NASA CR-158907

SONIC-BOX METHOD EMPLOYING LOCAL MACH NUMBER FOR OSCILLATING WINGS WITH THICKNESS

(NASA-CR-158907) SONIC-BOX METHOD EMPLOYING LOCAL MACH NUMBER FOR OSCILLATING WINGS WITH THICKNESS Final Report (Lockheed-Georgia Co., Marietta.) 73 p HC A04/MF A01 CSCL 01A
N79-10999
Unclas
G3/02 36971

By S. Y. Ruo

September 1978

Prepared under Contract No. NAS1-13613 by
LOCKHEED-GEORGIA COMPANY
Marietta, Georgia

for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

CONTENTS

SONIC-BOX METHOD EMPLOYING LOCAL MACH NUMBER FOR OSCILLATING WINGS WITH THICKNESS

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 has been extended to include the effects of the swept trailing edge and the thickness of the wing. Account for the non-uniform flow caused by finite thickness is made by application of the local linearization concept. The thickness effect, expressed in terms of the local Mach number, is included in the basic solution to replace the coordinate transformation method used in the earlier work. Calculations were made for a delta wing and a rectangular wing performing plunge and pitch oscillations, and the results were compared with those obtained from other methods. An input guide and a complete listing of the computer code are presented.

INTRODUCTION

In reference 1, the sonic-box method computer program was developed for calculation of unsteady transonic flow aerodynamics for oscillating planar wings with unswept trailing edge by approximating the wing planform with a matrix of square boxes. Later, it was extended to include the swept trailing edge and control surfaces in reference 2. The sonic-box method uses a doublet velocity potential as the basic solution to satisfy the linearized transonic flow, unsteady small-perturbation velocity-potential equation with the associated boundary conditions.

In references 3 and 4, the wing thickness effect is partially recovered by the inclusion of local Mach number in the governing equation for the unsteady transonic flow. It uses the concept of local linearization to reduce the nonlinear small-perturbation equation to a linear one with non-constant coefficients. This is further reduced to a linear equation with constant coefficients by an appropriate coordinate transformation. This final equation and the associated boundary condition in the transformed space become identical to those treated in the physical space by Rodemich and Andrew in reference 1. The numerical results for the wing with thickness were obtained by adopting the sonic-box method in the transformed space. Because of the assumptions made in deducing the governing equation to a manageable form, this technique is applicable only to relatively thin wings. That is, the local mean Mach number on the wing surface must not be very different from unity. Further, it is assumed that there is no flow separation and no strong shock waves on the wing surface.

The computer programs developed in references 1, 2, and 3 use the least-square method to fit some of the input data, such as wing deflection or steady Mach number distribution on the wing, and to fit the computed velocity potential with a form of pre-determined polynomial surface for the subsequent calculation of the unsteady pressure and the generalized aerodynamic force coefficients. The computer program described in reference 4 adopts the natural cubic spline for fitting calculated velocity potential and the spline-surface for fitting input modal deflections and Mach number distribution instead of the polynomial-surface fitting used in references 1, 2, and 3. The codes in references 3 and 4 allow the computation of generalized aerodynamic force coefficients for wings of zero and finite thickness; the swept trailing edges are allowed but not the control surfaces.

The computer program presented in this report is developed according to the "alternate technique" described in reference 5. The coordinate transformation technique as used in references 3 and 4 fails when the mean local Mach number on the wing becomes very different from unity. In order to avoid this problem, an alternate technique was proposed in reference 5 to approximately account for the thickness effect, expressed in terms of mean local Mach number on the wing, by including it directly in the basic doublet solution to replace the coordinate transformation. The computer program thus developed is smaller than that of reference 4 and the amount of computation required has also been reduced.

In general, the basic assumptions and limitations applied to the computer code in reference 4 also apply to the present code. However, the present formulation avoids the difficulties associated with the artificial wake and wing-surface fold-over due to multivalued transformation which limits the usefulness of the coordinate transformation formulation of reference 4. The zero thickness wing portion of the computer code is unchanged from that of reference 4. Input is identical in both codes and the output differs very little between them.

SYMBOLS

b	reference length (dimension = L)
C_p	pressure coefficient
exp, e	exponential function
i	$\sqrt{-1}$
k	reduced frequency, $\omega b/U_\infty$
L	unit of length
L_{ij}	generalized aerodynamic force coefficient
M	local Mach number
T	unit of time
U_∞	reference velocity (freestream), (dimension = L/T)
x,y,z	dimensionless Cartesian coordinates (reference length = b)
θ_{ij}	phase angle of L_{ij}
τ	maximum thickness to root chord ratio
φ_o	magnitude of oscillatory dimensionless small perturbation velocity potential
ω	angular velocity (dimension = radian/T)
$(\ )_{\ell e}$	subscripts denote quantity at leading edge
$(\ )_{te}$	subscripts denote quantity at trailing edge

METHOD

The computer program described in reference 4 is based on the coordinate transformation technique to reduce the locally linearized equation with nonconstant coefficients for nonzero thickness wing at sonic speed to a linear one with constant coefficients. This linear equation with the associated boundary conditions can, then, be solved with sonic-box method. When the mean local Mach number on the wing becomes very different from that of the free-stream, the transformation may become multivalued and consequently an artificial wake or wing-surface fold-over may be created in the transformed space. This technique fails once it happens. In order to avoid this problem, an alternate technique was proposed in reference 5 to approximately account for the thickness effect, expressed in terms of mean local Mach number on the wing, by including it in the basic doublet solution.

The governing equation for unsteady transonic small perturbation velocity potential is

$$\varphi_{o_{yy}} + \varphi_{o_{zz}} - M^2(2ik\varphi_{o_x} - k^2\varphi_o) = 0 , \tag{1}$$

where

$$\varphi_o(x,y,z) = \varphi(x,y,z,t)\cdot e^{-ikt} ,$$

which is also equation (1) of reference 4.

The basic solution for equation (1), representing a point doublet oriented parallel to the z-axis at the origin and satisfying the required condition at infinity for a small finite region on the wing where the value of M, the Mach number, is considered to be constant, may be written as

$$\varphi_o = \begin{cases} 0 , & x \le 0 \\ \dfrac{ik}{2\pi}\dfrac{zM^2}{x^2} \exp\{-\tfrac{1}{2}ik[x + \dfrac{M^2(y^2+z^2)}{x}]\} , & x>o, \end{cases} \tag{2}$$

in which M is regarded as a parameter. This solution satisfies equation (1) only in a small finite region of the wing; so the solution may be considered to be of the locally linearized form.

The only quantity in the program of reference 4 requiring modification is the velocity influence coefficient for the wings with thickness. It is presently written as

$$A = \frac{ik}{2\pi} M^2 \iint\limits_E \frac{1}{(x-\xi)^2} \exp \{ - \frac{ik}{2} [(x-\xi) + \frac{M^2(y-\eta)^2}{(x-\xi)}]\} \, d\xi d\eta \tag{3}$$

$$= \frac{ik}{2\pi} M \iint \frac{1}{u^2} \exp [- \frac{i\ell}{2}(u + \frac{v^2}{u})] \, dudv \tag{4}$$

where

H = length of the box side
k = reduced frequency
M = mean local Mach number
$i = \sqrt{-1}$
$\ell = kH$
E = box at (ξ,η)
$u = (x-\xi)/H$
$v = M(y-\eta)/H$

The value of the velocity influence coefficient computed in the sonic-box computer program is with $M = 1.0$ in equations (3) and (4). Under this condition, the velocity influence coefficient is function of the wing geometry only. For $M \neq 1.0$, it becomes function of the Mach number also. The value of the modified velocity influence coefficient required in this alternate technique to account for the wing thickness effect may be evaluated from the table computed for $M = 1.0$ condition for the same reduced frequency.

To evaluate the modified velocity influence coefficient for this alternate technique, one may do the following:

1. take the average value of the mean local Mach number at the center of the receiving, (x,y), and sending, (ξ,η), boxes,

2. multiply the spanwise distance between these two box centers by the value of the average Mach number,
3. interpolate the modified velocity influence coefficient from the original table for $M = 1.0$ with the value of the modified spanwise distance, v,
4. multiply this value by the mean local Mach number at the center of the sending box.

The rest of the computation remains practically unchanged except that the computation in the transformed space is totally eliminated.

COMMENTS ON THE PROGRAM

The velocity potential influence coefficients for a wing of zero thickness at a given frequency are only a function of the geometry. However, in addition to the geometry, they are also a function of the local Mach number distribution for the nonzero thickness wing under present formulation. It may be possible to perform the integration in equation (3) analytically with a new formula or with that already in the earlier program with some approximation. No attempt was made to derive the totally new formulation. One of the approximate methods which was studied but not implemented in the present program is to substitute the local Mach number, M, in the integrand of equation (3) with $(1-\varepsilon)$, where ε is a positive or a negative small number. After expanding the expotential function involving ε term and neglecting all ε^2 or higher terms, one obtains an approximate form of the integrand, for a doublet at the origin and $z = 0$, as follows:

$$\left(\frac{1}{x^2} + i\varepsilon k \frac{y^2}{x^3}\right) \cdot \exp\left[-\frac{ik}{2}\left(x + \frac{y^2}{x}\right)\right]. \qquad (5)$$

The exponential function in equation (5) is the same as that used in the case for $M = 1.0$ and the routines in the earlier sonic-box computer program may be utilized to perform the integration. Due to its complexity, and the additional computer storage and time required, this approximate method was not adopted to generate the new velocity influence coefficient matrix with the Mach number effect. Instead, it is interpolated from the velocity influence coefficient matrix for the zero thickness wing as described in the preceding section. The Mach number appearing in the exponential function in equation (3) is only associated with the distance between receiving and sending points and it is regarded as to modify the effective distance between these two points. Therefore, the average Mach number is used to maintain its interchangeability. Another Mach number in equation (3) is regarded as to modify the doublet strength. Since the integration is performed over the surface of the sending box, it is logical to use the Mach number at that point. This simplification in coupling the Mach number effect enables a reduction of the size of the computer code and the computation time. The computed results appeared to be reasonable under the assumptions of small perturbation theory and local linearization concepts.

In the present formulation, it implies as in the coordinate transformation formulation (ref. 4) that the Mach number variation in the spanwise direction is not large. The accuracy of these methods decreases when a large Mach number variation in spanwise direction exists. The present method, however, does not fail abruptly as does the coordinate transformation method when spanwise variation of Mach number becomes large enough to cause multivalued transformation and hence fold-over of wing-surface.

Since no smoothing has been applied on either the input data or any computed values in the data fitting process during the computation, the calculated unsteady pressure coefficient distribution may not be smooth and should be used with caution. In order to use it, the computed pressure coefficient should be put through a smoothing process such as the smoothing portion of the two-dimensional cubic-spline fitting routines in the present program. The pressure coefficient is obtained by differentiation of a set of numerical values whereas the generalized aerodynamic force coefficient is obtained by integration. Since integration itself is a smoothing process, the resulting generalized aerodynamic force coefficient is considered to be acceptable within the bounds of the accuracy of the numerical techniques and the adequacy of the sonic-box method. The option of data smoothing is not provided in the three-dimensional spline-surface fitting process used in the present code for input data such as wing deflections and mean local Mach number. The spline-surface is required to go through all input points.

RESULTS

Sample calculations are made for a delta wing and a rectangular wing oscillating in plunge (Mode 1) and in pitch about the apex (Mode 2). The mean angle of attack is zero and the freestream is at sonic speed.

Delta Wing

The delta wing considered here is a flattened elliptic cone of aspect ratio 1.5 and thickness-to-root-chord ratio $\tau = 0.1$. Convergence with respect to the number of boxes along the root chord for the generalized force coefficients (L_{ij}) due to plunge and pitch about the apex, at a reduced frequency of $k = 0.2$, is shown in figure 2. The maximum numerical difference within the applied range of 15 and 30 boxes along the root chord is about 4 percent, and the trend of convergence with and without thickness is essentially the same. Based on the results shown in figure 2, it appears that the gain in convergence by using a large number of boxes to represent the wing is not obvious as compared with a fortuitous selection of the number of boxes to use. The numerical fluctuation in the convergence plot is largely caused by the box arrangement along the wing leading edge which, in turn, is dependent on the number of boxes selected for use along the wing root chord. Contribution from the partial boxes along the leading edge has been taken into account, but the fluctuation still exists.

The variation of each force coefficient, using 30 boxes along the root chord, versus the reduced frequency is plotted in figure 3. The results from figure 7 of reference 4 are also shown. The numerical difference between the results for wings with and without thickness is very small, generally less than one percent. This is a result of the Mach number, at each box-center used in the computation, lying within the narrow range of 0.92 and 0.98 in chordwise direction and remaining constant in spanwise direction (see fig. 8

of ref. 4). However, the thickness effect on flutter speed can be significant (ref. 6). The results for the case with thickness obtained from the present method and that of reference 4 are not very different.

Rectangular Wing

The rectangular wing considered here has aspect ratio 2.0 and a biconvex (circular arc) airfoil with thickness-to-chord ratio $\tau = 0.0521$. The variation of each force coefficient, using 20 boxes along the root chord, versus the reduced frequency is plotted in figure 4. The results obtained from the present method, and from references 4 and 7, are included in the figure. The thickness effect on the rectangular wing is seen to be slightly larger than that on the delta wing. This is probably caused by the wide range of Mach number variation (fig. 6) on the rectangular wing, even though the thickness ratio is only 0.0521 for the rectangular wing against 0.1 for the delta wing.

The present method predicts values higher than either Landahl's results (ref. 7) for the zero thickness case or the results of reference 4 for the nonzero thickness case. The difference of the generalized aerodynamic force coefficients for the nonzero thickness case between the results obtained from the present method and that of reference 4 is quite large. This might be caused by the difference in interpretation of the effective distance between the sending and the receiving points in the present method and the coordinate transformation method used in reference 4. It is felt, however, that the interpretation used in the present program is more physically sound than that used in reference 4. The phase angle predicted by the sonic-box method at very low reduced frequency becomes meaningless when the magnitude of any force coefficient approaches to zero with decreasing reduced frequency (for example, see figs. 4(b) and 4(d)). This is due to numerical inaccuracy and not to any inadequacy of the method.

The steady-state pressure coefficient obtained from reference 8 for the rectangular wing considered here is shown in figure 5(a) for the chordwise (x-direction) distribution and in figure 5(b) for the spanwise (y-direction) distribution. The corresponding Mach numbers at the box-centers, interpolated from the fitted spline-surface, are plotted in figures 6(a) and 6(b). The interpolated values deviate from the input data more near the leading edge than near the trailing edge. This probably was caused by the use of more dense spacing of input points near the leading edge as compared with those near the trailing edge in chordwise direction and by the lack of input points near the leading edge in spanwise direction, especially in the inboard portion of the wing. A better fit than that shown in figures 6(a) and 6(b) may be obtained by using more evenly spaced input points than those shown in figures 5(a) and 5(b).

CONCLUDING REMARKS

A sonic-box method computer program is presented for the application of a local linearization concept capable of accounting approximately for wing thickness effects in unsteady sonic flow. The thickness effect, expressed in terms of the local steady Mach number, is directly included in the basic solution. The local doublet strength is adjusted from the sonic flow condition to that for the local flow, and the governing equation is reduced to the one used in the original sonic-box method for zero thickness wings. Thus, the original sonic-box method concept can be used directly to treat nonzero thickness wings.

Convergence of the numerical results with respect to the number of boxes used in representing the wing planform seemed to depend more on the arrangement of the boxes along a swept leading edge than on the total number of boxes used, even though the partial boxes along the leading edge were included in the computation. For a wing with unswept leading edge, the use of a small number of boxes (say, 15 to 20 along the root chord) appeared to be sufficient to obtain results that were essentially converged.

When the input data require spline-surface fitting, the input points must be selected in such a way that they are as uniformly spaced as possible to avoid locally-concentrated large errors. A smoothing option for the two-dimensional cubic-spline has been included in the present program, but it was not utilized in the sample runs shown in this report. Since the box method itself is numerical in nature, the distribution of calculated values may not always be smooth; and it may become necessary to perform the smoothing before any gradients are evaluated.

Based on the sample runs made, the contribution due to thickness was not found to be very large in comparison with the results calculated from the coordinate transformation method. Due to the lack of reliable experimental data, it is rather difficult to assess the validity of the present approach in accounting for thickness effects.

REFERENCES

1. 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.

2. Stenton, T. E. and Andrew, L. V.: Transonic Unsteady Aerodynamics for Planar Wings with Trailing Edge Control Surfaces. AFFDL-TR-67-180, August 1968, Air Force Flight Dynamics Lab., Wright-Patterson Air Force Base, Ohio.

3. Ruo, S. Y.: Calculation of Unsteady Transonic Aerodynamics for Oscillating Wings with Thickness (Computer Program). NASA CR-132477, Sept. 1974.

4. Ruo, S. Y.: Improved Sonic-Box Computer Program for Calculating Transonic Aerodynamic Loads on Oscillating Wings with Thickness. NASA CR-158906, Sept. 1978.

5. Ruo, S. Y. and Theisen, J. G.: Calculation of Unsteady Transonic Aerodynamics for Oscillating Wings with Thickness. NASA CR-2259, June 1975.

6. Ruo, S. Y.; Yates, E. C., Jr.; and Theisen, J. G.: Calculation of Unsteady Transonic Aerodynamics for Oscillating Wings with Thickness. *AIAA Journal of Aircraft*, Vol. 11, No. 10, Oct. 1974, pp. 601-608.

7. Landahl, M. T.: *Unsteady Transonic Flow.* Pergamon Press, New York, 1961.

8. 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.

