
      A(1,K)=A(1,K)+C5-P1*C6+P3*C3-P4*C9
      A(2,K)=A(2,K)+C6+P1*C5+P3*C4-P4*C10
      P1=P1+DH
      P2=P2+CN*DK4
      CN=CN+2.0
      C7=C1
```

```
      C8=C2
      C9=C3
      C10=C4
      B6=B6+DK12
   10 K=K+M2
      CM=CM+DK
      DM=DM+DDM
   12 DDM=DDM+DD
      D3=CK/(2.0*3.14159265)
      M1=M2-M
      K=1
      A1=0.0
      DO 14 J=1,N
      C1=D3* SIN(A1)
      C2=-D3* COS(A1)
      DO 13 I=1,M
      DFE   =A(1,K)*C1+A(2,K)*C2
      A(2,K)=A(2,K)*C1-A(1,K)*C2
      A(1,K)=DFE
   13 K=K+1
      K=K+M1
   14 A1=A1+DH
      RETURN
      END


      SUBROUTINE DRED
      COMMON     A(2,100,50),W(2,50,50),DA(1015),PS(2,5,50),DF(2,5,50)
     1           ,ML(2,50),AXY(9,9),AY(9),XEDGI(8),YEDGI(8)
     2           ,IEDG(2),NS(2),EM(50,50),XEDG(32),YEDG(32),HM(5,5)
     3           ,AREA,D,DI,DH,CK,L,M,NEW,IR,IW
C     FOLLOWING COMMON COMMUNICATES BETWEEN 'DRED' AND 'LSUQA'
      COMMON     XQ(150),YQ(150),DEFQ(150),WTQ(150),COE(5,5)
      NCONT=NEW+1000
      IF(NEW.EQ.2) GO TO 400
      NP=DA(98)
      IF (NP) 730,170,100
C     A POLYNOMIAL FOR THE DEFLECTION IS FITTED TO VALUES
C     OF DEFLECTION AT GIVEN POINTS.
  100 NX=DA(99)
      NY=DA(100)
      IF (NX) 710,710,105
  105 IF (NY) 720,720,110
  110 IF (150-NP) 730,115,115
  115 IF (NP-NX) 710,120,120
  120 IF (NP-NY) 720,125,125
  125 CONTINUE
      KP=100
      DO 140 IP=1,NP
      XQ(IP)=DA(KP+1)/DA(24)
      YQ(IP)=DA(KP+2)/DA(24)
      DEFQ(IP)=DA(KP+3)
      WTQ(IP)=DA(KP+4)
      IF(WTQ(IP)) 740,740,140
  140 KP=KP+4
C     USE LEAST SQUARE METHOD TO CURVE-FIT DATA
      IF(DA(NCONT)) 150,160,150
```

```
  150 WRITE
     O   (IW,30) M
      WRITE(IW,10)
  160 CONTINUE
      CALL LSQUA(NP,NX,NY,NCONT)
      GO TO 200
  170 YP=1.0
      K=1
      DO 190 J=1,5
      XYP=YP
      DO 180 I=1,5
      COE(I,J)=XYP*DA(K+45)
      K=K+1
  180 XYP=XYP*DA(24)
  190 YP=YP*DA(24)**2
  200 K=25*(M-1)
      DO 220 I=1,25
      K=K+1
  220 DF(1,K,1)=COE(I,1)
      RETURN
C     FOR MODIFIED WING
C     1-CALCULATE DEFLECTION AT 100 OR LESS ON THE ORIGINAL WING
C     2-TRANSFORM X, Y, DEF TO THE TRANSFORMED PLANE
C     3-CURVE-FIT THE DATA
  400 MM=5*(M-1)
      J1=MM+1
      J2=MM+5
      NTOT=0
      DO 410 I=1,L
  410 NTOT=NTOT+ML(1,I)
      IF(DA(1014)) 420,430,420
  420 WRITE(IW,50)
  430 CONTINUE
      NJ=FLOAT(NTOT)/100. + 0.5
      IF(NJ.LT.1) NJ=1
      NPT=0
      NP=0
      K=0
      DO 520 I=1,NTOT,NJ
      NP=NP+1
  440 IF(I.LE.NPT) GO TO 450
      K=K+1
      NPTP=NPT
      NPT=NPT+ML(1,K)
      GO TO 440
  450 XQ(NP)=DH+FLOAT(K-1)*D
      JY=I-NPTP
      YQ(NP)=DH+FLOAT(JY-1)*D
      Y2=YQ(NP)*YQ(NP)
      DEFQ(NP)=0.
      WTQ(NP)=1.0
      YP=1.0
      DO 470 J=J1,J2
      XYP=YP
      DO 460 IX=1,5
      DEFQ(NP)=DEFQ(NP)+DF(1,IX,J)*XYP
  460 XYP=XYP*XQ(NP)
  470 YP=YP*Y2
      IF(DA(1014)) 480,490,480
```