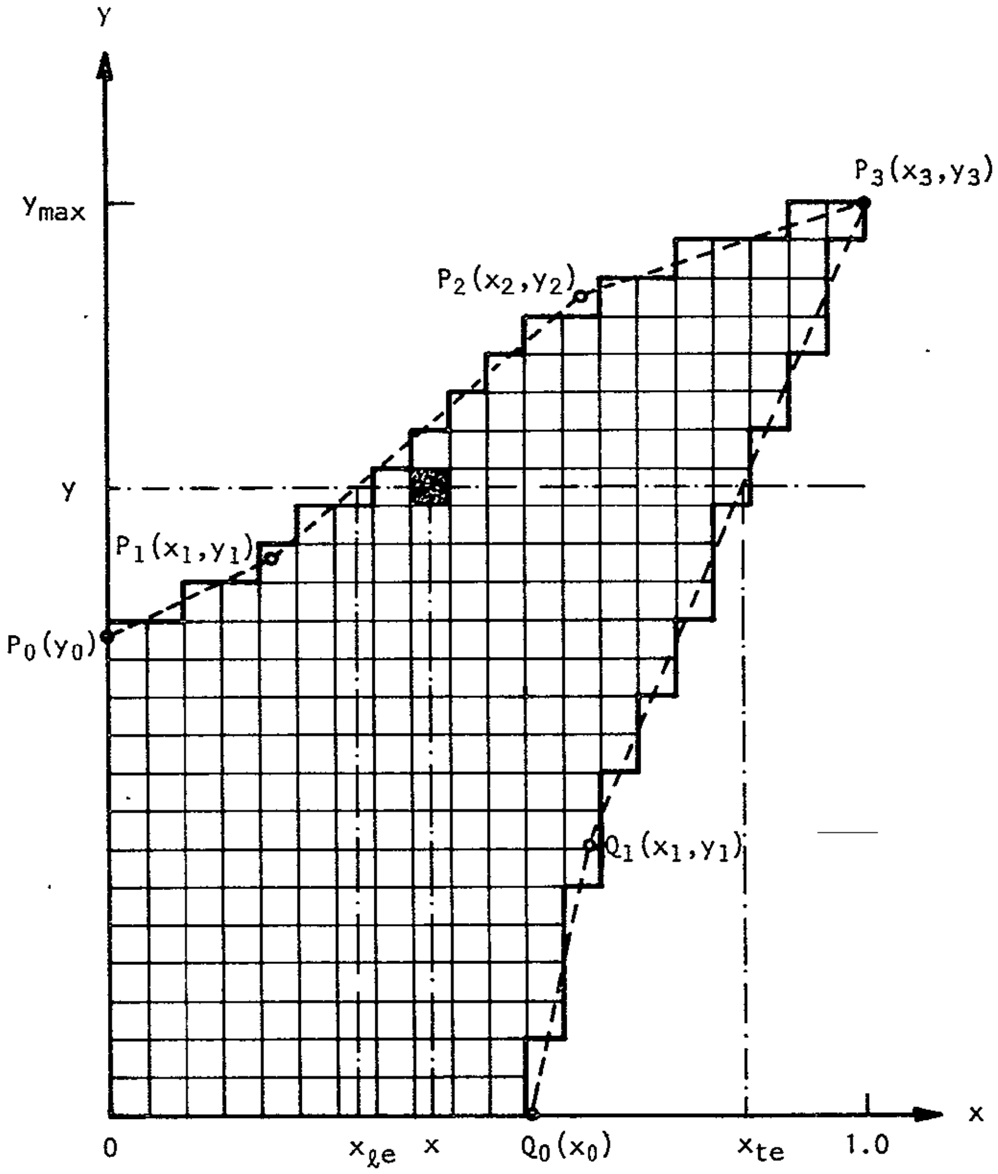
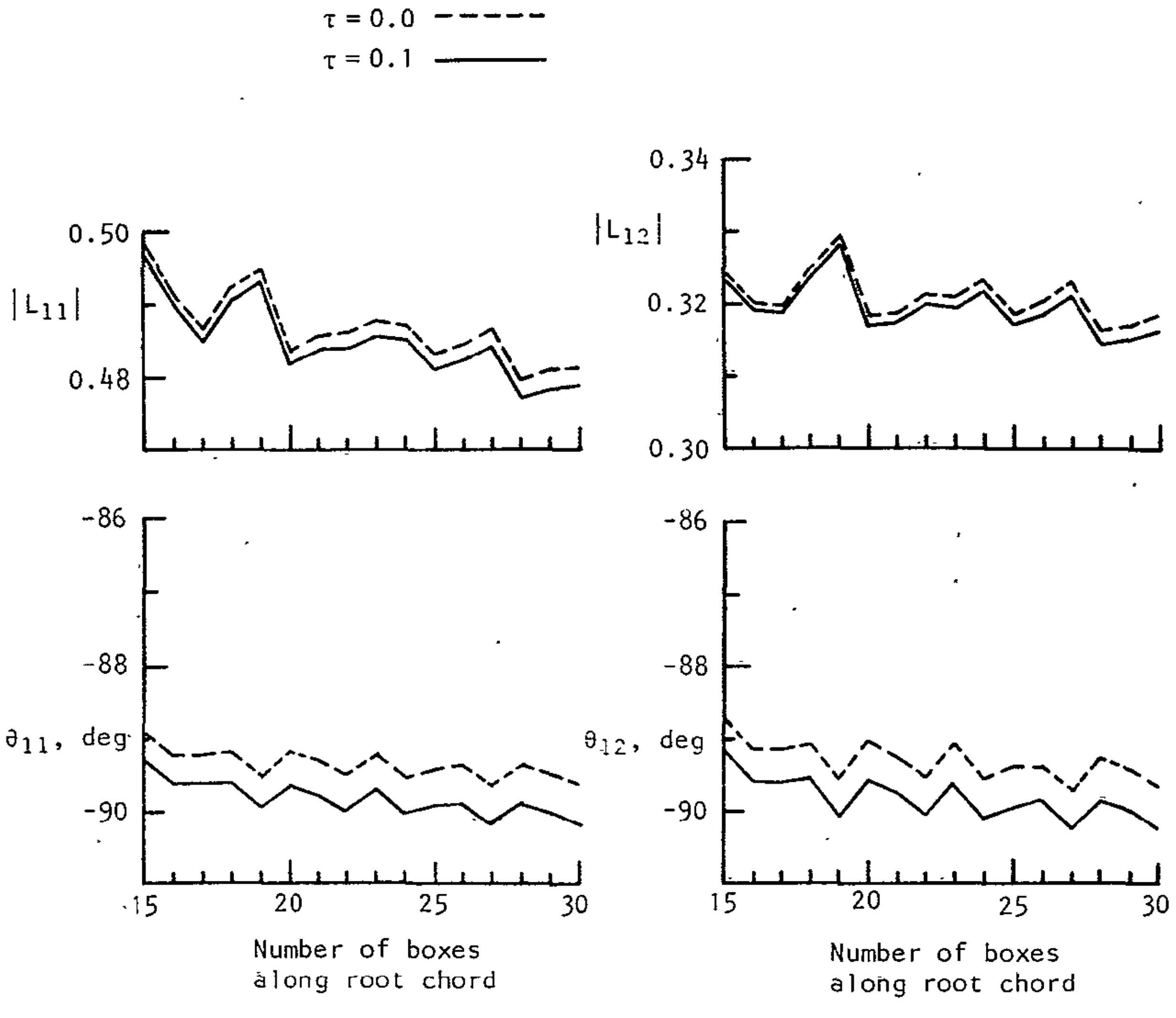


Figure 1. - Half wing geometry.

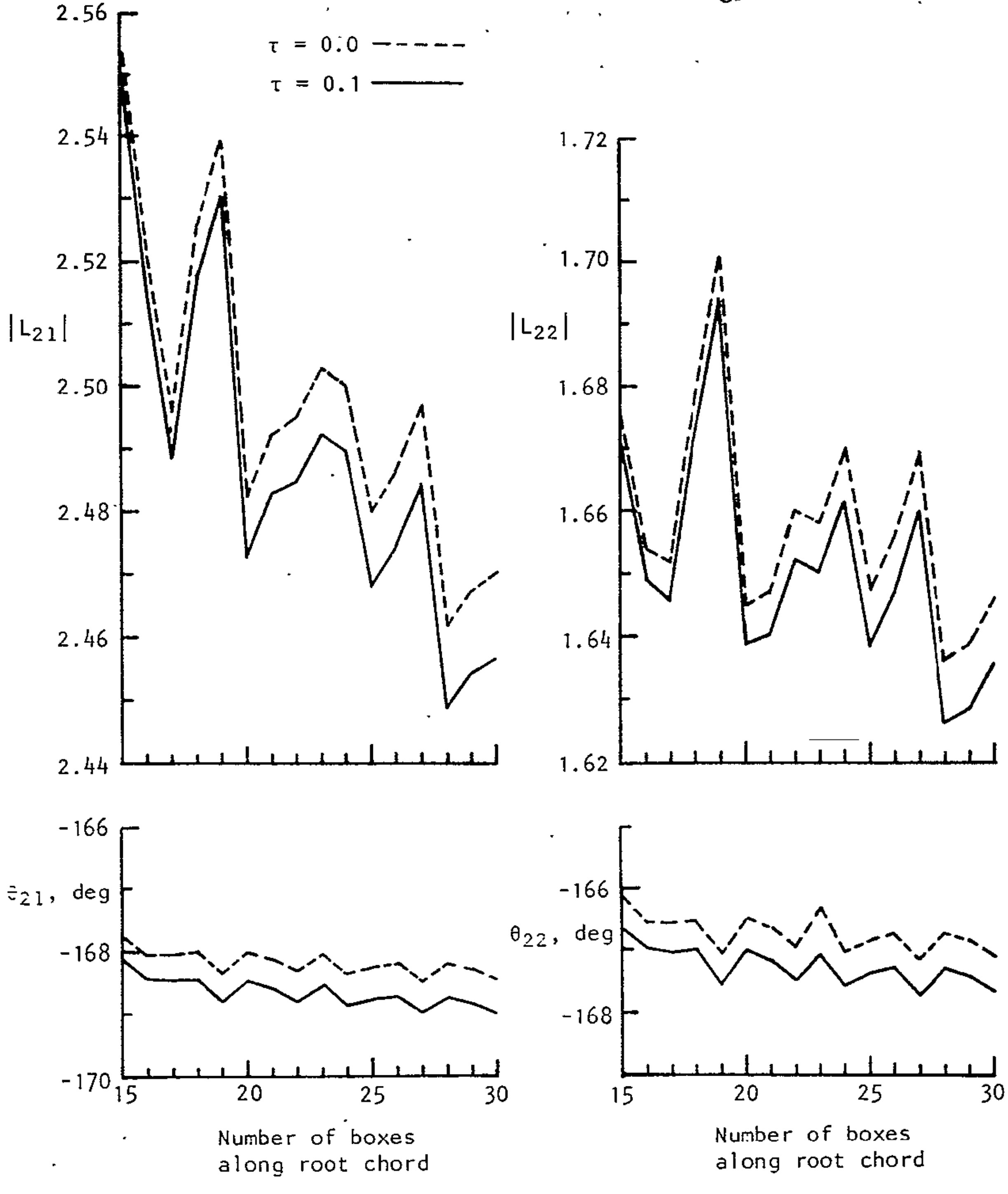


(a) Lift due to plunge.

(b) Moment due to plunge.

Figure 2. - Convergence of force coefficients due to plunge and pitch for delta wing of aspect ratio 1.5 at reduced frequency 0.2.

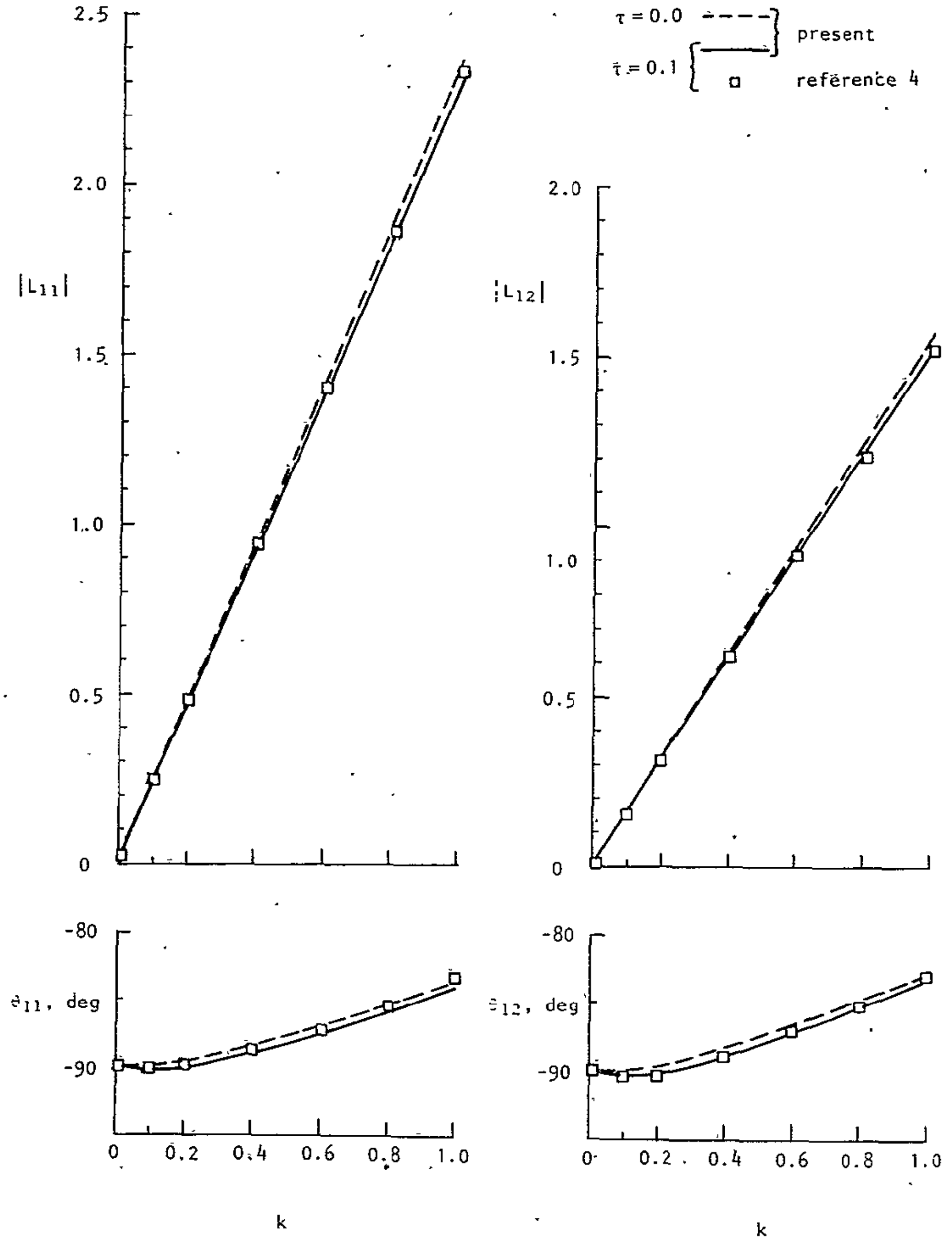
ORIGINAL PAGE IS
OF POOR QUALITY



(c) Lift due to pitch.

(d) Moment due to pitch.

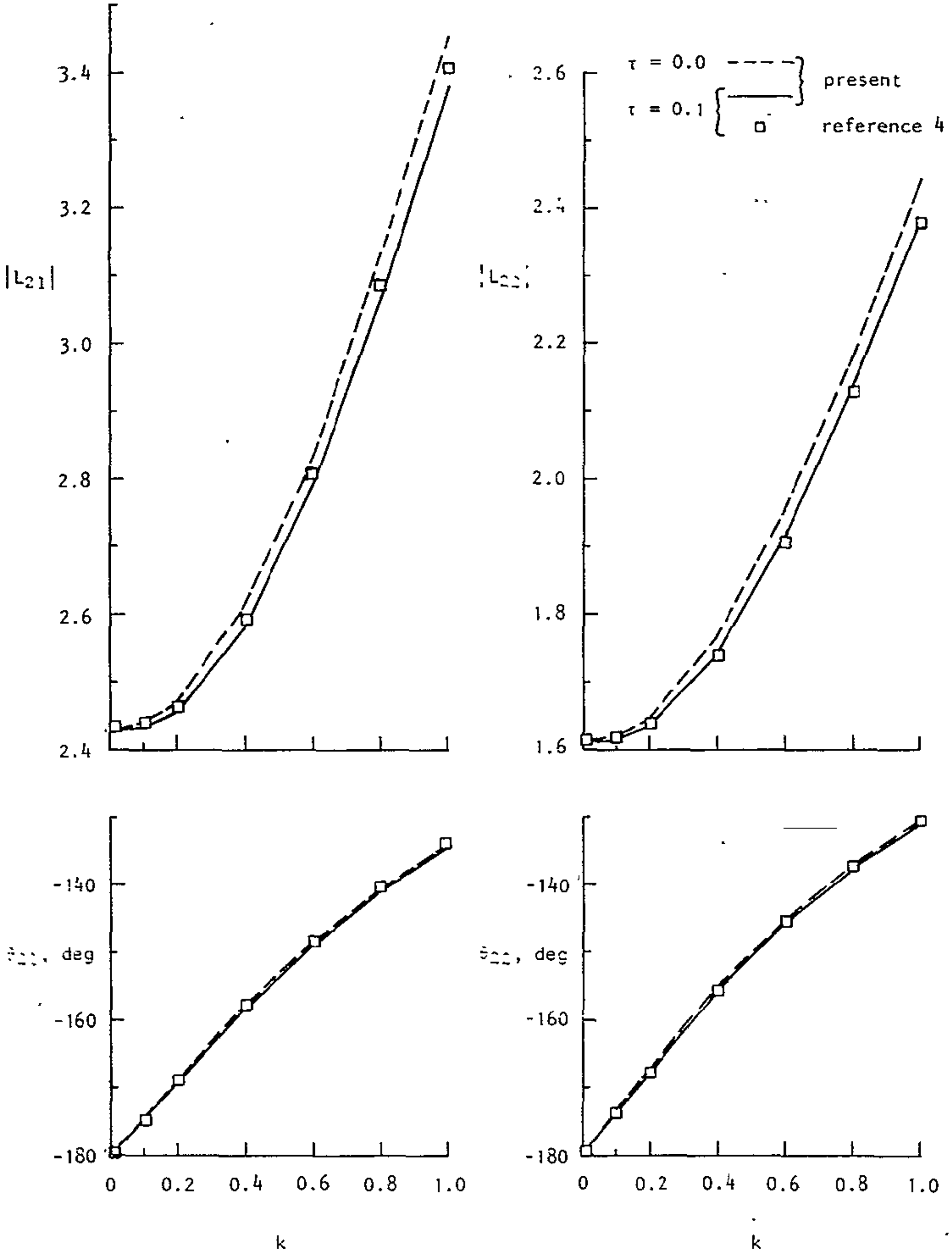
Figure 2. - Concluded.



(a) Lift due to plunge.

(b) Moment due to plunge.

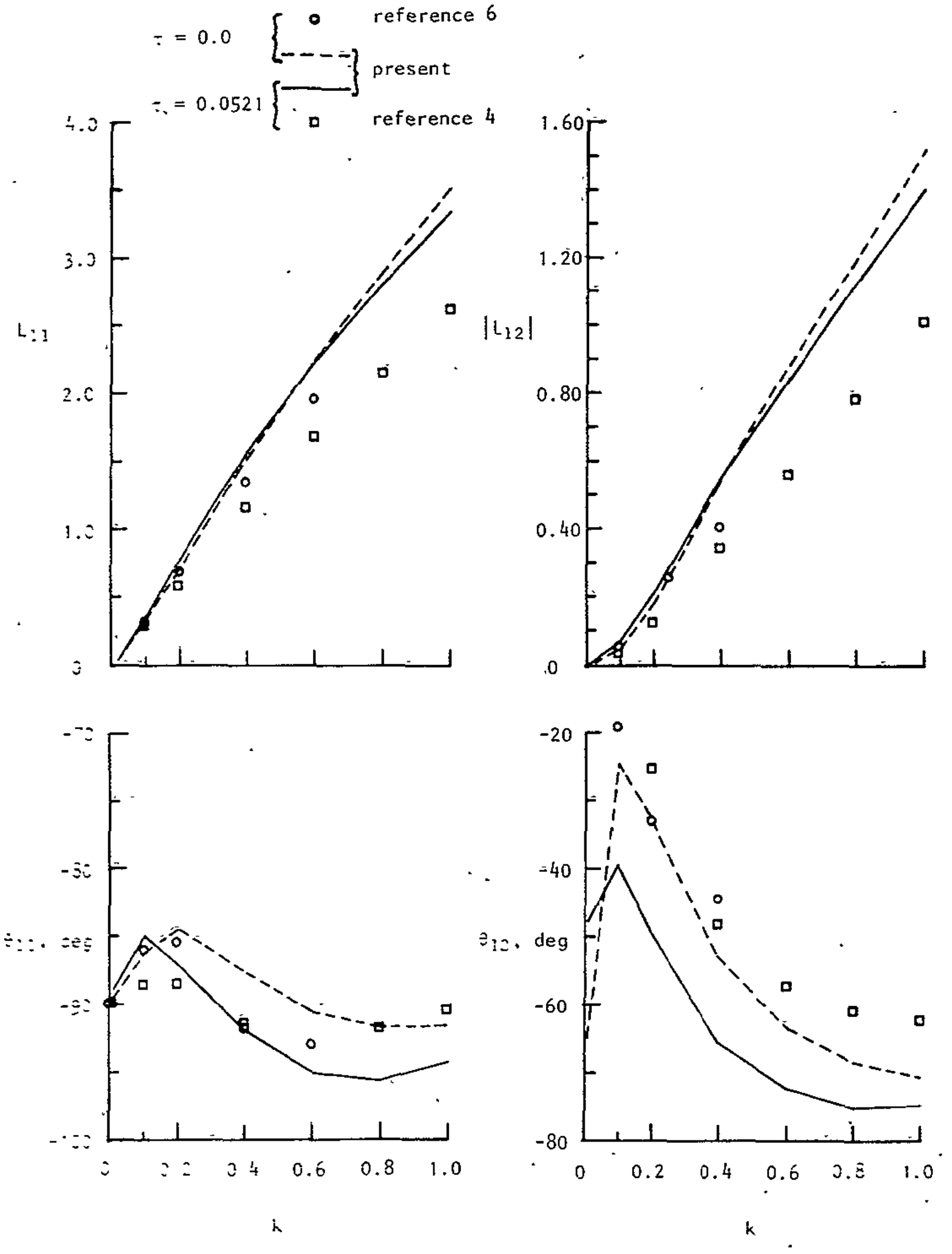
Figure 3. - Force coefficients due to plunge and pitch for delta wing of aspect ratio 1.5 with 30 boxes along root chord.



(c) Lift due to pitch.

(d) Moment due to pitch.

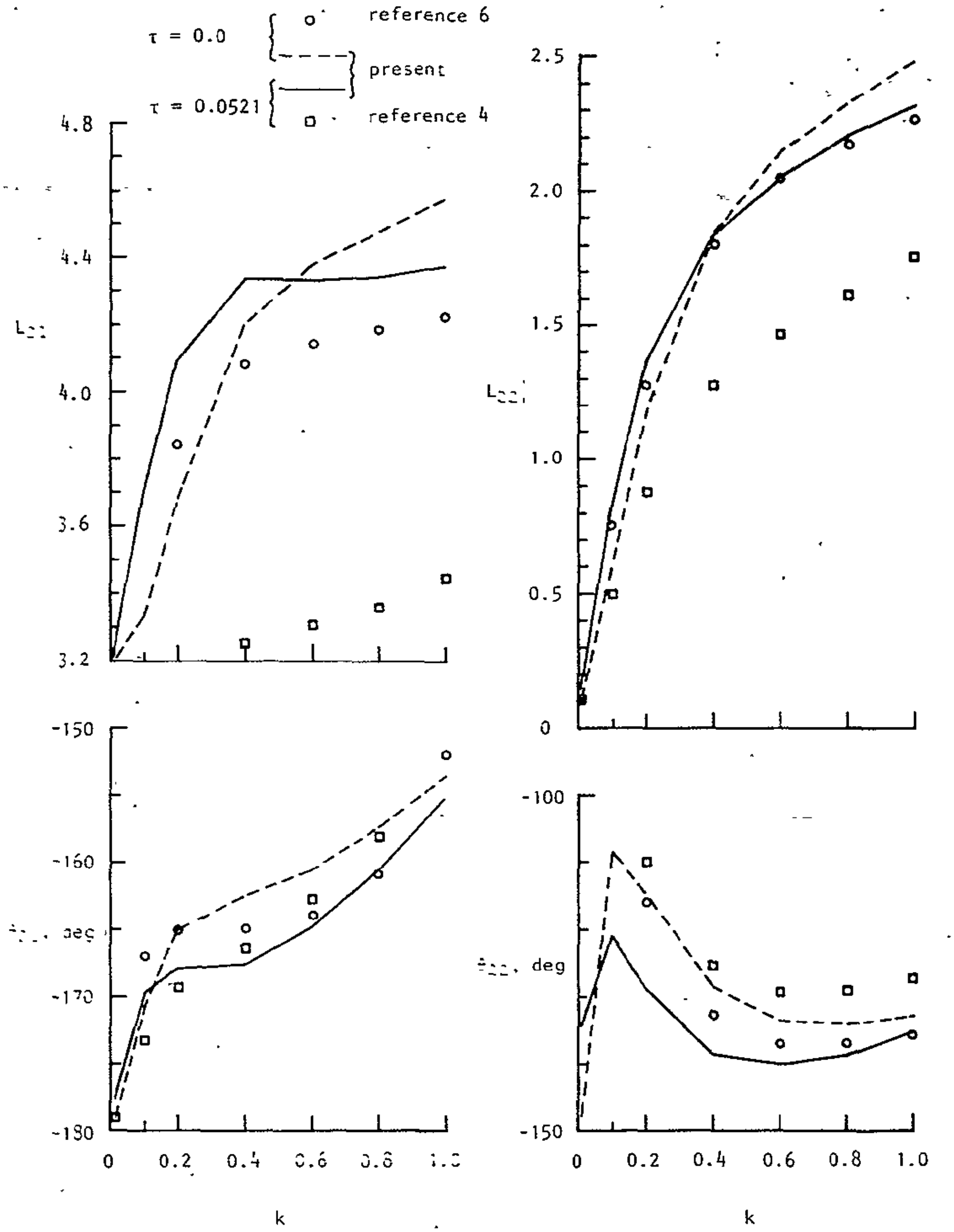
Figure 3. - Concluded.



(a) Lift due to plunge.

(b) Moment due to plunge.

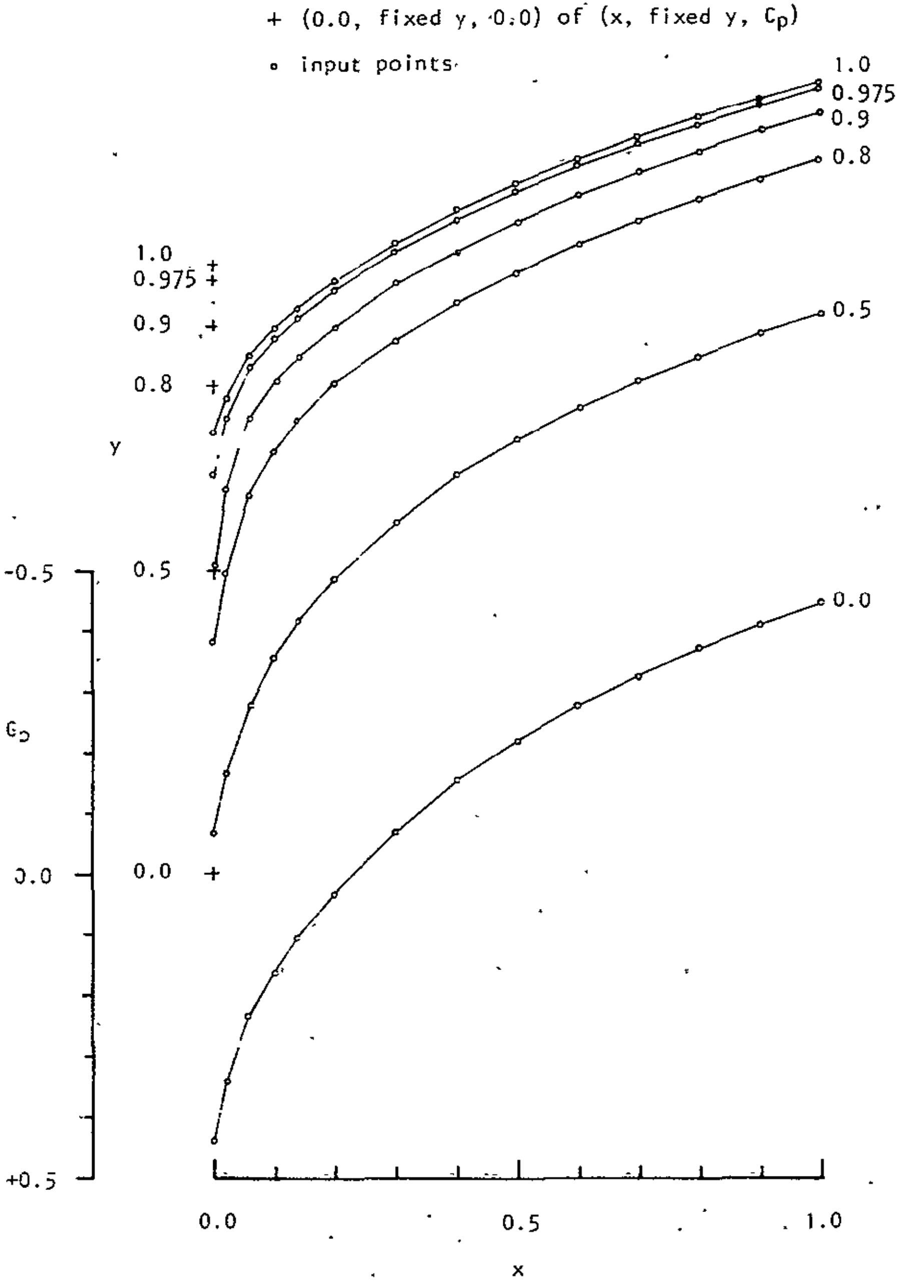
Figure 4. - Force coefficients due to plunge and pitch for rectangular wing of aspect ratio 2.0 with 20 boxes along root chord.



(c) Lift due to pitch.

(d) Moment due to pitch.

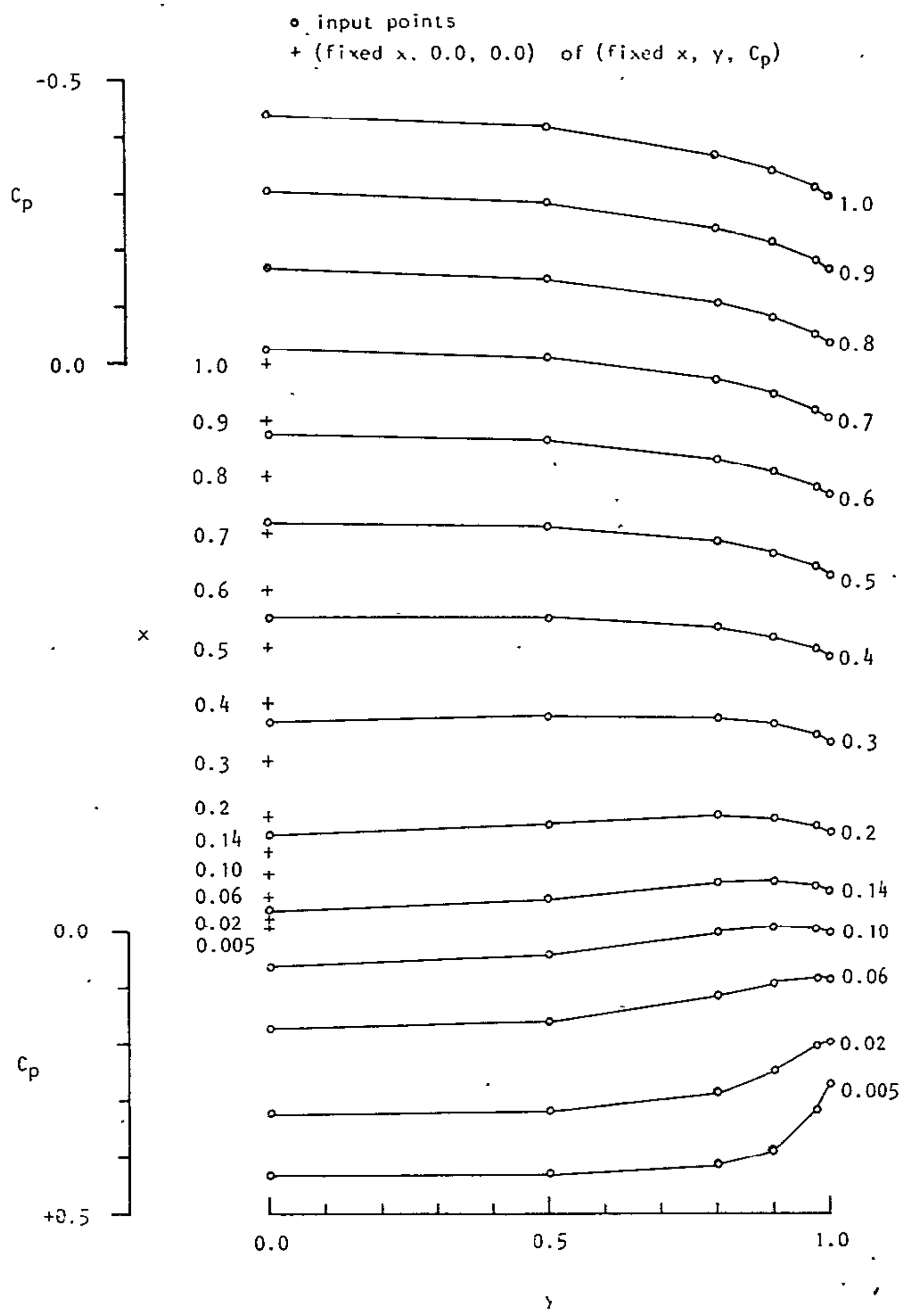
Figure 4. - Concluded.



(a) Chordwise distribution.