```
  480 Y2=YQ(NP)
      YP=DEFQ(NP)
  490 CONTINUE
      YQ(NP)=YQ(NP)*EM(JY,K)
C     DEFQ(NP)=DEFQ(NP)/(EM(JY,K)+1.E-10)
      DEFQ(NP)=DEFQ(NP)
      IF(DA(1014)) 500,520,500
  500 WRITE
     O    (IW,60) I,K,JY,NP,XQ(NP),YQ(NP),Y2,DEFQ(NP),YP,EM(JY,K)
  520 CONTINUE
C     USE LEAST SQUARE METHOD TO CURVE-FIT DATA
      IF(DA(NCONT)) 550,560,550
  550 WRITE
     O    (IW,40) M
      WRITE(IW,10)
  560 CONTINUE
C
      CALL LSQUA(NP,NP,NP,NCONT)
      DO 580 J=1,5
      DO 580 I=1,5
  580 DF(2,I,MM+J)=COE(I,J)
      RETURN
  710 IPR=99
      GO TO 750
  720 IPR=100
      GO TO 750
  730 IPR=98
      GO TO 750
  740 IPR=K+4
  750 WRITE
     O  (IW,20)IPR
      STOP
   10 FORMAT(1H010X,56HCOMPUTED DEFLECTION = SUM OF DEF(N,M)*X**(N-1)*Y*
     1*(2M-2)/1H010X,54H(IN DIMENSIONLESS COORDINATES - DISTANCE/CHORD L
     2ENGTH)/1H09X,1HN7X,1HM16X,8HDEF(N,M))
   20 FORMAT(1H0,10X,16HDRED -- BAD DATA,I5)
   30 FORMAT(1H0,8X,26HPHYSICAL PLANE -- MODE NO.,I3)
   40 FORMAT(1H0,8X,29HTRANSFORMED PLANE -- MODE NO.,I3)
   50 FORMAT(1H1, 5X,42HTRANSFORMED DEFLECTION INFORMATION -- DRED/)
   60 FORMAT(1H , 5X,4I5,1P8E12.5)
      END


      SUBROUTINE LSQUA(NP,NX,NY,NRITE)
C     CURVE FITS WITH LEAST SQUARE METHOD
      COMMON    A(2,100,50),W(2,50,50),DA(1015),PS(2,5,50),DF(2,5,50)
     1          ,ML(2,50),AXY(9,9),AY(9),XEDGI(8),YEDGI(8)
     2          ,IEDG(2),NS(2),EM(50,50),XEDG(32),YEDG(32),HM(5,5)
     3          ,AREA,D,DI,DH,CK,L,M,NEW,IR,IW
C     FOLLOWING COMMON ONLY COMMUNICATES WITH THE CALLING ROUTINE
      COMMON    XQ(150),YQ(150),DEFQ(150),WTQ(150),COE(5,5)
C     FOLLOWING COMMON IS USED AS A TEMPORARY STORAGE IN THIS ROUTINE
      COMMON    IXB(5),B(25),C(25,25),G(25)
      RITE=DA(NRITE)
      MX=  MIN0(NX,5)
      MY=  MIN0(NY,5)
```

```
      IY=MY
      IXY=MX+MY
      DO 2 J=1,5
      DO 1 I=1,5
    1 COE(I,J)=0.0
    2 IXB(J)=MX
      NC=MX*MY
    3 DO 5 I=1,NC
      DO 4 J=1,NC
    4 C(I,J)=0.0
    5 B(I)=0.0
      DO 11 IP=1,NP
      X=XQ(IP)
      Y2=YQ(IP)*YQ(IP)
      DEF=DEFQ(IP)
      WT=WTQ(IP)
      YP=1.0
      K=1
      DO 8 J=1,IY
      XYP=YP
      JX=IXB(J)
      DO 7 I=1,JX
      G(K)=XYP
      XYP=X*XYP
    7 K=K+1
    8 YP=Y2*YP
      DO 10 I=1,NC
      DO 9 J=1,NC
    9 C(I,J)=C(I,J)+G(I)*G(J)*WT
   10 B(I)=B(I)+G(I)*DEF*WT
   11 CONTINUE
      K=MSIMER(25,NC,1,C,B)
      IF(K-1)22,22,15
   15 DO 16 I=1,IY
      IP=IY+1-I
      IF (IXB(IP)+IP-IXY) 16,17,17
   16 CONTINUE
      IXY=IXY-1
      GO TO 15
   17 IXB(IP)=IXB(IP)-1
      IF (IXB(IP)) 18,18,19
   18 IY=IP-1
   19 NC=0
      DO 20 I=1,IY
   20 NC=NC+IXB(I)
      GO TO 3
   22 K=1
      DO 23 J=1,IY
      JX=IXB(J)
      DO 23 I=1,JX
      COE(I,J)=B(K)
   23 K=K+1
      IF (RITE) 61,66,61
   61 CONTINUE
      DO 64 J=1,IY
      JX=IXB(J)
      DO 63 I=1,JX
      WRITE
     0     (IW,42)I,J,COE(I,J)
```

```
   63 CONTINUE
   64 CONTINUE
   66 CONTINUE
   42 FORMAT(3X,2I8,1PE25.5)
      RETURN
      END
```

```
      SUBROUTINE WVAL
C     EVALUATION OF THE UPWASH ARRAY
      COMMON    A(2,100,50),W(2,50,50),DA(1015),PS(2,5,50),DF(2,5,50)
     1          ,ML(2,50),AXY(9,9),AY(9),XEDGI(8),YEDGI(8)
     2          ,IEDG(2),NS(2),EM(50,50),XEDG(32),YEDG(32),HM(5,5)
     3          ,AREA,D,DI,DH,CK,L,M,NEW,IR,IW
C     FOLLOWING COMMON IS USED AS A TEMPORARY STORAGE IN THIS ROUTINE
      COMMON    G(5,5),H(5,5)
      J1=5*(M-1)
      DO 3 J=1,5
      J1=J1+1
      CI=1.0
      DO 2 I=1,5
      IF(I.LT.5) G(I,J)=CI*DF(NEW,I+1,J1)
      H(I,J)=CK*DF(NEW,I,J1)
    2 CI=CI+1.0
    3 G(5,J)=0.0
C     G AND H ARE THE COEFFICIENTS OF THE REAL AND IMAGINARY PARTS
C      OF THE UPWASH
      X=DH
      DO 10 I=1,L
      JL=ML(NEW,I)
      IF (JL) 5,10,5
    5 Y=DH
      DO 9 J=1,JL
      Y2=Y*Y
      W(1,J,I)=0.0
      W(2,J,I)=0.0
      YP=1.0
      DO 8 J1=1,5
      XYP=YP
      DO 7 I1=1,5
      W(1,J,I)=W(1,J,I)+G(I1,J1)*XYP
      W(2,J,I)=W(2,J,I)+H(I1,J1)*XYP
    7 XYP=X*XYP
    8 YP=Y2*YP
    9 Y=Y+D
   10 X=X+D
      RETURN
      END
```

```
      SUBROUTINE BOXPO
C     SOLUTION OF SIMULTANEOUS EQUATIONS FOR THE POTENTIAL
      COMMON     A(2,100,50),S(2,50,50),DA(1015),PS(2,5,50),DF(2,5,50)
     1          ,ML(2,50),AXY(9,9),AY(9),XEDGI(8),YEDGI(8)
     2          ,IEDG(2),NS(2),EM(50,50),XEDG(32),YEDG(32),HM(5,5)
     3          ,AREA,D,DI,DH,CK,L,M,NEW,IR,IW
C     FOLLOWING COMMON IS USED AS A TEMPORARY STORAGE IN THIS ROUTINE
      COMMON     E(2,50,51)
      I1=0
      DO 9 I=1,L
C     ADJUST UPSTREAM INFLUENCE
      K0 = 1
      NFLNS = DA(43)
      IF(NFLNS.EQ.0) GO TO 3
      K0 = MAX0(1,I-NFLNS+1)
    3 CONTINUE
      JL=ML(NEW,I)
      IF (JL.EQ.0) GO TO 9
      IF (I1.EQ.0) GO TO 6
C     SUBTRACTION OF CONTRIBUTIONS OF PRECEDING ROWS TO UPWASH
      DO 5 J=1,JL
      DO 5 K=K0,I1
      KL=ML(NEW,K)
      K1=I+1-K
      IF (KL.EQ.0) GO TO 5
      DO 4 N=1,KL
      N1=N+J
      N2=IABS(N-J)+1
      A1=A(1,N1,K1)+A(1,N2,K1)
      A2=A(2,N1,K1)+A(2,N2,K1)
      S(1,J,I)=S(1,J,I)-A1*S(1,N,K)+A2*S(2,N,K)
    4 S(2,J,I)=S(2,J,I)-A2*S(1,N,K)-A1*S(2,N,K)
    5 CONTINUE
C     SETTING UP MATRIX FOR SIMULTANEOUS EQUATIONS
    6 DO 8 J=1,JL
      DO 8 K=1,J
      N1=J+K
      N2=IABS(J-K)+1
      E(1,J,K)=A(1,N1,1)+A(1,N2,1)
      E(2,J,K)=A(2,N1,1)+A(2,N2,1)
      E(1,K,J)=E(1,J,K)
    8 E(2,K,J)=E(2,J,K)
C     SOLUTION OF EQUATIONS
      K = MSIMEC(50,JL,1,E,S(1,1,I))
      IF(K.NE.1) GO TO 12
    9 I1=I1+1
      RETURN
   12 WRITE (IW,41)
   41 FORMAT(1H010X,59HSOLUTION OF SIMULTANEOUS EQUATIONS FOR THE POTENT
     1IAL FAILED)
      STOP
      END
```

```
      SUBROUTINE BOXP
      COMMON    A(2,100,50),S(2,50,50),DA(1015),PS(2,5,50),DF(2,5,50)
     1          ,ML(2,50),AXY(9,9),AY(9),XEDGI(8),YEDGI(8)
     2          ,IEDG(2),NS(2),EM(50,50),XEDG(32),YEDG(32),HM(5,5)
     3          ,AREA,D,DI,DH,CK,L,M,NEW,IR,IW
C     FOLLOWING COMMON IS USED AS A TEMPORARY STORAGE IN THIS ROUTINE
      COMMON    EDG(50),PR(2,50),PSI(2,50),G(20),X0(50),IXB(4)
     1          ,C(20,20),B(20,2)
      DIMENSION XTEMP(8),YTEMP(8)
      NEV=NEW
      IF (DA(NEV+1002)) 100,200,100
  100 WRITE  (IW,10)
      IF(NEW.EQ.1) WRITE
     0   (IW,12) M
      IF(NEW.EQ.2) WRITE
     0   (IW,14) M
   10 FORMAT(1H1,10X,47HUPWASH (REAL, IMAGINARY, ABSOLUTE, PHASE ANGLE))
   12 FORMAT(1H+,58X,28H---PHYSICAL PLANE---MODE NO.,I3/)
   14 FORMAT(1H+,58X,31H---TRANSFORMED PLANE---MODE NO.,I3/)
      DO 170 I=1,L
      JL=ML(NEV,I)
      IF (JL) 110,170,110
  110 WRITE
     0  (IW,20)I
      IF (I-1) 120,120,140
  120 DO 130 J=1,JL
      S1=S(1,J,I)*1.57079633
      S2=S(2,J,I)*1.57079633
      S3=SQRT(S1*S1+S2*S2)
      S4=57.29578*ATAN2(S2,S1)
  130 WRITE
     0  (IW,25)  J,S1,S2,S3,S4
      GO TO 170
  140 JLP=JL/2
      IF(JL-2*JLP.NE.0) JLP=JLP+1
      DO 160 J=1,JLP
      S1=SQRT(S(1,J,I)*S(1,J,I)+S(2,J,I)*S(2,J,I))
      S2=57.29578*ATAN2(S(2,J,I),S(1,J,I))
      J1=J+JLP
      IF(J1.LE.JL) GO TO 150
      J1=0
      S3=0.0
      S4=0.0
      GO TO 160
  150 CONTINUE
      S3=SQRT(S(1,J1,I)*S(1,J1,I)+S(2,J1,I)*S(2,J1,I))
      S4=57.29578*ATAN2(S(2,J1,I),S(1,J1,I))
  160 WRITE
     0   (IW,25)J,S(1,J,I),S(2,J,I),S1,S2,J1,S(1,J1,I),S(2,J1,I),S3,S4
  170 CONTINUE
C     THESE ARE THE UPWASHES (ABOVE)
  200 CONTINUE
      CALL BOXPO
C      BOXPO COMPUTES THE POTENTIAL VALUES IN EACH BOX.
C     THEY ARE STORED IN THE ARRAY S.
      IF (DA(NEV+1004)) 210,270,210
  210 WRITE   (IW,30)
```