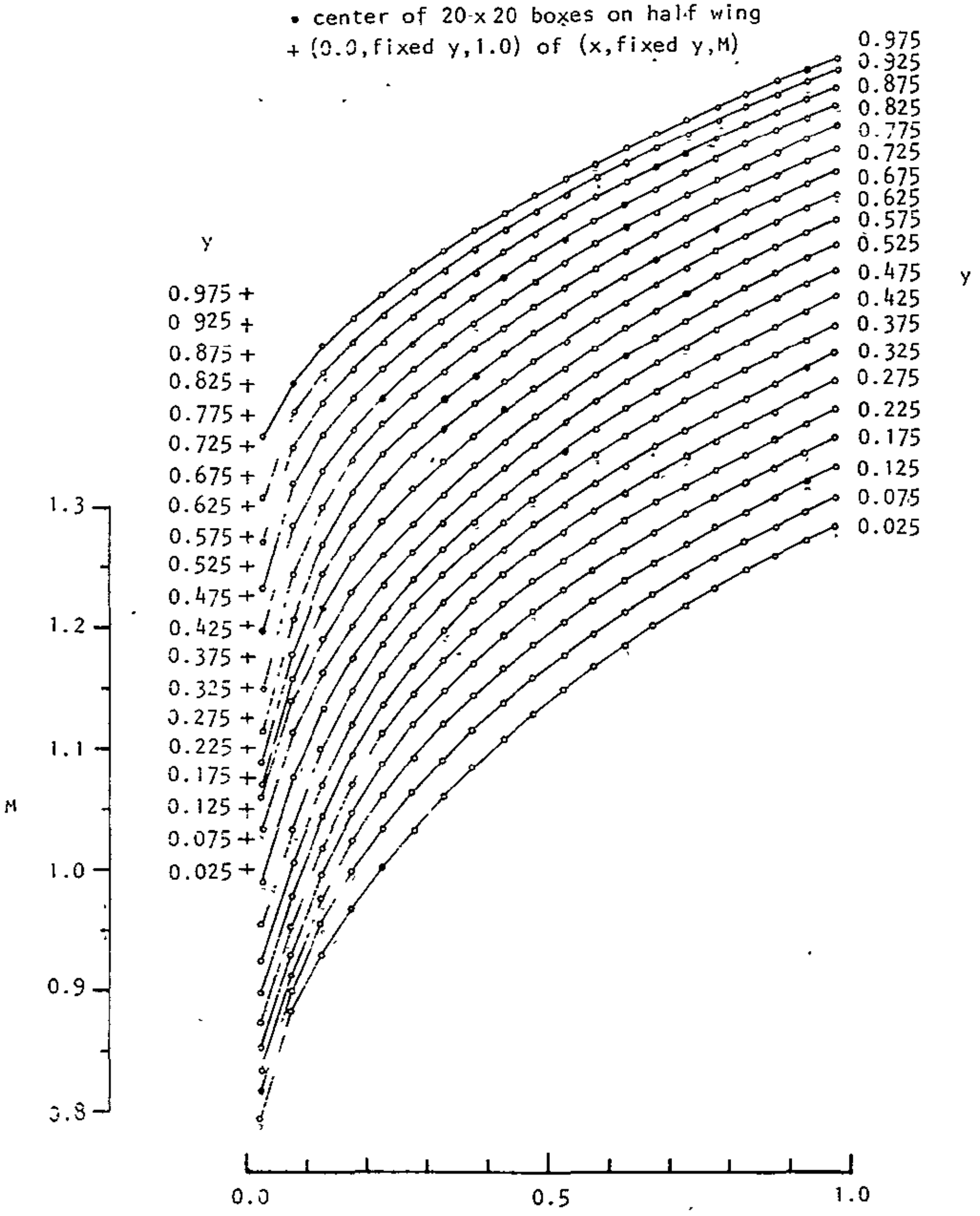
Figure 5. - Input steady state (mean) pressure coefficient on rectangular wing of aspect ratio 2.0 and thickness ratio 0.0521 (reference 8).

ORIGINAL PAGE IS
OF POOR QUALITY



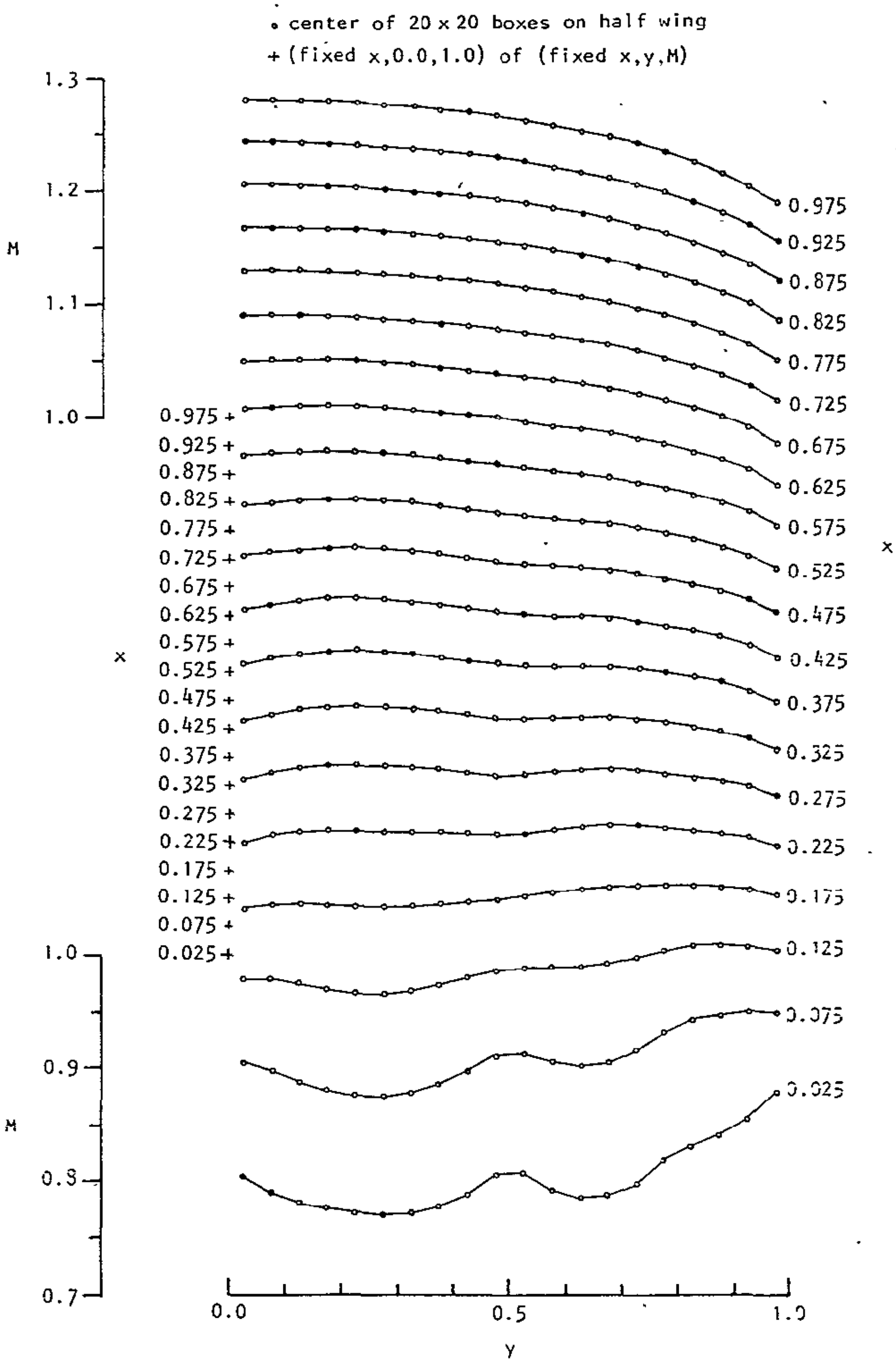
(b) Spanwise direction.

Figure 5. - Concluded.



(a) Chordwise distribution.

Figure 6. - Computed steady state (mean) Mach number on rectangular wing of aspect ratio 2.0 and the thickness ratio 0.0521.



(b) Spanwise distribution.

Figure 6. - Concluded.

APPENDIX

COMPUTER PROGRAM

The computer program listed in this appendix is dimensioned to handle a maximum of 30 boxes, either in chordwise or spanwise directions, in approximating the wing planform. The maximum numbers of leading and trailing edge segments are, respectively, 7 and 2 per semispan. The program can handle up to 3 wing deflection mode shapes. The maximum number of points used in spline-surface fitting is 100, so the maximum number of input points to describe the wing deflections and the Mach number or pressure coefficient distribution is also 100. These limitations can easily be increased by changing the dimensions of the corresponding variables in the computer program.

In order to activate the smoothing option in two-dimensional cubic-spline data fitting, it is required that a two-digit fixed point number be assigned to the last argument, NSMOS, of subroutine SPLN1. The right digit is for the control of the number of smoothings desired; and the left-digit is for pre-interpolation, zero for omitting pre-interpolation and non-zero for including pre-interpolation. The pre-interpolation is a process to increase the number of known points to be used in the interpolation by inserting an additional point betwen every successive pair of input points in the original set.

Input Guide

Data are input through the subroutine DATRD using the one dimensional array DA with a size of 1005. The allowable maximum number for some of the input data as indicated below may be changed if the dimensions of the corresponding storage array and computational operations are also changed accordingly. Subroutine DATRD initializes DA(1) through DA(22) to blank, the weighting factors in DA(104), DA(108), ---, DA(500) to 1.0, and the remaining portion of the DA array to 0.0. Consequently, these are the default values. The layout of the array DA(k) as it is presently used is similar to that in reference 4 and is as follows:

1-7: Title
8-12: Not used
13-19: Mode title
20-22: Not used
23: Frequency, (cycle/sec)
24: Overall length of wing in streamwise direction, (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 in streamwise direction (maximum 30)
28: Number of deflection modes (maximum 3)
29: Number (m) of segments of leading edge per semispan to be given, excluding segment from origin to y_0 ($m_{max} = 7$)
30-44: Coordinates of points on the leading edge, (ft or meter) (in sequence of $y_0, x_1, y_1, x_2, y_2, ---, x_m, y_m$), $m_{max} = 7$
45: Number (n) of segments of trailing edge per semispan to be given, (default: unswept trailing edge), $n_{max} = 2$
46-48: Coordinates of points on the trailing edge, (ft or meter)
(in sequence of x_0, y_1, x_1 for $n = 2$,
or x_0 (only) for $n = 1$,
no input for $n = 0$;
last trailing edge point coincides with the last leading edge point and is set internally)

49: Number of boxes allowed for upstream influence (if this location is left blank or assigned a zero, it will assume DA(49)=DA(27) and in no case DA(49)>DA(27) is allowed).

50: (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

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

52-53: Coefficients of the deflection polynomial (in the sequence of a_0 and a_1)

54-70: Not used*

71-72: Coefficients of the Mach number distribution polynomial (in the sequence of a_0 and a_1)

73-95: Not used*

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

99-100: Not used*

101-500: Deflection data for a maximum of 100 points (in the sequence of x, y, deflection and weighting factor)

501-700: Not used**

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 positive integer number should be entered at locations in the DA array corresponding to the information from the one particular subroutine that is needed.

1001: CBA: for wing deflection (DRED)

$\left.\begin{array}{l}\text{A=1, for NEW=1}\\ \text{B=1, for NEW=2}\end{array}\right\}$ spline-surface fitted results

C - not applicable

1002: BA: for wing upwash (WVAL)

A=1, - upwash

B - not applicable

1003: FEDCBA for velocity potential (BOXP and BOXPO)

$\left.\begin{array}{l}\text{A=1, for NEW=1}\\ \text{B=1, for NEW=2}\end{array}\right\}$ velocity potential

$\left.\begin{array}{l}\text{C=1, for NEW=1}\\ \text{D=1, for NEW=2}\end{array}\right\}$ influence coefficient and solution matrices

$\left.\begin{array}{l}\text{E=1, for NEW=1}\\ \text{F=1, for NEW=2}\end{array}\right\}$ pressure coefficient

1004: A: for Mach number (MRED)

A=1 - spline-surface fitted results

1005: DCBA: for wing shape (SHAPE and PLNFM)

$\left.\begin{array}{l}\text{A=1, for NEW=1}\\ \text{B=1, for NEW=2}\end{array}\right\}$ distributions of box, box area, leading and trailing edges

C=1 - Mach number at box centers

D - not applicable

The format of the input data card is (A1, A5, I6, 6A10, A8). 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 and fifth fields are 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.

Those storages currently not used in array DA marked with * are reserved for future improvements in the method used for the functional form of data input. Those marked with ** are reserved for the case where large numbers of data input points for deflections or Mach numbers are required.

Sample Input

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

The input format is (A1, A5, I6, 6A10, A8).

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 1).

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 (Mach number for this case), number of points on the wing this information to be given.

Cards 10 to 69:
chordwise and spanwise coordinates (ft) of a point on the wing, and the Mach number 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 70 and 71:
additional frequencies for the same wing, one card is read in at one time.

Card 72: blank card to make an exit from the computer.

The card images are as follows:

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

+ALPHA     1ASPECT RATIO 1.5 DELTA WING (TAU=0.10)                             1
 ALPHA    13PLUNGE                                                             2
          23 0.159154941        10.0      1000.0                               3
          27          30           2           1                               4
          30         0.0        10.0        3.75                               5
-         52         1.0                                                       6
 ALPHA    13PITCH ABOUT ROOT LEADING EDGE X=0.0                                7
          52         0.0         0.1                                           8
          96           1          60                                           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
         779         1.7         0.5    0.122340                              36
         782         3.3         1.0    0.113564                              37
         785         4.5         1.0    0.107428                              38
         788         5.7         1.0    0.100872                              39
         791         6.9         1.0    0.093129                              40
         794         8.3         1.0    0.080342                              41
         797         9.5         1.0    0.056823                              42
         800         4.5         1.5    0.107428                              43
         803         6.3         2.0    0.097215                              44
         806         7.5         2.0    0.088395                              45
         809         8.7         2.0    0.074970                              46
         812         9.7         2.0    0.047661                              47
         815         8.3         2.5    0.080342                              48
         818         9.7         3.0    0.047661                              49
         821         2.5         0.5    0.117746                              50
         824         3.9         0.5    0.110510                              51
         827         5.1         0.5    0.104243                              52
         830         6.3         0.5    0.097215                              53
         833         7.5         0.5    0.088395                              54
         836         9.1         0.5    0.067825                              55
         839         9.9         0.5    0.029651                              56
         842         5.1         1.5    0.104243                              57
         845         6.3         1.5    0.097215                              58
         848         7.9         1.5    0.084697                              59
         851         9.1         1.5    0.067825                              60
         854         9.9         1.5    0.029651                              61
         857         5.7         2.0    0.100872                              62
         860         6.9         2.5    0.093129                              63
         863         9.5         2.5    0.056823                              64
         866         9.9         2.5    0.029651                              65
         869         8.3         3.0    0.080342                              66
         872         9.1         3.0    0.067825                              67
         875         9.5         3.5    0.056823                              68
         878         9.9         3.5    0.029651                              69
          23 1.591549407                                   1                  70
          23 6.366197628                                   1                  71
                                                                              72
```

Sample Output

```
        ASPECT RATIO 1.5 DELTA WING (TAU=0.10)

        30 BOXES ALONG ROOT CHORD                  ROOT CHORD LENGTH    =   10.00 FT

        REDUCED FREQUENCY =  .010                  FREE STREAM VELOCITY = 1000.00 FT/SEC

        FREQUENCY =  1.592E-01 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         2.15396E-06       -2.42783E-02         2.42783E-02        -89.9949

   1     2        -1.69639E-07       -1.61159E-02         1.61159E-02        -90.0006

        GENERALIZED FORCES (NO THICKNESS EFFECT)

   MODES
 PRES. DEFL.         REAL PART          IMAG PART           ABS. VALUE       PHASE ANGLE

   2     1        -2.42788E+00       -2.44689E-02         2.42800E+00       -179.4226

   2     2        -1.61162E+00       -1.81349E-02         1.61173E+00       -179.3553

             MODE NO.  1

             MODE NO.  2

        GENERALIZED FORCES (WITH THICKNESS EFFECT)

   MODES
 PRES. DEFL.         REAL PART          IMAG PART           ABS. VALUE       PHASE ANGLE

   1     1        -1.00531E-05       -2.43535E-02         2.43535E-02        -90.0237

   1     2        -1.29338E-05       -1.61534E-02         1.61534E-02        -90.0459

        GENERALIZED FORCES (WITH THICKNESS EFFECT)

   MODES
 PRES. DEFL.         REAL PART          IMAG PART           ABS. VALUE       PHASE ANGLE

   2     1        -2.43541E+00       -2.33020E-02         2.43552E+00       -179.4518

   2     2        -1.61539E+00       -1.68848E-02         1.61548E+00       -179.4011
```

```
ASPECT RATIO 1.5 DELTA WING (TAU=0.10)

30 BOXES ALONG ROOT CHORD                    ROOT CHORD LENGTH     =   10.00 FT

REDUCED FREQUENCY =  .100                    FREE STREAM VELOCITY = 1000.00 FT/SEC

FREQUENCY =  1.592E+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        3.64268E-04      -2.42141E-01        2.42141E-01       -89.9138

  1     2        1.20006E-04      -1.60604E-01        1.60604E-01       -89.9572

GENERALIZED FORCES (NO THICKNESS EFFECT)

  MODES
PRES. DEFL.       REAL PART         IMAG PART          ABS. VALUE       PHASE ANGLE

  2     1       -2.42556E+00      -2.45153E-01        2.43792E+00      -174.2287

  2     2       -1.60948E+00      -1.81827E-01        1.61972E+00      -173.5545


     MODE NO.  1

     MODE NO.  2

GENERALIZED FORCES (WITH THICKNESS EFFECT)

  MODES
PRES. DEFL.       REAL PART         IMAG PART          ABS. VALUE       PHASE ANGLE

  1     1       -7.33272E-04      -2.42506E-01        2.42507E-01       -90.1732

  1     2       -1.03563E-03      -1.60598E-01        1.60601E-01       -90.3695

GENERALIZED FORCES (WITH THICKNESS EFFECT)