```
      IF(NEW.EQ.1) WRITE
     0    (IW,32) M
      IF(NEW.EQ.2) WRITE
     0    (IW,34) M
   30 FORMAT(1H1,10X,61HPOTENTIAL CALCULATED (REAL, IMAGINARY, ABSOLUTE,
     $ PHASE ANGLE))
   32 FORMAT(1H+,71X,28H---PHYSICAL PLANE---MODE NO.,I3/)
   34 FORMAT(1H+,71X,31H---TRANSFORMED PLANE---MODE NO.,I3/)
      DO 250 I=1,L
      JL=ML(NEV,I)
      IF (JL) 220,250,220
  220 WRITE
     0  (IW,20)I
      JLP=JL/2
      IF(JL-2*JLP.NE.0) JLP=JLP+1
      DO 240 J=1,JLP
      S1=SQRT(S(1,J,I)*S(1,J,I)+S(2,J,I)*S(2,J,I))
      S2=57.29578*ATAN2(S(2,J,I),S(1,J,I))
      J1=J+JLP
      IF(J1.LE.JL) GO TO 230
      J1=0
      S3=0.0
      S4=0.0
      GO TO 240
  230 CONTINUE
      S3=SQRT(S(1,J1,I)*S(1,J1,I)+S(2,J1,I)*S(2,J1,I))
      S4=57.29578*ATAN2(S(2,J1,I),S(1,J1,I))
  240 WRITE(IW,25) J,S(1,J,I),S(2,J,I),S1,S2
     0            ,J1,S(1,J1,I),S(2,J1,I),S3,S4
  250 CONTINUE
C     THESE ARE THE POTENTIALS (ABOVE)
  270 CONTINUE
C     FIT OF A SERIES TO THE POTENTIAL VALUES
  300 JL=ML(NEV,L)
      NSS=NS(NEV)+1
      DY=D/YEDG(NSS)
      Y=0.5*DY
      DO 330 J=1,JL
      IF (IEDG(NEV)) 310,320,310
  310 EDG(J)= SQRT(1.0-Y*Y)
      Y=Y+DY
      GO TO 330
  320 EDG(J)=1.0
  330 CONTINUE
      N=0.5+DI*YEDG(1)
      IF (N) 360,360,340
  340 DO 350 I=1,N
  350 X0(I)=0.0
  360 X1=0.0
      N1=N
      Y1=YEDG(1)
      NSS=NS(NEV)
      DO 400 K=1,NSS
      X2=XEDG(K+1)
      Y2=YEDG(K+1)
      N=DI*Y2+0.5
      IF (N1-N) 370,390,390
  370 N1=N1+1
      DO 380 I=N1,N
```

```
      Y=D*( FLOAT(I)-0.5)
      X0(I)=X1+(X2-X1)*(Y-Y1)/(Y2-Y1)
  380 CONTINUE
  390 X1=X2
      Y1=Y2
      N1=N
  400 CONTINUE
      AS=0.0
      DO 430 I=1,L
      JL=ML(NEV,I)
      IF (JL) 430,430,410
  410 DO 420 J=1,JL
      AS=AMAX1(AS, ABS(S(1,J,I)), ABS(S(2,J,I)))
  420 CONTINUE
  430 CONTINUE
      IX=5
      IY=4
      IXY=9
  440 DO 450 I=1,IY
  450 IXB(I)=IX
  460 NC=0
      DO 470 I=1,IY
  470 NC=NC+IXB(I)
      DO 490 I=1,NC
      DO 480 J=1,NC
  480 C(I,J)=0.0
      B(I,1)=0.0
  490 B(I,2)=0.0
      X1=DH
      DO 580 I=1,L
      JL=ML(NEV,I)
      Y=DH
      IF (JL) 580,580,510
  510 DO 570 J=1,JL
      XR=X1-X0(J)
      IF(XR.GE.0.) GO TO 515
      IERROR=515
      WRITE
     0    (IW,90) IERROR,NEW,NEV+I,J,X1,X0(J),XR
      XR=0.5*ABS(XR)
  515 CONTINUE
C
      IF(YEDG(1)) 520,520,530
  520 XR=XR*(X1+X0(J))
  530 XR= SQRT(XR)
      YP=1.0
      K=1
      DO 550 N1=1,IY
      XP=XR*YP
      JX=IXB(N1)
      DO 540 N=1,JX
      G(K)=XP*EDG(J)
      XP=X1*XP
  540 K=K+1
  550 YP=Y*Y*YP
      Y=Y+D
      DO 570 N1=1,NC
      DO 560 N=1,NC
  560 C(N1,N)=C(N1,N)+G(N1)*G(N)
```

```
      DO 570 N=1,2
  570 B(N1,N)=B(N1,N)+G(N1)*S(N,J,I)
  580 X1=X1+D
      K=MSIMER(20,NC,2,C,B)
      IF(K-1)640,640,600
C     IF MSIMER FAILS, THE FOLLOWING SECTION REDUCES THE NUMBER OF TERMS
C      IN THE SERIES.
  600 DO 610 I=1,IY
      IP=IY+1-I
      IF (IXB(IP)+IP-IXY) 610,620,620
  610 CONTINUE
      IXY=IXY-1
      GO TO 600
  620 IXB(IP)=IXB(IP)-1
      IF (IXB(IP)) 630,630,460
  630 IY=IP-1
      GO TO 460
  640 K=1
      AC=0.0
      DO 670 I=1,IY
      JX=IXB(I)
      CJ=1.0
      DO 660 J=1,JX
      C(K,1)=B(K,1)
      C(K,2)=B(K,2)
      AC=AMAX1(AC, ABS(C(K,1)), ABS(C(K,2)))
      IF (J.EQ.1) GO TO 660
      B(K-1,1)=CJ*C(K,1)
      B(K-1,2)=CJ*C(K,2)
      CJ=CJ+1.0
  660 K=K+1
      B(K-1,1)=0.0
  670 B(K-1,2)=0.0
      IF (AC-50.0*AS) 690,690,600
C     GO TO 600 IF COEFFICIENTS ARE TOO LARGE.
  690 IF (DA(NEV+1006)) 700,770,700
C     PRINTOUT OF COEFFICIENTS OF POTENTIAL
  700 CONTINUE
      IF(NEV-1) 995,702,703
  702 WRITE
     0    (IW,42) M
      IF(NEW.EQ.1) WRITE(IW,45)
      IF(NEW.EQ.2) WRITE(IW,46)
      GO TO 704
  703 WRITE
     0    (IW,44) M
  704 CONTINUE
   42 FORMAT(1H1,5X,25HPHYSICAL PLANE---MODE NO.,I3 )
   44 FORMAT(1H1,5X,28HTRANSFORMED PLANE---MODE NO.,I3/)
   45 FORMAT(1H+,33X,23H  (NO THICKNESS EFFECT)/)
   46 FORMAT(1H+,33X,25H  (WITH THICKNESS EFFECT)/)
      WRITE  (IW,50)
   50 FORMAT(1H-10X,53HPOTENTIAL = SUM OF PO(M,N)*X**(M-1)*Y**(2N-2)*SQR
     1TF(X)
      IF (YEDG(1)) 710,710,730
  710 WRITE  (IW,52)
      IF (IEDG(NEV)) 720,750,720
  720 WRITE  (IW,54)
      GO TO 750
```

```
 730 WRITE  (IW,56)
     IF (IEDG(NEV)) 740,750,740
 740 WRITE  (IW,58)
 750 WRITE  (IW,60)
  52 FORMAT(1H+63X,10H**2-X0**2))
  54 FORMAT(1H+73X,21H*SQRTF(1-(Y/YMAX)**2))
  56 FORMAT(1H+63X,4H-X0))
  58 FORMAT(1H+67X,21H*SQRTF(1+(Y/YMAX)**2))
  60 FORMAT(1H015X,52HWHERE X = X0(Y) IS THE EQUATION OF THE LEADING ED
    1GE./1H020X,21HCOEFFICIENTS  PO(M,N)/1H07X,1HM7X,1HN14X,9HREAL PART
    216X,10HIMAG. PART)
     K=1
     DO 760 I=1,IY
     JL=IXB(I)
     DO 760 J=1,JL
     WRITE
    0   (IW,65) J,I,C(K,1),C(K,2)
 760 K=K+1
     WRITE  (IW,70)
 770 IF(DA(NEV+1008)) 780,860,780
C     PRINTOUT OF VALUES OF POTENTIAL AND PRESSURE
 780 WRITE  (IW,80)
     IF(NEV-1) 995,782,783
 782 WRITE
    0   (IW,81) M
     IF(NEW.EQ.1) WRITE(IW,83)
     IF(NEW.EQ.2) WRITE(IW,84)
     GO TO 784
 783 WRITE
    0   (IW,82) M
 784 WRITE(IW,85)
  80 FORMAT(/1H ,10X,37HSURFACE FITTED POTENTIAL AND PRESSURE)
  81 FORMAT(1H+,47X,28H---PHYSICAL PLANE---MODE NO.,I3)
  82 FORMAT(1H+,47X,31H---TRANSFORMED PLANE---MODE NO.,I3)
  83 FORMAT(1H+,78X,23H ---NO THICKNESS EFFECT)
  84 FORMAT(1H+,78X,25H ---WITH THICKNESS EFFECT)
  85 FORMAT(1H ,22X,46H(REAL, IMAGINARY, ABSOLUTE VALUE, PHASE ANGLE)/)
     X1=DH
     DO 850 I=1,L
     JL=ML(NEV,I)
     Y=DH
     IF (JL) 850,850,790
 790 WRITE
    0   (IW,20) I
     DO 840 J=1,JL
     XR=X1-X0(J)
     IF(XR.GE.0.) GO TO 795
     IERROR=795
     WRITE
    0   (IW,90) IERROR,NEW,NEV,I,J,X1,X0(J),XR
     XR=0.5*ABS(XR)
 795 CONTINUE
     XQ=0.5
     IF (YEDG(1)) 800,800,810
 800 XQ=X1
     XR=XR*(X1+X0(J))
 810 XQ=XQ/XR
     XR= SQRT(XR)
     DO 830 N=1,2
```