  MODES
PRES. DEFL.       REAL PART         IMAG PART          ABS. VALUE       PHASE ANGLE

  2     1       -2.43011E+00      -2.34356E-01        2.44138E+00      -174.4915

  2     2       -1.61039E+00      -1.70175E-01        1.61936E+00      -173.9678
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
ASPECT RATIO 1.5 DELTA WING (TAU=0.10)

30 BOXES ALONG ROOT CHORD                 ROOT CHORD LENGTH    =   10.00 FT

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

FREQUENCY =  6.366E+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        2.60265E-02      -9.51379E-01        9.51735E-01        -88.4330

 1     2        1.93704E-02      -6.28441E-01        6.28740E-01        -88.2345

GENERALIZED FORCES (NO THICKNESS EFFECT)

 MODES
PRES. DEFL.       REAL PART         IMAG PART          ABS. VALUE       PHASE ANGLE

 2     1       -2.41529E+00      -9.98983E-01        2.61373E+00       -157.5296

 2     2       -1.60030E+00      -7.44074E-01        1.76483E+00       -155.0635

     MODE NO.  1

     MODE NO.  2

GENERALIZED FORCES (WITH THICKNESS EFFECT)

 MODES
PRES. DEFL.       REAL PART         IMAG PART          ABS. VALUE       PHASE ANGLE

 1     1        1.95356E-02      -9.43362E-01        9.43564E-01        -88.8137

 1     2        1.17730E-02      -6.19079E-01        6.19191E-01        -88.9105

GENERALIZED FORCES (WITH THICKNESS EFFECT)

 MODES
PRES. DEFL.       REAL PART         IMAG PART          ABS. VALUE       PHASE ANGLE

 2     1       -2.39926E+00      -9.75339E-01        2.58993E+00       -157.8775

 2     2       -1.58205E+00      -7.16462E-01        1.73672E+00       -155.6357
```

Program Listing

```
      PROGRAM SBOXR2(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)             SBOXR2
C     MAIN PROGRAM                                                      SBOXR2
C     MB=MAXIMUM NUMBER OF BOXES ALONG THE CENTERLINE CHORD ALLOWED     SBOXR2
C        IN DIMENSION STATEMENT                                         SBOXR2
C     MD=MAXIMUM NUMBER OF MODES ALLOWED IN DIMENSION STATEMENT         SBOXR2
C     NB=MAXIMUM NUMBER OF BOXES ALLOWED FOR SPLINE FITTING             SBOXR2
C     NM=NB+3 ALLOWED IN DIMENSION STATEMENT                            SBOXR2
      COMMON /A/ XX(30),YY(30),XLE(30),XTE(30),EDG(30),AR(30,30)        A
     C          ,ML(30),MLE(30),MLT(2,30),MLC(2,30),MB,AMA(30,30)       A
     C          ,XEDG(8),YEDG(8),NEDGI,NS,XTDG(3),YTDG(3),NTDGI,NST     A
     C          ,SFMX(100),SFMY(100),SFMH(103),KSFM,T(103,104),NB,NM    A
     C          ,DA(1005),AREA,CK,D,DH,DI,YMAX,JMAX,JMAK,IEDG,IW,L,NEW  A
      DIMENSION A(2, 60,30),S(2,30,30)                                  SBOXR2
     1          ,SFDX(100,3),SFDY(100,3),SFDH(103,3),KSFD(3),G(3)       SBOXR2
      DATA Z/1H /                                                       SBOXR2
      IR=5                                                              SBOXR2
      IW=6                                                              SBOXR2
      NEDGI=8                                                           SBOXR2
      NTDGI=3                                                           SBOXR2
      MB=30                                                             SBOXR2
      MD=5                                                              SBOXR2
      NB=100                                                            SBOXR2
C     MATCH ABOVE NUMBERS TO THE DEFINED DIMENSION IN STORAGE ARRAYS    SBOXR2
      ME=MB+1                                                           SBOXR2
      MC=2*MB                                                           SBOXR2
      NM=NB+3                                                           SBOXR2
      WRITE(IW,55)                                                      SBOXR2
C     READ DATA FOR THE ACTUAL WING                                     SBOXR2
  100 CALL DATRD(DA)                                                    SBOXR2
      NEW=1                                                             SBOXR2
      TEST1=DA(1001)+DA(1002)+DA(1003)+DA(1004)+DA(1005)                SBOXR2
      CK=DA(23)*DA(24)/DA(25)*6.28318531                                SBOXR2
      DI=DA(27)                                                         SBOXR2
      L=DI                                                              SBOXR2
      D=1.0/DI                                                          SBOXR2
      DH=0.5*D                                                          SBOXR2
      WRITE(IW,60) (DA(I),I=1,7)                                        SBOXR2
  110 IF (L) 600,600,120                                                SBOXR2
  120 IF (MB-L) 600,130,130                                             SBOXR2
  130 WRITE(IW,65) L,DA(24),CK,DA(25),DA(23)                            SBOXR2
      IF (DA(26)) 160,150,160                                           SBOXR2
  150 CALL SHAPE                                                        SBOXR2
      IF(NEW.EQ.2) GO TO 180                                            SBOXR2
      AC=8.0/AREA                                                       SBOXR2
  160 LIM=ML(L)                                                         SBOXR2
      IF (LIM-MB) 170,170,650                                           SBOXR2
  170 LIM2=2*MB                                                         SBOXR2
      NFLNS = DA(49)                                                    SBOXR2
      IF(NFLNS.EQ.0) NFLNS = L                                          SBOXR2
      LPOT = MINO(L,NFLNS)                                              SBOXR2
      LIM1=2*MB                                                         SBOXR2
      CALL           POT2(LIM1,LIM2,LPOT,CK,D,A)                        SBOXR2
  180 CONTINUE                                                          SBOXR2
      M=0                                                               SBOXR2
      K=DA(28)                                                          SBOXR2
      GO TO 230                                                         SBOXR2
C                                                                       SBOXR2
C     PRELIMINARY CALCULATIONS ARE FINISHED.                            SBOXR2
C     THE NEXT SECTION IS GONE THROUGH FOR EACH MODE.                   SBOXR2
C                                                                       SBOXR2
  200 IF (DA(26)) 230,210,230                                           SBOXR2
  210 IF(NEW.EQ.2) GO TO 230                                            SBOXR2
      CALL DATRD(DA)                                                    SBOXR2
```

```
  230 K=K-1                                                          SBOXR2      61
      M=M+1                                                          SBOXR2      62
      IF(TEST1.LT.1.0) GO TO 250                                     SBOXR2      63
      WRITE(IW,55)                                                   SBOXR2      64
  250 WRITE(IW,15) M                                                 SBOXR2      65
      WRITE(IW,16) (DA(I),I=13,19)                                   SBOXR2      66
   15 FORMAT(1H0,15X,8HMODE NO.,I3)                                  SBOXR2      67
   16 FORMAT(1H+,30X,7A10)                                           SBOXR2      68
      IF (DA(26)) 290,280,290                                        SBOXR2      69
C                                                                    SBOXR2      70
  280 IF (NEW.EQ.2) GO TO 290                                        SBOXR2      71
      IPRINT=DA(1001)                                                SBOXR2      72
      CALL        DRED(SFDX,SFDY,SFDH,KSFD,SFMX,SFMY,SFMH,KSFM,DA,T   SBOXR2      73
     $                 ,XX,YY,IW,L,MLC,M,MD,NB,NEW,NM,KSFS,IPRINT)   SBOXR2      74
C                                                                    SBOXR2      75
      G(M)=DA(51)                                                    SBOXR2      76
  290 IF(K) 310,310,300                                              SBOXR2      77
  300 IF (M-MD) 200,310,310                                          SBOXR2      78
  310 CONTINUE                                                       SBOXR2      79
C                                                                    SBOXR2      80
      IF(IFIX(DA(50)).EQ.2.AND.NEW.EQ.1) GO TO 490                   SBOXR2      81
      DO 460 M1=1,M                                                  SBOXR2      82
      IF (G(M1))  460,320,460                                        SBOXR2      83
  320 CONTINUE                                                       SBOXR2      84
C                                                                    SBOXR2      85
      IPRINT=DA(1002)                                                SBOXR2      86
      CALL        KVAL(XX,YY,XTDG,YTDG,XTE,SFDX,SFDY,SFDH,KSFD,S,D,CK SBOXR2      87
     $                 ,IW,L,ML,M1,NEW,NST,MB,MD,NB,NM,KSFS,IPRINT)  SBOXR2      88
C                                                                    SBOXR2      89
      IF (EDG(1)) 360,360,330                                        SBOXR2      90
  330 EDGE=EDG(1)                                                    SBOXR2      91
C     LEADING EDGE CORRECTION                                        SBOXR2      92
      DO 350 J=1,LMZ                                                 SBOXR2      93
  350 S(J,1,1)=S(J,1,1)*EDGE/3.14159265                              SBOXR2      94
  360 CONTINUE                                                       SBOXR2      95
C                                                                    SBOXR2      96
      YE=YLE(1)                                                      SBOXR2      97
      JMAK   =DA(49)                                                 SBOXR2      98
      IPRINT=DA(1003)                                                SBOXR2      99
      CALL        CUYF(XX,YY,XTDG,YTDG,XTE,XLE,A,AR,DAN,T,EDG,S,CK,D,YE SBOXR2    100
     $                 ,IPRINT,MLT,MLC,AMA                           SBOXR2     101
     $                 ,JMAK,L,ML,MLH,M1,NEW,NST,IW,MB,MC,ME,MD)     SBOXR2     102
C                                                                    SBOXR2     103
  460 CONTINUE                                                       SBOXR2     104
C                                                                    SBOXR2     105
C     COMPUTATION OF GENERALIZED FORCES FOR EACH MODE                SBOXR2     106
C                                                                    SBOXR2     107
      IF(TEST1.LT.1.0) GO TO 410                                     SBOXR2     108
      WRITE (IW,55)                                                  SBOXR2     109
      WRITE(IW,60) (DA(I),I=1,7)                                     SBOXR2     110
      WRITE(IW,61) L,DA(4),CK,DA(25),DA(23)                          SBOXR2     111
  410 WRITE (IW,20)                                                  SBOXR2     112
      IF(NEW.EQ.1) WRITE(IW,22)                                      SBOXR2     113
      IF(NEW.EQ.2) WRITE(IW,24)                                      SBOXR2     114
      WRITE(IW,25)                                                   SBOXR2     115
   20 FORMAT(/1H010X,18HGENERALIZED FORCES)                          SBOXR2     116
   22 FORMAT(1H+,28X,22H (NO THICKNESS EFFECT))                      SBOXR2     117
   24 FORMAT(1H+,26X,24H (WITH THICKNESS EFFECT))                    SBOXR2     118
   25 FORMAT(1H0,5X,5HMODES/4X,11HPRES. DEFL.,8X,9HREAL PART,10X,9HIMAG SBOXR2     119
     $PART,10X,10HABS. VALUE,6X,11HPHASE ANGLE)                      SBOXR2     120
C                                                                    SBOXR2     121
      DO 470 M2=1,M                                                  SBOXR2     122
C                                                                    SBOXR2     123
      CALL        FORCI(XX,YY,S,SFDX,SFDY,SFDH,KSFD,KSFS,XLE,XTE,YMAX SBOXR2     124
     1                 ,AMA,JMAK,MLT,L,NEW,M2,MB,MD,NB,NM,CK,T)      SBOXR2     125
C                                                                    SBOXR2     126
      S1=AC*T(1,1)                                                   SBOXR2     127
      S2=AC*T(2,1)                                                   SBOXR2     128
      S3= SQRT(S1**2+S2**2)                                          SBOXR2     129
      S4=57.29578*ATAN2(S2,S1)                                       SBOXR2     130
```

```
  450 WRITE(IW,30) M1,M2,S1,S2,S3,S4                                    SBOXR2     131
   30 FORMAT (1H0216,1P3E19.5,0P1F16.4)                                 SBOXR2     132
  470 CONTINUE                                                          SBOXR2     133
  480 WRITE  (IW,55)                                                    SBOXR2     134
  490 CONTINUE                                                          SBOXR2     135
      IF(DA(26).GT.0.) GO TO 510                                        SBOXR2     136
      DO 500  I=13,22                                                   SBOXR2     137
  500 DA(I)=Z                                                           SBOXR2     138
  510 IF (IFIX(DA(50))-1.EQ.0) GO TO 100                                SBOXR2     139
      IF(NEW.EQ.2) GO TO 100                                            SBOXR2     140
      NEW=2                                                             SBOXR2     141
      IF(DA(26)) 180,580,180                                            SBOXR2     142
  580 IPRINT=DA(1004)                                                   SBOXR2     143
      CALL        MRED(DA,T,NM,NB,KSFM,SFMX,SFMY,SFMH,IW,IPRINT)        SBOXR2     144
      GO TO 150                                                         SBOXR2     145
C                                                                       SBOXR2     146
C     ERROR EXITS                                                       SBOXR2     147
C                                                                       SBOXR2     148
  590 IPR=30                                                            SBOXR2     149
      GO TO 610                                                         SBOXR2     150
  600 IPR=27                                                            SBOXR2     151
  610 WRITE(IW,35) IPR                                                  SBOXR2     152
   35 FORMAT(1H010X,8HBAD DATAI4)                                       SBOXR2     153
      GO TO 700                                                         SBOXR2     154
  650 WRITE   (IW,40)                                                   SBOXR2     155
   40 FORMAT(1H010X,42HLATERAL LIMIT ON NUMBER OF BOXES EXCEEDED.)      SBOXR2     156
  700 STOP                                                              SBOXR2     157
   55 FORMAT(1H1)                                                       SBOXR2     158
   60 FORMAT(1H0,10X,7A10)                                              SBOXR2     159
   65 FORMAT(1H0,10X,I2,23H BOXES ALONG ROOT CHORD,                     SBOXR2     160
     1             15X,22HROOT CHORD LENGTH    =,F8.2,3H FT/            SBOXR2     161
     2        1H0,10X,19HREDUCED FREQUENCY =,F6.3,                      SBOXR2     162
     3             15X,22HFREE STREAM VELOCITY =,F8.2,7H FT/SEC/        SBOXR2     163
     4        1H0,10X,11HFREQUENCY =,1PE11.3,10H CYCLE/SEC)             SBOXR2     164
      END                                                               SBOXR2     165
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      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=1,7)
    2 FORMAT(A1,A5,I6,6A10,A8)
      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=23,1005
  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=1,6
      DDRBU(I)=DRBU(I)
    3 CONTINUE
      DECODE(80,990,DDRBU)(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
      GO TO 11
C     ALPHA CARD
    9 DO 10 I=1,7
      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
   12 RETURN
C     END OF DATA CARDS
   20 WRITE (IW,995)
      CALL EXIT
      STOP
C     BAD CARD
    8 CONTINUE
      WRITE
     1    (IW,993) EMIN,ALP,IND,(DRBU(I),I=1,7)
      WRITE    (IW,991)
      STOP
  990 FORMAT(8E10.0)
  991 FORMAT([illegible]H BAD DATA ON THIS CARD. JOB TERMINATED .)
  993 FORMAT(12H0CARD IMAGE=A1,A5,I6,7A10)
  995 FORMAT(///1H ,10X,17H END OF DATA CARD/)
      END
```

```
      [illegible]
     1 [illegible],YY( ),XLE(30),XTL(30),EDG(30),AR(30,30)
     2 [illegible],MLT(2,30),MLC(2,30),MB,AMA(30,30)
     3 ,XEDG(8),YEDG(8),NEDGI,NS,XTDG(3),YTDG(3),NTDGI,NST
     4 ,SFMX(100),SFMY(100),SFMH(103),KSFM,T(103,104),NB,NM
     5 ,[illegible],YMAX,JMAX,JMAK,IEDG,IW,L,NEW
      [illegible]
      [illegible]
      KRITE=IPRIN/1000
      IPRIN=IPRIN-1000*KRITE
      JRITE=IPRIN/100
      IPRIN=IPRIN-100*JRITE
      IRITE=IPRIN/10
      IF(NEW.EQ.1) IRITE=IPRIN-10*IRITE
      IF(NEW.EQ.2) GO TO 200
      IEDG=0
      NS=DA(29)
      IF (NS) 850,850,100
  100 IF (NS-NEDGI+1) 105,105,850
  105 NSP=NS+1
      NS=NSP
      IF (DA(24)) 855,855,110
  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
      NST=DA(45)
      IF (NST) 871,125,130
  125 XTDG(1)=1.0
      NST=1
      GO TO 141
  130 IF (NST-NTDGI+1) 135,135,871
  135 NSTP=NST
      DO 140 I=1,NSTP
      YTDG(I)=DA(2*I+43)/DA(24)
  140 XTDG(I)=DA(2*I+44)/DA(24)
  141 NSTP=NST+1
      NST=NSTP
      XTDG(NSTP)=XEDG(NSP)
      YTDG(NSTP)=YEDG(NSP)
      YTDG(1)=0.0
      IF (NST.EQ.NTDGI) GO TO 150
      JL=NST+1
      DO 142 I=JL,NTDGI
      XTDG(I)=XTDG(I-1)
  142 YTDG(I)=YTDG(I-1)
  150 IF(ABS(XTDG(NSTP)-XEDG(NSP)+YTDG(NSTP)-YEDG(NSP)).GT.ERRR)
     $   GO TO 871
      GO TO 550
C
C     FOR WING WITH THICKNESS
C     COMPUTE MACH NUMBER DISTRIBUTION ON PHYSICAL WING
  200 DM=0.
      DO 220 I=1,MB
      DO 210 J=1,MB
  210 AMA(J,I)=1.0
      DO 215 I=1,L
      IF (MLC(1,I).EQ.0) GO TO 215
      J=MLC(1,I)
      K=MLC(2,I)
      CALL SURF2(XX(I),YY,J,K,1,AMA(1,I),DM,DM,SFMX,SFMY,SFMH,KSFM,1)
  215 CONTINUE
      IF(JRITE.EQ.0) GO TO 235
C     PRINT OUT MACH DISTRIBUTION
      WRITE(IW,60)
      DO 230 I=1,L
      IF (MLC(1,I).EQ.0) GO TO 230
```