```
      PSI(N,J)=0.0
      PR(N,J)=0.0
      K=1
      YP=EDG(J)
      DO 830 N1=1,IY
      XP1=XR*YP
      JX=IXB(N1)
      DO 820 M1=1,JX
      PSI(N,J)=PSI(N,J)+C(K,N)*XP1
      PR(N,J)=PR(N,J)+B(K,N)*XP1
      XP1=X1*XP1
  820 K=K+1
  830 YP=Y*Y*YP
      PR(1,J)=2.0*(PR(1,J)+PSI(1,J)*XQ-PSI(2,J)*CK)
      PR(2,J)=2.0*(PR(2,J)+PSI(2,J)*XQ+PSI(1,J)*CK)
      S1=SQRT(PSI(1,J)*PSI(1,J)+PSI(2,J)*PSI(2,J))
      S2=57.29578*ATAN2(PSI(2,J),PSI(1,J))
      S3=SQRT(PR(1,J)*PR(1,J)+PR(2,J)*PR(2,J))
      S4=57.29578*ATAN2(PR(2,J),PR(1,J))
      WRITE(IW,25) J,PSI(1,J),PSI(2,J),S1,S2,J,PR(1,J),PR(2,J),S3,S4
  840 Y=Y+D
  850 X1=X1+D
  860 CONTINUE
      IF(NEV.EQ.1) GO TO 970
C     GET POTENTIAL FROM CORRESPONDING POINT ON TRANSFORMAED WING
      NEV=1
      X1=DH
      DO 950 I=1,L
      JL=ML(1,I)
      Y=DH
      IF(JL) 950,950,870
  870 DO 940 J=1,JL
      YT=Y*EM(J,I)
      JX0=YT/D + 0.5
      IF(JX0.LE.0) JX0=1
      XR=X1-X0(JX0)
      XQ=0.5
      IF(YEDG(1)) 880,880,890
  880 XQ=X1
      XR=XR*(X1+X0(JX0))
  890 CONTINUE
      IF(X1.GT.X0(JX0)) GO TO 900
      IERROR=900
      WRITE
     0   (IW,90) IERROR,NEW,NEV,I,J,X1,X0(JX0),XR
      XR=0.5*ABS(XR)
  900 XQ=XQ/XR
      XR=SQRT(XR)
      DO 930 N=1,2
      PSI(N,J)=0.
      K=1
      YP=EDG(JX0)
      DO 920 N1=1,IY
      XP1=XR*YP
      JX=IXB(N1)
      DO 910 M1=1,JX
      PSI(N,J)=PSI(N,J)+C(K,N)*XP1
      XP1=X1*XP1
  910 K=K+1
```

```
  920 YP=YT*YT*YP
  930 S(N,J,I)=PSI(N,J)/(EM(J,I)+1.E-20)
  940 Y=Y+D
  950 X1=X1+D
      DO 960 I=1,8
      IF(M.GT.1.OR.DA(26).GT.0.) GO TO 958
      XTEMP(I)=XEDG(I)
      YTEMP(I)=YEDG(I)
  958 CONTINUE
      XEDG(I)=XEDGI(I)
  960 YEDG(I)=YEDGI(I)
      GO TO 300
  970 CONTINUE
      IF(NEW.EQ.1) GO TO 975
C     RESTORE PHYSICAL LEADING EDGES (XEDGI $ YEDGI) IN XEDG $ YEDG
C     IF TRANSFORMED WING COMPUTATION FOR ALL MODES IS COMPLETED
C     THIS PART OF RESTORATION HAS BEEN MADE BEFORE COMING HERE
      IF(M.GE.IFIX(DA(28)).AND.IFIX(DA(44)).LT.2) GO TO 978
C     RESTORE TRANSFORMED LEADING EDGES (XTEMP $ YTEMP) IN XEDG $ YEDG
C     TO CONTINUE OTHER MODES COMPUTATION IN TRANSFORMED PLANE
      DO 972 I=1,8
      XEDG(I)=XTEMP(I)
  972 YEDG(I)=YTEMP(I)
      GO TO 978
  975 IF(IFIX(DA(44)).EQ.1) GO TO 978
      IF(M.LT.IFIX(DA(28)).OR.IFIX(DA(26)).EQ.0) GO TO 978
C     RESTORE TRANSFORMED LEADING EDGES (XTEMP $ YTEMP) IN XEDG $ YEDG
C     SINCE PHYSICAL WING COMPUTATION FOR ALL MODES IS COMPLETED
      DO 977 I = 1,8
      XEDG(I)=XTEMP(I)
  977 YEDG(I)=YTEMP(I)
  978 CONTINUE
      JMO=5*(M-1)
      CJ=0.0
      DO 990 I=1,5
      DO 980 J=1,5
      JO=J+JMO
      PS (1,I,JO)=0.0
      PS (2,I,JO)=0.0
      K=1
      DO 980 K2=1,IY
      J2=J+K2-1
      JX=IXB(K2)
      DO 980 K1=1,JX
      J1=K1+I-1
      X1=AY(J2)
      IF(J1.GT.1) X1=X1-CJ*AXY(J1-1,J2)
      Y=AXY(J1,J2)*CK
      PS (1,I,JO)=PS (1,I,JO)+C(K,1)*X1-C(K,2)*Y
      PS (2,I,JO)=PS (2,I,JO)+C(K,1)*Y +C(K,2)*X1
  980 K=K+1
  990 CJ=CJ+1.0
      RETURN
  995 IERROR=995
      WRITE
     O   (IW,90) IERROR,NEW,NEV
      STOP
   20 FORMAT(1H0,5X,I2,6HTH ROW)
   25 FORMAT(1H ,5X,2(2X,I3,1P4E13.5))
   65 FORMAT(1H 2I8,1P2E25.5)
```

```
   70 FORMAT(1H1)
   90 FORMAT(1H0,10X,37HBAD NUMBER IN BOXP NEAR STATEMENT NO.,I5,
     1       /1H ,15X,4I5,1P5E13.5/)
      END
```

```
      SUBROUTINE MRED
      COMMON     A(2,100,50),W(2,50,50),DA(1015),PS(2,5,50),DF(2,5,50)
     1           ,ML(2,50),AXY(9,9),AY(9),XEDGI(8),YEDGI(8)
     2           ,IEDG(2),NS(2),EM(50,50),XEDG(32),YEDG(32),HM(5,5)
     3           ,AREA,D,DI,DH,CK,L,M,NEW,IR,IW
C     FOLLOWING COMMON COMMUNICATES BETWEEN 'MRED' AND 'LSUGA'
      COMMON     XQ(150),YQ(150),DEFQ(150),WTQ(150),COE(5,5)
      CONST=0.28571429
      NP=DA(97)
      IF (NP) 83,24,30
C     A POLYNOMIAL FOR THE PRESSURE/MACH IS FITTED TO VALUES
C     OF PRESSURE/MACH AT GIVEN POINTS.
   30 NX=DA(97)
      NY=DA(97)
   32 IF (100-NP) 83,35,35
   35 CONTINUE
      KP=700
      DO 11 IP=1,NP
      XQ(IP)=DA(KP+1)/DA(24)
      YQ(IP)=DA(KP+2)/DA(24)
      DEFQ(IP)=DA(KP+3)
      WTQ(IP)=1.0
C     DA(96)=1, INPUT DATA ARE PRESSURE COEFFICIENT
C     DA(96)=2, INPUT DATA ARE LOCAL MACH NUMBER
      IF(DA(96)-1.0) 82,105,106
C     CONVERT PRESSURE COEFFICIENT INTO LOCAL MACH NUMBER
  105 DEFQ(IP)=SQRT(5.*(1.2/((1.+0.7*DEFQ(IP))**CONST)-1.))
  106 CONTINUE
   11 KP=KP+3
C     USE LEAST SQUARE METHOD TO CURVE FIT DATA
      IF(IFIX(DA(1011)).NE.0) WRITE(IW,41)
      CALL LSQUA(NP,NX,NY,1011)
      DO 111 J=1,5
      DO 111 I=1,5
  111 HM(I,J)=COE(I,J)
      GO TO 28
C     PRESENTLY INPUT OF PRESSURE COEFFICIENT  IN
C           A POLYNOMIAL FORM IS NOT ALLOWED
   24 CONTINUE
      YP=1.0
      K=1
      DO 27 J=1,5
      XYP=YP
      DO 26 I=1,5
      HM(I,J)=XYP*DA(K+70)
      K=K+1
   26 XYP=XYP*DA(24)
   27 YP=YP*DA(24)**2
   28 CONTINUE
```

```
      RETURN
   82 IPR=96
      GO TO 85
   83 IPR=97
   85 WRITE
    0  (IW,45)IPR
      STOP
   41 FORMAT(1H010X,56HCOMPUTED  MACH(X,Y) = SUM OF  HM(N,M)*X**(N-1)*Y*
     1*(2M-2)/1H010X,54H(IN DIMENSIONLESS COORDINATES - DISTANCE/CHORD L
     2ENGTH)/1H09X,1HN7X,1HM16X,8H HM(N,M))
   45 FORMAT(1H0,10X,14HMRED--BAD DATA,I5)
      END
```

```
      FUNCTION CIN(X1,S)
C     SINE AND COSINE INTEGRAL SUBROUTINE
C
C     IF CALLED BY THE STATEMENT C=CIN(X,S)
C     C AND S ARE THE INTEGRALS OVER T FROM 1 TO INFINITY OF
C         COS(XT)/T  AND  SIN(XT)/T
C
      SG=1.0
      X=X1
      IF (X) 1,2,2
    1 SG=-SG
      X=-X
    2 X2=X*X
      IF (X-1.0) 3,3,4
C     FOR ABS(X) LESS THAN 1 A SERIES EXPANSION IS USED
    3 V=(((X2/98.0-0.6)*.05*X2+1.0)*X2/18.0-1.0)*X+1.57079633
      U=((X2/45.0-1.0)*X2/24.0+1.0)*X2/4.0-.577215665-ALOG(X)
      GO TO 5
C     FOR ABS(X) GREATER THAN 1 APPROXIMATIONS OF HASTINGS ARE USED
    4 P=(((X2+19.394119)*X2+47.411538)*X2+8.493336)/((((X2+21.361055)
     1  *X2+70.376496)*X2+30.038227)*X)
      Q=(((X2+21.383724)*X2+49.719775)*X2+5.089504)/((((X2+27.177958)
     1  *X2+119.918932)*X2+76.707876)*X2)
      CO=COS (X)
      SI=SIN (X)
      U=Q*CO-P*SI
      V=P*CO+Q*SI
    5 S=V*SG
      CIN=U
      RETURN
      END
```