```
      JL=MLC(2,I)-MLC(1,I)+1                                            SHAPE     66
      IF(JL.EQ.0) GO TO 230                                             SHAPE     67
      WRITE(IW,70)  I                                                   SHAPE     68
      JLP=JL/6                                                          SHAPE     69
      IF(JL-6*JLP.NE.0) JLP=JLP+1                                       SHAPE     70
      K1=MLC(1,I)-1                                                     SHAPE     71
      JL=MLC(2,I)                                                       SHAPE     72
      DO 225 J=1,JLP                                                    SHAPE     73
      K=K1+J                                                            SHAPE     74
  225 WRITE(IW,80) ((J1,AMA(J1,I)),J1=K,JL,JLP)                         SHAPE     75
  230 CONTINUE                                                          SHAPE     76
C                                                                       SHAPE     77
  235 CONTINUE                                                          SHAPE     78
      GO TO 750                                                         SHAPE     79
  550 Y1=YEDG(1)                                                        SHAPE     80
      YMAX=YEDG(NSP)                                                    SHAPE     81
      IF(Y1.LT.0.0) GO TO 860                                           SHAPE     82
      AREA=0.0                                                          SHAPE     83
      XX(1)=DH                                                          SHAPE     84
      YY(1)=DH                                                          SHAPE     85
      DO 560 I=2,MB                                                     SHAPE     86
      YY(I)=YY(I-1)+D                                                   SHAPE     87
  560 XX(I)=XX(I-1)+D                                                   SHAPE     88
C                                                                       SHAPE     89
C     CALCULATE PHYSICAL FULL WING AREA  AND  ML(NEW,I)                 SHAPE     90
C     COMPUTE AND IDENTIFY BOX DISTRIBUTION ON THE WING PLANFORM        SHAPE     91
  565 CALL        PLNFM(XEDG,YEDG,XTDG,YTDG,XX,YY,XLE,XTE,AR,AREA,D     SHAPE     92
     $                  ,IW,L,MB,ML,MLC,MLW,NEW,NS,NST,IRITE)           SHAPE     93
C                                                                       SHAPE     94
      IF (ML(L).GT.MB) GO TO 825                                        SHAPE     95
C                                                                       SHAPE     96
      JMAX=0                                                            SHAPE     97
      JMAK=0                                                            SHAPE     98
      DO 570 I=1,L                                                      SHAPE     99
      JMAK=MAX0(JMAK,MLC(2,I))                                          SHAPE    100
  570 JMAX=MAX0(JMAX,ML (I))                                            SHAPE    101
      JMX=JMAK                                                          SHAPE    102
C                                                                       SHAPE    103
C     FIND ORDER OF LEADING EDGE BOX AT J-TH CHORDWISE ROW              SHAPE    104
C                                                                       SHAPE    105
      DO 620 J=1,JMAX                                                   SHAPE    106
      EDG(J)=0.0                                                        SHAPE    107
      DO 620 K=1,2                                                      SHAPE    108
  620 MLT(K,J)=0                                                        SHAPE    109
      I1=1                                                              SHAPE    110
      I2=1                                                              SHAPE    111
      K1=0                                                              SHAPE    112
      DO 655 J=1,JMX                                                    SHAPE    113
  630 IF (MLC(2,I1).GE.J) GO TO 632                                     SHAPE    114
      I1=I1+1                                                           SHAPE    115
      GO TO 630                                                         SHAPE    116
  632 NZ1=I1                                                            SHAPE    117
C     FIND ORDER OF TRAILING EDGE BOX AT J-TH CHORDWISE ROW             SHAPE    118
  634 IF(I2.GT.L) GO TO 638                                             SHAPE    119
      IF (MLC(2,I2).GT.0) K1=1                                          SHAPE    120
      IF(K1.EQ.0) GO TO 635                                             SHAPE    121
      IF (MLC(2,I2).EQ.0.AND.NZ1.NE.0) GO TO 646                        SHAPE    122
  635 MLS=IABS(MLW(I2))                                                 SHAPE    123
      IF (MLW(I2).NE.0) GO TO 636                                       SHAPE    124
      I2=I2+1                                                           SHAPE    125
      K2=1                                                              SHAPE    126
      GO TO 634                                                         SHAPE    127
  636 IF(MLS-J) 644,646,638                                             SHAPE    128
  638 IF(K2.LT.0) GO TO 648                                             SHAPE    129
      IF (MLW(I2)) 642,642,646                                          SHAPE    130
  642 I2=I2+K2                                                          SHAPE    131
      IF(I2.LE.L) GO TO 634                                             SHAPE    132
      N22=L                                                             SHAPE    133
      I2 =L                                                             SHAPE    134
      GO TO 652                                                         SHAPE    135
  644 K2=MLW(I2)/MLS                                                    SHAPE    136
```

```
      GO TO 642
  646 NZ2=I2-1
      GO TO 652
  648 NZ2=I2
  652 CONTINUE
      MLT(1,J)=NZ1
      MLT(2,J)=NZ2
  655 CONTINUE
C
C     FIND ORDER OF WING BOXES AT EACH I-TH SPANWISE COLUMN
      DO 670 I=1,L
      IF (MLW(I)) 664,662,662
  662 MLC(1,I)=MLW(I)+1
      IF (MLC(2,I).EQ.0) MLC(1,I)=0
      GO TO 670
  664 MLC(2,I)=IABS(MLW(I))-1
      MLC(1,I)=1
  670 CONTINUE
C
      IF(IRITE.EQ.0) GO TO 678
      WRITE(IW,65)
      IF(NEW.EQ.1) WRITE(IW,77)
      IF(NEW.EQ.2) WRITE(IW,78)
      WRITE(IW,82)
      DO 672 J=1,JMX
  672 WRITE(IW,85) J,MLT(1,J),MLT(2,J)
      WRITE(IW,83)
      DO 674 J=1,L
C 674 WRITE(IW,85) J,MLC(1,J),MLC(2,J)
  674 WRITE(IW,85) J,MLC(1,J),MLC(2,J),MLW(J),ML(J)
  678 CONTINUE
C
      IF(XEDG(NSP).GT.1.) GO TO 965
      NSM=NSP-1
      DY=YEDG(NSP)-YEDG(NSM)
      IF(ABS(DY).LE.ERRR) IEDG=1
      IF(DY.LT.0.) GO TO 827
C      COMPUTE VALUES FOR LEADING EDGE CORRECTION
      YMAX2=YMAX*YMAX
      DY=0
      DO 720 J=1,JMX
      IF (IEDG) 710,715,710
C     DY IN FOLLOWING EXPRESSION IS ARBITRARY
  710 EDG(J)=SQRT((YMAX2-YY(J)*YY(J))/(DY*(2.*YMAX-DY)))
      IF (EDG(J).GT.1.0) GO TO 715
      GO TO 720
  715 EDG(J)=1.0
  720 CONTINUE
C
  750 CONTINUE
C
  790 CONTINUE
C
      RETURN
C
  800 IERROR=800
      T(1,1)=XEDG(K)
      T(2,1)=YEDG(K)
      IER=2
      GO TO 840
  805 IERROR=805
      GO TO 840
  810 IERROR=810
      T(1,1)=DA(27)
      T(2,1)=NSP
      IER=2
      GO TO 840
  815 IERROR=815
      GO TO 840
  820 IERROR=820
      T(1,1)=XEDG(NSP)
```

```
      T(2,1)=YEDG(NSP)                                                  SHAPE     208
      T(3,1)=NSP                                                        SHAPE     209
      IER=3                                                             SHAPE     210
      GO TO 840                                                         SHAPE     211
  825 IERROR=825                                                        SHAPE     212
      T(1,1)=NEW                                                        SHAPE     213
      T(2,1)=K                                                          SHAPE     214
      T(3,1)=ML(K)                                                      SHAPE     215
      T(4,1)=ML(K-1)                                                    SHAPE     216
      IER=4                                                             SHAPE     217
      GO TO 840                                                         SHAPE     218
  827 IERROR=827                                                        SHAPE     219
      T(1,1)=YEDG(NSP)                                                  SHAPE     220
      T(2,1)=YEDG(NSM)                                                  SHAPE     221
      T(3,1)=NSP                                                        SHAPE     222
      T(4,1)=NSM                                                        SHAPE     223
      IER=4                                                             SHAPE     224
      GO TO 840                                                         SHAPE     225
  830 IERROR=730                                                        SHAPE     226
      T(1,1)=I                                                          SHAPE     227
      T(2,1)=J                                                          SHAPE     228
      T(3,1)=X                                                          SHAPE     229
      T(4,1)=Y                                                          SHAPE     230
      T(5,1)=EM                                                         SHAPE     231
      T(6,1)=EMY                                                        SHAPE     232
      T(7,1)=DEL                                                        SHAPE     233
      IER=7                                                             SHAPE     234
  840 WRITE(IW,20) IERROR, (T(I,1),I=1,IER)                             SHAPE     235
      STOP                                                              SHAPE     236
  850 IPR=29                                                            SHAPE     237
      GO TO 890                                                         SHAPE     238
  855 IPR=24                                                            SHAPE     239
      GO TO 890                                                         SHAPE     240
  860 IPR=30                                                            SHAPE     241
      GO TO 890                                                         SHAPE     242
  865 K=  MINO(K,NS)                                                    SHAPE     243
      IPR=2*K+29                                                        SHAPE     244
      GO TO 890                                                         SHAPE     245
  871 IPR=45                                                            SHAPE     246
  890 WRITE                                                             SHAPE     247
     0  (IW,10)IPR                                                      SHAPE     248
   10 FORMAT(1H0,10X,17HSHAPE -- BAD DATA,I5)                           SHAPE     249
   20 FORMAT(1H0,10X,38HBAD NUMBER IN SHAPE NEAR STATEMENT NO.,I5,      SHAPE     250
     1       /1H ,15X,1P8E14.6)                                         SHAPE     251
   40 FORMAT(/1H0,5X,3HNO.,I2,43H REDISTRIBUTION OF WING LEADING EDGES, SHAPE     252
     1 NS(,I1,4H) = ,I2,7H,  IEDG,3H = ,I1)                             SHAPE     253
   45 FORMAT(/1H0,5X,25HWING TRAILING EDGES, NST(,I1,4H) = ,I2,         SHAPE     254
     $        10H,  IEDG = ,I1)                                         SHAPE     255
   50 FORMAT(4(6X,I3,1P2E11.4))                                         SHAPE     256
   60 FORMAT(1H1,5X,47HLOCAL MACH NUMBER DISTRIBUTION ON PHYSICAL WING/) SHAPE     257
   65 FORMAT(1H1)                                                       SHAPE     258
   70 FORMAT(1H0,5X,I2,19H-TH SPANWISE COLUMN)                          SHAPE     259
   75 FORMAT(1H0,5X,I2,17H-TH CHORDWISE ROW)                            SHAPE     260
   77 FORMAT(40X,13HPHYSICAL WING///)                                   SHAPE     261
   78 FORMAT(40X,16HTRANSFORMED WING///)                                SHAPE     262
   80 FORMAT(1H ,5X,6(2X,I3,1PE13.5))                                   SHAPE     263
   82 FORMAT(15X,68HORDER OF FIRST(LEADING) AND LAST(TRAILING) WING BOX SHAPE     264
     1IN CHORDWISE ROW//20X,1HJ,3X,12HMLT(NEW,1,J),3X,12HMLT(NEW,2,J)/) SHAPE     265
   83 FORMAT(///15X,62HORDER OF FIRST(ROOT) AND LAST(TIP) WING BOX IN SP SHAPE     266
     1ANWISE COLUMN//20X,1HI,3X,12HMLC(NEW,1,I),3X,12HMLC(NEW,2,I)      SHAPE     267
     2    ,4X,11HMLW(NEW,I),5X,10HML(NEW,I))/)                          SHAPE     268
   85 FORMAT(16X,I5,5X,I5,3(10X,I5))                                    SHAPE     269
   86 FORMAT(20X,58HGAUSSIAN INTEGRATION POINTS IN CHORDWISE ROW - GX(NE SHAPE     270
     1W,K,J)//)                                                         SHAPE     271
   87 FORMAT(///20X,52HGAUSSIAN INTEGRATION POINTS IN SEMI-SPAN - GY(NEW SHAPE     272
     1,K)//)                                                            SHAPE     273
   88 FORMAT(///20X,60HSPANWISE COORDINATE (GYP) AND MACH NUMBER (GMP) A SHAPE     274
     $T GX(2,K,J)///)                                                   SHAPE     275
   89 FORMAT(1H ,5X,3(2X,I3,1P2E16.6))                                  SHAPE     276
      STOP                                                              SHAPE     277
      END                                                               SHAPE     278
```

```
      SUBROUTINE PLNFM(XEDG,YEDG,XTDG,YTDG,XX,YY,XLE,XTE,AR,AREA,D     PLNFM
     $              ,IW,L,NB,NL,NLC,NLW,NEW,NS,NST,JRYTE)              PLNFM
      DIMENSION XEDG(1),YEDG(1),XTDG(1),YTDG(1),XX(1),YY(1),AR(NB,1)   PLNFM
     1         ,NEW(1),NLC(2,1),NLW(1),XLE(1),XTE(1)                   PLNFM
      DATA ERRR/1.E-06/,ERR1/0.001/,ERR2/1.E-05/,AWAK/0.5/             PLNFM
      DATA Z1/2HF=/,Z2/2HG=/                                           PLNFM
      DH=0.5*D                                                         PLNFM
      D2=D*D                                                           PLNFM
      NSP =NS                                                          PLNFM
      NSTP=NST                                                         PLNFM
      DO 30 I=1,NB                                                     PLNFM
      NLC(2,I)=0                                                       PLNFM
      NLW(I)=0                                                         PLNFM
      XLE(I)=XEDG(NSP)                                                 PLNFM
      XTE(I)=XEDG(NSP)                                                 PLNFM
      DO 30 J=1,NB                                                     PLNFM
   30 AR(J,I)=0.0                                                      PLNFM
C                                                                      PLNFM
  210 K1=0                                                             PLNFM
      K2=NST                                                           PLNFM
      JK=1                                                             PLNFM
      XR=XEDG(1)                                                       PLNFM
      YR=YEDG(1)                                                       PLNFM
      X2=XR                                                            PLNFM
      Y2=YR                                                            PLNFM
      XE2=0.0                                                          PLNFM
      SUM=0.0                                                          PLNFM
      AREA=0.0                                                         PLNFM
      ASNG=1.0                                                         PLNFM
      JLT=1                                                            PLNFM
C                                                                      PLNFM
  220 DO 530 I=1,L                                                     PLNFM
      ICHQ=0                                                           PLNFM
      IFIN=0                                                           PLNFM
      IEDZ=0                                                           PLNFM
      XL=XR                                                            PLNFM
      YL=YR                                                            PLNFM
      XR=XX(I)+DH                                                      PLNFM
      J=0                                                              PLNFM
      SAM=0.0                                                          PLNFM
      IF(ASNG) 225,230,230                                             PLNFM
  225 IF(XR.LT.XTDG(1).AND.XTDG(1)-XR.GT.ERR2) GO TO 530               PLNFM
      IF(K2.EQ.0) GO TO 440                                            PLNFM
      GO TO 240                                                        PLNFM
  230 CONTINUE                                                         PLNFM
      IF (K1+1.GT.NS) GO TO 420                                        PLNFM
      IF(K1.EQ.0) GO TO 400                                            PLNFM
  240 ICHQ=0                                                           PLNFM
      IF(XR.GT.XE2) ICHQ=1                                             PLNFM
  250 J=J+1                                                            PLNFM
      JFIN=0                                                           PLNFM
      Y=YY(J)+DH                                                       PLNFM
      IF(ASNG) 252,256,256                                             PLNFM
  252 IF(ABS(XTDG(1)-XTDG(NSTP)).GT.ERR2.OR.I.NE.L) GO TO 254          PLNFM
      IF(Y-DH.GT.YTDG(NSTP)) GO TO 530                                 PLNFM
      XTE(J)=XTDG(1)                                                   PLNFM
      GO TO 250                                                        PLNFM
  254 SUM=SUM-AR(J,I)                                                  PLNFM
      GO TO 258                                                        PLNFM
  256 AR(J,I)=0.0                                                      PLNFM
  258 CONTINUE                                                         PLNFM
      IF(YL.LT.Y) GO TO 280                                            PLNFM
      IF(ASNG) 265,270,270                                             PLNFM
  265 AR(J,I)=0.0                                                      PLNFM
      GO TO 250                                                        PLNFM
  270 CONTINUE                                                         PLNFM
      AR(J,I)=1.0                                                      PLNFM
      SAM=SAM+AR(J,I)                                                  PLNFM
      GO TO 250                                                        PLNFM
```