```
      FUNCTION MSIMER(M,N,L,A,B)
      DIMENSION A(M,1),B(M,1)
      ERRR=1.E-14
      DO 30 I = 1,N
      C = 0.0
      DO 10 J = 1,N
   10 C = AMAX1 (C,ABS(A(I,J)))
      IF(C.EQ.0.0) GO TO 1000
      DO 20 J = 1,N
   20 A(I,J) = A(I,J)/C
      DO 30 J = 1,L
   30 B(I,J) = B(I,J)/C
      IF(N.EQ.1) GO TO 205
      NM = N - 1
      DO 200 J = 1,NM
      C = 0.0
      K = 0
      DO 40 I = J,N
      D = ABS(A(I,J))
      IF (C.GE.D) GO TO 40
      K = I
      C = D
   40 CONTINUE
      IF(K.EQ.0.OR.C.LT.ERRR) GO TO 1000
      IF(K.EQ.J) GO TO 70
      DO 50 JJ = J,N
      C = A(J,JJ)
      A(J,JJ) = A(K,JJ)
   50 A(K,JJ) = C
      DO 60 JJ = 1,L
      C = B(J,JJ)
      B(J,JJ) = B(K,JJ)
   60 B(K,JJ) = C
   70 C = A(J,J)
      JP = J + 1
      DO 80 JJ = JP,N
   80 A(J,JJ) = A(J,JJ)/C
   90 DO 100 JJ = 1,L
  100 B(J,JJ) = B(J,JJ)/C
      DO 200 I = 1,N
      IF(I.EQ.J) GO TO 200
      C = A(I,J)
      DO 110 JJ = JP,N
  110 A(I,JJ) = A(I,JJ) - C*A(J,JJ)
      DO 120 JJ = 1,L
  120 B(I,JJ) = B(I,JJ) - C*B(J,JJ)
  200 CONTINUE
  205 C = A(N,N)
      IF(ABS(C).LT.ERRR) GO TO 1000
      DO 210 J = 1,L
  210 B(N,J) = B(N,J)/C
      IF(N.EQ.1) GO TO 230
      DO 220 I = 1,NM
      C = A(I,N)
      DO 220 JJ = 1,L
  220 B(I,JJ) = B(I,JJ) - C*B(N,JJ)
  230 MSIMER = 1
      RETURN
```

```
1000 MSIMER = 2
     RETURN
     END



     FUNCTION MSIMEC(M,N,L,A,B)
     DIMENSION A(M,1),B(M,1)
     COMPLEX A,B,G
     ERRR=1.E-14
     DO 30 I = 1,N
     C = 0.0
     DO 10 J = 1,N
  10 C=AMAX1(C,ABS(REAL(A(I,J))),ABS(AIMAG(A(I,J))))
     IF(C.EQ.0.0) GO TO 1000
     DO 20 J = 1,N
  20 A(I,J) = A(I,J)/C
     DO 30 J = 1,L
  30 B(I,J) = B(I,J)/C
     IF(N.EQ.1) GO TO 205
     NM = N - 1
     DO 200 J = 1,NM
     C = 0.0
     K = 0
     DO 40 I = J,N
     D=ABS(REAL(A(I,J)))+ABS(AIMAG(A(I,J)))
     IF(C.GE.D) GO TO 40
     K = I
     C = D
  40 CONTINUE
     IF(K.EQ.0.OR.C.LT.ERRR) GO TO 1000
     IF(K.EQ.J) GO TO 70
     DO 50 JJ = J,N
     G = A(J,JJ)
     A(J,JJ) = A(K,JJ)
  50 A(K,JJ) = G
     DO 60 JJ = 1,L
     G = B(J,JJ)
     B(J,JJ) = B(K,JJ)
  60 B(K,JJ) = G
  70 G = 1.0/A(J,J)
     JP = J + 1
     DO 80 JJ = JP,N
  80 A(J,JJ) = A(J,JJ)*G
  90 DO 100 JJ = 1,L
 100 B(J,JJ) = B(J,JJ)*G
     DO 200 I = 1,N
     IF(I.EQ.J) GO TO 200
     G = A(I,J)
     DO 110 JJ = JP,N
 110 A(I,JJ) = A(I,JJ) - G*A(J+JJ)
     DO 120 JJ = 1,L
 120 B(I,JJ) = B(I,JJ) - G*B(J,JJ)
 200 CONTINUE
 205 G = A(N,N)
     IF (ABS(REAL(G)) + ABS(AIMAG(G)).LT.ERRR) GO TO 1000
     DO 210 J = 1,L
 210 B(N,J) = B(N,J)/G
```

```
      IF(N.EQ.1) GO TO 230
      DO 220 I = 1,NM
      DO 220 JJ = 1,L
  220 B(I,JJ) = B(I,JJ) - A(I,N)*B(N,JJ)
  230 MSIMEC = 1
      RETURN
 1000 MSIMEC = 2
      RETURN
      END
```