```
  280 IF(Y-YL+ERR2.GE.D) GO TO 290                                      PLNFM     71
      AR(J,I)=AR(J,I)+ASNG*(YL-Y+D)/D                                   PLNFM     72
      X1=XL                                                             PLNFM     73
      Y1=YL                                                             PLNFM     74
      IF(ICHQ.NE.1) GO TO 310                                           PLNFM     75
      IF(JK) 284,285,285                                                PLNFM     76
  284 IF(YE2.LE.Y-D) GO TO 320                                          PLNFM     77
      GO TO 286                                                         PLNFM     78
  285 IF(YE2.GE.Y) GO TO 320                                            PLNFM     79
  286 CONTINUE                                                          PLNFM     80
      IF(XE1.GT.XL) GO TO 400                                           PLNFM     81
      IEDZ=2                                                            PLNFM     82
      GO TO 400                                                         PLNFM     83
  290 X1=X2                                                             PLNFM     84
      Y1=Y2                                                             PLNFM     85
  295 IEDZ=0                                                            PLNFM     86
      IF(ICHQ.NE.1) GO TO 310                                           PLNFM     87
      YR=YE2                                                            PLNFM     88
      GO TO 315                                                         PLNFM     89
  310 YR=YE1+TNG*(XR-XE1)                                               PLNFM     90
  315 IF(JK) 317,316,316                                                PLNFM     91
  316 IF(YR.GE.Y) GO TO 320                                             PLNFM     92
      GO TO 318                                                         PLNFM     93
  317 IF(YR.LE.Y-D) GO TO 320                                           PLNFM     94
  318 CONTINUE                                                          PLNFM     95
      IF(ICHQ.EQ.1) GO TO 325                                           PLNFM     96
      IFIN=1                                                            PLNFM     97
      X2=XR                                                             PLNFM     98
      Y2=YR                                                             PLNFM     99
      GO TO 330                                                         PLNFM    100
  320 JFIN=1                                                            PLNFM    101
      IF(JK) 322,321,321                                                PLNFM    102
  321 Y2=Y                                                              PLNFM    103
      GO TO 323                                                         PLNFM    104
  322 Y2=Y-D                                                            PLNFM    105
  323 X2=XE1+(Y2-YE1)/TNG                                               PLNFM    106
      IF(ABS(X2-XR).GT.ERRR) GO TO 330                                  PLNFM    107
      IFIN=1                                                            PLNFM    108
      JFIN=0                                                            PLNFM    109
      GO TO 330                                                         PLNFM    110
  325 IEDZ=1                                                            PLNFM    111
      X2=XE2                                                            PLNFM    112
      Y2=YE2                                                            PLNFM    113
  330 AR(J,I)=AR(J,I)+0.5*ASNG*(2.*XR-X2-X1)*(Y2-Y1)/DZ                 PLNFM    114
C                                                                       PLNFM    115
C                                                                       PLNFM    116
      IF(ABS(TNG).LT.ERRR) TNG=1.E-20                                   PLNFM    117
 1331 CONTINUE                                                          PLNFM    118
      IF(JK) 332,331,331                                                PLNFM    119
  331 IF(J.LT.JLT) GO TO 334                                            PLNFM    120
      XLT=XE1+(YY(JLT)-YE1)/TNG                                         PLNFM    121
      IF(XLT.GT.XE2.AND.ASNG.GT.0.0) GO TO 334                          PLNFM    122
      IF(YEDG(1).GT.0.0.AND.YY(JLT).LE.YL.AND.ASNG.GT.0.0) XLT=0.0      PLNFM    123
      IF(YY(JLT).GE.YTDG(NSTP)) XLT=XTDG(NSTP)                          PLNFM    124
      IF (ASNG.GT.0.0) XLE(JLT)=XLT                                     PLNFM    125
      IF (ASNG.LT.0.0) XTE(JLT)=XLT                                     PLNFM    126
      GO TO 333                                                         PLNFM    127
  332 IF(J.GT.JLT) GO TO 334                                            PLNFM    128
      XLT=XE1+(YY(JLT)-YE1)/TNG                                         PLNFM    129
      IF(YY(JLT).GE.YTDG(NSTP)) XLT=XTDG(NSTP)                          PLNFM    130
      XTE(JLT)=XLT                                                      PLNFM    131
  333 JLT=JLT+JK                                                        PLNFM    132
      GO TO 1331                                                        PLNFM    133
  334 CONTINUE                                                          PLNFM    134
C                                                                       PLNFM    135
      IF(IEDZ.EQ.1) GO TO 400                                           PLNFM    136
      SAM=SAM+AR(J,I)                                                   PLNFM    137
      IF(JK) 344,342,342                                                PLNFM    138
  342 CONTINUE                                                          PLNFM    139
      IF(JFIN.EQ.1) GO TO [illegible]                                   PLNFM    140
```

```
      GO TO 350                                                         PLNFM 141
  344 IF(JFIN.EQ.1) GO TO 495                                           PLNFM 142
  350 IF(ASNG) 355,360,360                                              PLNFM 143
  355 IF(IFIN.EQ.1) GO TO 478                                           PLNFM 144
      GO TO 530                                                         PLNFM 145
  360 IF(IFIN.EQ.1) GO TO 465                                           PLNFM 146
      GO TO 500                                                         PLNFM 147
C                                                                       PLNFM 148
C     LEADING EDGE AND TRAILING EDGE COMPUTATION                        PLNFM 149
C                                                                       PLNFM 150
  400 IF(ASNG) 440,405,405                                              PLNFM 151
  405 IF(JK) 430,410,410                                                PLNFM 152
  410 K1=K1+JK                                                          PLNFM 153
      IF(K1.GE.NSP) GO TO 420                                           PLNFM 154
      XE1=XEDG(K1)                                                      PLNFM 155
      YE1=YEDG(K1)                                                      PLNFM 156
      K=K1+JK                                                           PLNFM 157
      XE2=XEDG(K)                                                       PLNFM 158
      YE2=YEDG(K)                                                       PLNFM 159
      GO TO 450                                                         PLNFM 160
  420 IF(JK.LT.0) GO TO 240                                             PLNFM 161
      JK=-1                                                             PLNFM 162
      JLT=J                                                             PLNFM 163
      MLC(2,I)=0                                                        PLNFM 164
      K2=K2-JK                                                          PLNFM 165
  430 K2=K2+JK                                                          PLNFM 166
      XE1=XTDG(K2)                                                      PLNFM 167
      YE1=YTDG(K2)                                                      PLNFM 168
      K=K2+JK                                                           PLNFM 169
      XE2=XTDG(K)                                                       PLNFM 170
      YE2=YTDG(K)                                                       PLNFM 171
      GO TO 450                                                         PLNFM 172
  440 K2=K2+1                                                           PLNFM 173
      XE1=XTDG(K2)                                                      PLNFM 174
      YE1=YTDG(K2)                                                      PLNFM 175
      XE2=XTDG(K2+1)                                                    PLNFM 176
      YE2=YTDG(K2+1)                                                    PLNFM 177
  450 G=XE2-XE1                                                         PLNFM 178
      F=YE2-YE1                                                         PLNFM 179
      TNG=F/(G+1.E-20)                                                  PLNFM 180
      IF(G.LT.0.) GO TO 650                                             PLNFM 181
      AREA=AREA+ASNG*G*(YE2+YE1)                                        PLNFM 182
C                                                                       PLNFM 183
      ICHQ=0                                                            PLNFM 184
      IF(XR.GT.XE2) ICHQ=1                                              PLNFM 185
      IF(IEDZ.EQ.1) GO TO 290                                           PLNFM 186
      IF(IEDZ.EQ.2) GO TO 295                                           PLNFM 187
      GO TO 250                                                         PLNFM 188
C                                                                       PLNFM 189
C     NUMBER OF BOXES IN SPANWISE COLUMN                                PLNFM 190
C     WAKE TYPE AND ORDER OF WAKE BOX                                   PLNFM 191
C                                                                       PLNFM 192
  465 IF(ASNG) 478,467,467                                              PLNFM 193
  467 IF(JK) 476,469,469                                                PLNFM 194
  469 JJ=0                                                              PLNFM 195
      IF (AR(J,I).LT.ERR2) JJ=1                                         PLNFM 196
      ML(I)=J-JJ                                                        PLNFM 197
C                                                                       PLNFM 198
      JJ=ML(I)                                                          PLNFM 199
      DO 471 K=1,JJ                                                     PLNFM 200
      KK=K                                                              PLNFM 201
      IF (AR(K,I).LT.AWAK) GO TO 472                                    PLNFM 202
  471 CONTINUE                                                          PLNFM 203
      MLC(2,I)=KK                                                       PLNFM 204
      GO TO 473                                                         PLNFM 205
  472 MLC(2,I)=KK-1                                                     PLNFM 206
  473 CONTINUE                                                          PLNFM 207
C     ADJUSTMENT OF FIRST BOX AT LEADING EDGE ALONG KK-TH CHORDWISE ROW PLNFM 208
      KK=MLC(2,I)                                                       PLNFM 209
      IF (XX(I).GE.XLE(KK)) GO TO 474                                   PLNFM 210
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      MLC(2,I)=MLC(2,I)-1
      GO TO 473
  474 CONTINUE
C
      GO TO 500
  476 MLS=-1
      ML(I)=ML(I-1)
      K=ML(I)
      GO TO 480
  478 MLS=1
      K=J
  480 DO 490 JJ=1,K
      IF (AR(JJ,I)) 481,484,484
  481 IF (ABS(AR(JJ,I))-ERR2) 483,483,482
C
  482 WRITE(IW,96) ERR2,JJ,I,AR(JJ,I)
C
  483 AR(JJ,I)=0.0
  484 CONTINUE
      IF (MLW(I).NE.0) GO TO 490
      IF(ASNG) 487,486,486
C     WING TIP WAKE
  486 IF (AR(JJ,I).GE.AWAK) GO TO 490
      MLW(I)=JJ
      GO TO 488
C     WING ROOT WAKE
  487 IF (AR(JJ,I).LT.AWAK) GO TO 490
      MLW(I)=JJ-1
  488 MLW(I)=MLW(I)*MLS
  490 CONTINUE
      GO TO 500
C
C     TRAILING EDGE BOX AREA ADJUSTMENT FOR WING TIP WAKE
C
  495 CONTINUE
      J=J-1
      K=K-J
      SPAN=0
      SAM=SAM-AR(J,I)
      GO TO 290
C
C     END OF A SPANWISE COLUMN
C
  500 CONTINUE
      IF(ASN .GT.0.0) GO TO 516
C     ADJUST WING TIP BOX FOR WING ROOT WAKE
      JJ=ML(I)
  510 IF (AR(JJ,I).LT.AWAK) GO TO 512
      MLC(2,I)=JJ
      GO TO 516
  512 JJ=JJ-1
      IF(JJ) 514,514,510
  514 MLC(2,I)=0
      IF (MLW(I).EQ.0) GO TO 516
      WRITE(IW,155) I,1,MLW(I),I,MLC(2,I)
      STOP
  516 CONTINUE
      SUM=SUM+SAM
C
  530 CONTINUE
C
      IF(ASNG.LT.0.0) GO TO 700
      IF(XTOG(1).GE.XTOG(NSTP)) GO TO 700
C
C     INITIALIZATION FOR WING ROOT WAKE
C
      K2=0
      XR=0.0
      YR=0.0
      XE2=XTOG(1)
```

```
      YE2=YTDG(1)
      X2=XE2
      Y2=YE2
      ASNG=-1.0
      JLT=1
      GO TO 220
C     ERROR MESSAGE
  650 K=450
      A=G
      Z=Z2
      WRITE(IW,150) K,Z,A
      STOP
C
  700 CONTINUE
      IF(JRITE.EQ.0) GO TO 760
      WRITE(IW,65)
      IF(NEW.EQ.1) WRITE(IW,66)
      IF(NEW.EQ.2) WRITE(IW,67)
      DO 750 I=1,L
      JL=ML(I)
      K=JL/5
      IF(JL.LE.5*K) GO TO 710
      K=K+1
  710 WRITE(IW,70) I
      WRITE(IW,72) MLC(2,I)
      IF (MLW(I)) 715,725,720
  715 WRITE(IW,75) MLW(I)
      GO TO 725
  720 WRITE(IW,80) MLW(I)
  725 DO 740 JJ=1,K
  740 WRITE(IW,85) (J,AR(J,I),J=JJ,JL,K)
  750 CONTINUE
C
      JL=ML(L)
      WRITE(6,140)
      WRITE(IW,145) (J,XLE(J),XTE(J),J=1,JL)
C
      SUM=2.*SUM*DZ
      WRITE(IW,90) AREA,SUM
  760 RETURN
   65 FORMAT(1H1,20X,37HNON-DIMENSIONAL BOX-AREA DISTRIBUTION)
   66 FORMAT(1H+,57X,21H - ( PHYSICAL   WING)//)
   67 FORMAT(1H+,57X,21H - (TRANSFORMED WING)//)
   70 FORMAT(//2X,I2,19H-TH SPANWISE COLUMN/)
   72 FORMAT(12X,16HFIRST L.E. BOX =,I3,22H-TH BOX FROM WING ROOT)
   75 FORMAT(12X,16HFIRST WAKE BOX =,I3,22H-TH BOX FROM WING ROOT/)
   80 FORMAT(12X,16HLAST  WAKE BOX =,I3,22H-TH BOX FROM WING ROOT/)
   85 FORMAT(5X,5(1X,I3,1PE11.4))
   90 FORMAT(//10X,25HWING AREA CALCULATED FROM/
     1         15X,26HLEADING AND TRAILING EDGES =,1PE11.4/
     2         15X,26HSUMMATION OF AREA OF BOXES =,1PE11.4)
   96 FORMAT(//10X,43HNEGATIVE BOX AREA EXCEEDS ALLOWABLE LIMIT (,
     1         1PE11.4,1H)/10X,3HAR(,I3,1H,,I3,1H,,6HNEW) =,1PE11.4,
     2         5X,21HCOMPUTATION CONTINUES//)
  140 FORMAT(1H1,9X,54HLEADING AND TRAILING EDGE POINTS AT EACH CHORDWIS
     SE ROW/13X,1HJ,6X,7HLEADING,8X,8HTRAILING/)
  145 FORMAT(10X,I3,1P2E15.5)
  150 FORMAT(1H0,4X,29HPLNFM--NEGATIVE VALUE NEAR SN,I4,2X,A2,1PE11.4)
  155 FORMAT(1H0,4X,25HORDER OF WING TIP BOX IN ,I2,43H-TH SPANWISE COLU
     1MN IS NOT PROPERLY DEFINED//
     2           15X,8HMLW(NEW,,I2,3H) =,I3/15X,8HMLL(NEW,,I2,3H) =,I3)
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE POT2(M2,MO,NO,CK,D,A)
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     MO,NO CONTROL THE NUMBER OF VALUES COMPUTED
C
C     M2 IS THE RANCE 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)
      M=MO
      N=NO
      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-DM
      B3=-0.5*B5
      D3=DM*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
      KI=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-DM/CN*B3
    9 B3= COS(A2)/CN
      B4= SIN(A2)/CN
      C3=B1*B3+B2*B4
      C4=B2*B3-B1*B4
      B5=DM*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(SFDX,SFDY,SFDH,KSFD,SFMX,SFMY,SFMH,KSFM,DA,T
     $                   ,XX,YY,IW,L,MLC,M,MD,NB,NEW,NM,KSFS,IPRINT)
      DIMENSION SFDX(NB,1),SFDY(NB,MD),SFDH(NM,MD),KSFD(1)
     1          ,SFMX(1),SFMY(1),SFMH(1),DA(1),T(NM,1),MLC(2,1)
     2          ,XX(1),YY(1)
C
      ERRR=1.E-06
      IRITE=IPRINT/10
      JRITE=IPRINT/100
      IF(NEW.EQ.1) IRITE=IPRINT-10*IRITE
      KSFD(M)=DA(98)
      NP=KSFD(M)
      IF (NP) 730,170,100
C
C     A SPLINE-SURFACE FOR THE DEFLECTION IS FITTED TO VALUES
C     OF DEFLECTION AT GIVEN POINTS.
C
  100 IF (NB-NP) 730,120,120
  120 CONTINUE
      KP=100
      DO 140 IP=1,NP
      SFDX(IP,M)=DA(KP+1)/DA(24)
      SFDY(IP,M)=DA(KP+2)/DA(24)
      SFDH(IP,M)=DA(KP+3)
  140 KP=KP+4
C     SPLINE-SURFACE FIT DATA
C
  145 CONTINUE
      IF(IRITE) 150,160,150
  150 WRITE(IW,30) M
      WRITE(IW,10)
C
  160 CONTINUE
      CALL SURFI(NM,KSFD(M),T,SFDX(1,M),SFDY(1,M),SFDH(1,M),IRITE)
      GO TO 200
C     PRESENTLY FOR PITCH AND PLUNGE OF FORM  Z=A0+A1*X
  170 SFDH(1,M)=DA(52)
      SFDH(2,M)=DA(53)*DA(24)
      SFDH(3,M)=0.0
      GO TO 145
  200 ICHECK=0
      RETURN
  730 IPR=98
      GO TO 750
  750 WRITE
     2  (IW,20)IPR
      STOP
   10 FORMAT(1H0,10X,73HCOMPUTED DEFLECTION = A0+A1*X+A2*Y+ SUM OF H(I)*
     1(R(I)**2)*(ALOG(R(I)**2)))
   20 FORMAT(1H0,10X,16HDRED -- BAD DATA,I5)
   30 FORMAT(1H0,8X,26HPHYSICAL PLANE -- MODE NO.,I3)
      END
```

```
      SUBROUTINE WVAL(XX,YY,XTDG,YTDG,XTE,SFDX,SFDY,SFDH,KSFD,S,D,CK
     $                ,IW,L,ML,MI,MEW,MST,MB,MD,NB,NH,KSFS,IPRINT)
C     CALCULATES DOWNWASH VELOCITY DISTRIBUTION (REAL AND IMAGINARY)
      DIMENSION XX(1),YY(1),XTDG(1),YTDG(1),XTE(1),S(2,MB,1),ML(1)
     $          ,SFDX(MB,1),SFDY(MB,MD),SFDH(MH,MD),KSFD(1)
      IRITE=IPRINT/10
      IF(MEW.EQ.1) IRITE=IPRINT-10*IRITE
      JLP=KSFD(MI)
      IF(KSFD(MI).EQ.0.AND.MEW.EQ.2) JLP=KSFS
      JL=ML(L)
      DO 5 I=1,L
      DO 5 J=1,JL
      DO 5 K=1,2
    5 S(K,J,I)=0.0
      DO 80 I=1,L
      JL=ML(I)
      IF(JL) 40,80,40
   40 CONTINUE
      DO 70 J=1,JL
      IF (XX(I).GT.(XTE(J)+D)) GO TO 70
      CALL SURF2(XX(I),YY(J),1,1,1,VALU,VALUD,DM,SFDX(1,MI)
     $           ,SFDY(1,MI),SFDH(1,MI),JLP,2)
      S(1,J,I)=VALUE
      S(2,J,I)=CK*VALU
   70 CONTINUE
   80 CONTINUE
      IF(IRITE) 100,200,100
  100 WRITE  (IW,10)
      WRITE(IW,12) MI
   10 FORMAT(1H1,10X,47HUPWASH (REAL, IMAGINARY, ABSOLUTE, PHASE ANGLE))
   12 FORMAT(1H+,58X,28H---PHYSICAL PLANE---MODE NO.,I3/)
      DO 170 I=1,L
      JL=ML(I)
      IF (JL) 110,170,110
  110 WRITE(IW,20) I
      JLP=JL/2
      IF(JL-2*JLP.NE.0) JLP=JLP+1
      DO 150 J=1,JLP
      S1=SQRT(S(1,J,I)*S(1,J,I)+S(2,J,I)*S(2,J,I))
      IF(S1.GT.0.0) GO TO 120
      S2=0.0
      GO TO 130
  120 S2=57.29578*ATAN2(S(2,J,I),S(1,J,I))
  130 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))
      IF(S3.GT.0.0) GO TO 155
      S4=0.0
      GO TO 160
  155 S4=57.29578*ATAN2(S(2,J1,I),S(1,J1,I))
  160 WRITE(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
C
  200 CONTINUE
      RETURN
   20 FORMAT(1H0,5X,I2,19H-TH SPANWISE COLUMN)
   25 FORMAT(1H ,5X,2(2X,I3,1P4E13.5))
      END
```

```
      SUBROUTINE BOXP(XX,YY,XTDG,YTDG,XTE,XLE,A,AR,DAN,T,EDG,S,CK,D,YE
     1                 ,IPRINT,MLT,MLC,AMA
     2                 ,JMAX,L,ML,MLW,M1,NEW,NST,IW,MB,MC,ME,MD)
C     ICHECK=0 - POTENTIAL AS COMPUTED IN  BOXPO
C     ICHECK=1 -WITH LEADING EDGE CORRECTION
C     ICHECK=2 - PRESSURE COEFFICIENT COMPUTED FROM CORRECTED POTENTIAL
C     ICHECK=3 - PRESSURE COEFFICIENT COMPUTED FROM TRANSFORMED VALUE
C                FOR PHYSICAL WING
      DIMENSION XX(1),YY(1),XTDG(1),YTDG(1),XTE(1),XLE(1),A(2,MC,MB)
     1          ,AR(MB,1),EDG(1),S(2,MB,MB),T(ME,ME,1)
     2          ,ML(1),MLW(1),MLT(2,1),MLC(2,MB),AMA(MB,MB)
      IPRIN=IPRINT
      KRITE=IPRIN/10000
      IPRIN=IPRIN-10000*KRITE
      JRITE=IPRIN/100
      IPRIN=IPRIN-100*JRITE
      IRITE=IPRIN
      IF(NEW.EQ.1) GO TO 140
      KRITE=KRITE/10
      JRITE=JRITE/10
      IRITE=IRITE/10
      GO TO 150
  140 KRITE=KRITE-(KRITE/10)*10
      JRITE=JRITE-(JRITE/10)*10
      IRITE=IRITE-(IRITE/10)*10
  150 CONTINUE
      NSMOS=0
      DH=0.5*D
C
      CALL     BOXPO(XX,YY,XTDG,YTDG,XTE,A,AR,DAN,T,S,CK,D,AMA
     $              ,IW,L,MB,MC,ML,MLC,MLW,M1,NEW,NST,JRITE)
C
C     BOXPO COMPUTES THE POTENTIAL VALUES IN EACH BOX.
C     THEY ARE STORED IN THE ARRAY S.
C
      ICHECK=0
      NEW=NEW
      JMAK=JMAX
      IF(IRITE.EQ.0) GO TO 270
      WRITE(IW,30)
  200 DO 205 I=1,L
      DO 205 J=1,JMAK
      DO 205 K=1,2
  205 T(J,I,K)=S(K,J,I)
  215 WRITE(IW,32) M1
   30 FORMAT(1H1,10X,61HPOTENTIAL CALCULATED (REAL, IMAGINARY, ABSOLUTE,
     $ PHASE ANGLE))
   32 FORMAT(1H+,71X,28H---PHYSICAL PLANE---MODE NO.,I3/)
C     PRINT-OUT
  218 DO 250 I=1,L
      JL=ML(I)
      IF (JL) 220,250,220
  220 WRITE(IW,20) I
      JLP=JL/2
      IF(JL-2*JLP.NE.0) JLP=JLP+1
      DO 240 J=1,JLP
      S1=SQRT(T(J,I,1)*T(J,I,1)+T(J,I,2)*T(J,I,2))
      IF(S1.NE.0.0) GO TO 224
      S2=0.0
      GO TO 226
  224 CONTINUE
      S2=57.29578*ATAN2(T(J,I,2),T(J,I,1))
  226 CONTINUE
      J1=J+JLP
      IF(J1.LE.JL) GO TO 230
      J1=0
      S3=0.0
      S4=0.
      T(J1,I,1)=0.0
```

```
      T(J1,I,2)=0.0                                                     BOXP  71
      GO TO 240                                                         BOXP  72
  230 CONTINUE                                                          BOXP  73
      S3=SQRT(T(J1,I,1)*T(J1,I,1)+T(J1,I,2)*T(J1,I,2))                  BOXP  74
      IF(S3.NE.0.0) GO TO 234                                           BOXP  75
      S4=0.0                                                            BOXP  76
      GO TO 236                                                         BOXP  77
  234 CONTINUE                                                          BOXP  78
      S4=57.29577951*ATAN2(T(J1,I,2),T(J1,I,1))                         BOXP  79
  236 CONTINUE                                                          BOXP  80
  240 WRITE(IW,25) J,T(J,I,1),T(J,I,2),S1,S2                            BOXP  81
     *              ,J1,T(J1,I,1),T(J1,I,2),S3,S4                       BOXP  82
  250 CONTINUE                                                          BOXP  83
C                                                                       BOXP  84
  270 CONTINUE                                                          BOXP  85
      IF(ICHECK.GE.1) GO TO 400                                         BOXP  86
C     LEADING EDGE CORRECTION                                           BOXP  87
      DO 380 J=1,JMAK                                                   BOXP  88
      IF (MLT(1,J).EQ.0) GO TO 285                                      BOXP  89
      I1=MLT(1,J)                                                       BOXP  90
      I2=MLT(2,J)                                                       BOXP  91
      S3=0.0                                                            BOXP  92
      DO 280 I=I1,I2                                                    BOXP  93
      K=I                                                               BOXP  94
      IF (AR(J,I).GE.1.0) GO TO 290                                     BOXP  95
  280 CONTINUE                                                          BOXP  96
  285 S3=0                                                              BOXP  97
  290 IF(K.EQ.1) S3=0                                                   BOXP  98
      S3=XX(K)-XLE(J)-DH+S3                                             BOXP  99
      DO 370 I=1,L                                                      BOXP 100
      S1=(XX(I)-XLE(J))/S3                                              BOXP 101
      IF(S1) 300,340,310                                                BOXP 102
  300 S1=0.0                                                            BOXP 103
      GO TO 340                                                         BOXP 104
  310 IF(YE) 320,320,330                                                BOXP 105
  320 S1=S1*(XX(I)+XLE(J))/(S3+2.*XLE(J))                               BOXP 106
  330 IF(S1.GT.1.0) S1=1.0                                              BOXP 107
      S1=EDG(J)*SQRT(S1)                                                BOXP 108
  340 DO 350 K=1,2                                                      BOXP 109
  350 S(K,J,I)=S1*S(K,J,I)                                              BOXP 110
  370 CONTINUE                                                          BOXP 111
  380 CONTINUE                                                          BOXP 112
      ICHECK=1                                                          BOXP 113
      IF(IRITE) 390,400,390                                             BOXP 114
  390 WRITE(IW,36)                                                      BOXP 115
   36 FORMAT(1H1,10X,61HPOTENTIAL CORRECTED  (REAL, IMAGINARY, ABSOLUTE, BOXP 116
     $ PHASE ANGLE))                                                    BOXP 117
C     TRANSFER TO THE PRINT-OUT OF CORRECTED VELOCITY POTENTIAL         BOXP 118
      GO TO 200                                                         BOXP 119
  400 CONTINUE                                                          BOXP 120
      IF(KRITE.EQ.0) GO TO 700                                          BOXP 121
      IF(ICHECK.GE.2) GO TO 520                                         BOXP 122
C     CALCULATE AND PRINT PRESSURE COEFFICIENT                          BOXP 123
      ICHECK=2                                                          BOXP 124
      DO 410 I=1,L                                                      BOXP 125
      JL=ML(I)                                                          BOXP 126
      DO 410 J=1,JL                                                     BOXP 127
      T(J,I,3)=S(1,J,I)                                                 BOXP 128
  410 T(J,I,4)=S(2,J,I)                                                 BOXP 129
  420 JMAK=JMAX                                                         BOXP 130
      DO 430 I=1,L                                                      BOXP 131
      JL=ML(I)                                                          BOXP 132
      DO 430 J=1,JL                                                     BOXP 133
      DO 430 K=1,2                                                      BOXP 134
  430 T(J,I,K)=0.0                                                      BOXP 135
      DO 500 J=1,JMAK                                                   BOXP 136
      I1=MLT(1,J)                                                       BOXP 137
      I2=MLT(2,J)-MLT(1,J)+1                                            BOXP 138
      II=I1                                                             BOXP 139
      IK=I2+1                                                           BOXP 140
```

```
      T(1, 6,5)=XLE(J)                                                  BOXP     141
      T(1, 8,5)=XLE(J)                                                  BOXP     142
      T(1,10,5)=0.0                                                     BOXP     143
      T(1,12,5)=0.0                                                     BOXP     144
      IF (ABS(XLE(J)-XX(II)).GT.1.E-05) GO TO 445                       BOXP     145
C     ADJUSTMENT -- LEADING EDGE AND FIRST BOX COINCIDES                BOXP     146
      II=II+1                                                           BOXP     147
      IK=IK-1                                                           BOXP     148
  445 CONTINUE                                                          BOXP     149
      DO 450 I=2,IK                                                     BOXP     150
      T(I, 6,5)=XX(II)                                                  BOXP     151
      T(I, 8,5)=XX(II)                                                  BOXP     152
      T(I,10,5)=T(J,II,3)                                               BOXP     153
      T(I,12,5)=T(J,II,4)                                               BOXP     154
      T(I-1,1,5)=XX(II)                                                 BOXP     155
  450 II=II+1                                                           BOXP     156
      CALL SPLN1(I2,T(1, 1,5),T(1, 2,5),T(1, 4,5)                       BOXP     157
     $           ,IK,T(1, 6,5),T(1,10,5),T(1,14,5),NSMOS)               BOXP     158
      CALL SPLN1(I2,T(1, 1,5),T(1, 3,5),T(1, 5,5)                       BOXP     159
     $           ,IK,T(1, 8,5),T(1,12,5),T(1,14,5),NSMOS)               BOXP     160
      II=I1                                                             BOXP     161
      DO 470 I=1,I2                                                     BOXP     162
      T(J,II,1)=T(I,4,5)-CK*T(I,3,5)                                    BOXP     163
      T(J,II,2)=T(I,5,5)+CK*T(I,2,5)                                    BOXP     164
  470 II=II+1                                                           BOXP     165
  500 CONTINUE                                                          BOXP     166
      WRITE(IW,40)                                                      BOXP     167
   40 FORMAT(1H1,10X,61HPRESSURE COEFFICIENT (REAL, IMAGINARY, ABSOLUTE, BOXP     168
     $ PHASE ANGLE))                                                    BOXP     169
C     TRANSFER TO PRINT-OUT SECTION                                     BOXP     170
      GO TO 215                                                         BOXP     171
  520 CONTINUE                                                          BOXP     172
  700 CONTINUE                                                          BOXP     173
      RETURN                                                            BOXP     174
   20 FORMAT(1H0,5X,I2,19H-TH SPANWISE COLUMN)                          BOXP     175
   25 FORMAT(1H ,5X,2(2X,I3,1P4E13.5))                                  BOXP     176
      END                                                               BOXP     177
```

```
      SUBROUTINE BOXPO(XX,YY,XTDG,YTDG,XTE,A,AR,DAN,T,S,CK,D,AMA        BOXPO   2
     $                 ,IW,L,MB,MC,ML,MLC,MLW,MI,NEW,NST,JRITE)         BOXPO   3
C     SOLUTION OF SIMULTANEOUS EQUATIONS FOR THE POTENTIAL              BOXPO   4
      DIMENSION XX(1),YY(1),XTDG(1),YTDG(1),XTE(1),A(2,MC,1)            BOXPO   5
     1          ,AR(MB,MB),T(2,MB,1),S(2,MB,1),AMA(MB,MB)               BOXPO   6
     2          ,ML(1),MLC(2,1),MLW(1)                                  BOXPO   7
      DH=0.5*D                                                          BOXPO   8
      DD=2.0*D                                                          BOXPO   9
      LIM2=2*ML(L)                                                      BOXPO  10
      IF(JRITE.EQ.0) GO TO 25                                           BOXPO  11
      IF(NEW.EQ.2.OR.MI.GT.1) GO TO 25                                  BOXPO  12
C     PRINT INFLUENCE COEFFICIENT                                       BOXPO  13
      WRITE(IW,100)                                                     BOXPO  14
      JL=2*ML(L)                                                        BOXPO  15
      K=JL/4                                                            BOXPO  16
      IF(JL-4*K.NE.0) K=K+1                                             BOXPO  17
      DO 20 I=1,L                                                       BOXPO  18
      I1=I-1                                                            BOXPO  19
      WRITE(IW,110) I1                                                  BOXPO  20
      DO 15 J1=1,K                                                      BOXPO  21
   15 WRITE(IW,120) ((J,A(1,J,I),A(2,J,I)),J=J1,JL,K)                   BOXPO  22
   20 CONTINUE                                                          BOXPO  23
   25 CONTINUE                                                          BOXPO  24
      I1=NST                                                            BOXPO  25
      AMI=1.0                                                           BOXPO  26
      DO 26 I=1,I1                                                      BOXPO  27
   26 AMI=AMIN1(XTDG(I),AMI)                                            BOXPO  28
C                                                                       BOXPO  29
      IF(JRITE.EQ.0) GO TO 28                                           BOXPO  30
      WRITE(IW,168)                                                     BOXPO  31
   28 CONTINUE                                                          BOXPO  32
      II=0                                                              BOXPO  33
      NFLNS = DAN                                                       BOXPO  34
      DO 90 I=1,L                                                       BOXPO  35
      X=XX(I)                                                           BOXPO  36
C     ADJUST UPSTREAM INFLUENCE                                         BOXPO  37
      KO = 1                                                            BOXPO  38
      IF(NFLNS.EQ.0) GO TO 30                                           BOXPO  39
      KO = MAX0(1,I-NFLNS+1)                                            BOXPO  40
   30 CONTINUE                                                          BOXPO  41
      JL=ML(I)                                                          BOXPO  42
      IF (JL.EQ.0) GO TO 90                                             BOXPO  43
C     DEFINE WING AND WAKE BOXES                                        BOXPO  44
C                                            JW=NO. OF WAKE BOXES IN ROW BOXPO  45
C                                            JE=NO. OF WING BOXES       BOXPO  46
C                                            JS=ORDER OF FIRST WING BOX BOXPO  47
C                                            JN=ORDER OF LAST  WING BOX BOXPO  48
C                                            JSW=ORDER OF FIRST WAKE BOX BOXPO  49
C                                            JNW=ORDER OF LAST  WAKE BOX BOXPO  50
      JW=0                                                              BOXPO  51
      IF(X.LE.AMI) GO TO 34                                             BOXPO  52
      J=1                                                               BOXPO  53
   32 IF(J.GT.JL) GO TO 34                                              BOXPO  54
      IF (X.LE.(XTE(J)+DH)) GO TO 33                                    BOXPO  55
      JW=JW+1                                                           BOXPO  56
   33 J=J+1                                                             BOXPO  57
      GO TO 32                                                          BOXPO  58
   34 JE=JL-JW                                                          BOXPO  59
      IF (MLW(I)) 36,35,35                                              BOXPO  60
   35 JS =JW+1                                                          BOXPO  61
      JSW=1                                                             BOXPO  62
      GO TO 37                                                          BOXPO  63
   36 JS =1                                                             BOXPO  64
      JSW=JE +1                                                         BOXPO  65
   37 JN =JS +JE-1                                                      BOXPO  66
      JNW=JSW+JW-1                                                      BOXPO  67
      IF (I1.EQ.0) GO TO 50                                             BOXPO  68
C     SUBTRACTION OF CONTRIBUTIONS OF PRECEDING ROWS TO UPWASH          BOXPO  69
      DO 47 J=JS,JN                                                     BOXPO  70
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DO 45 K=K0,I1
      KL=ML(K)
      K1=I+1-K
      IF (KL.EQ.0) GO TO 45
      DO 40 N=1,KL
      IF (NEW.EQ.2) GO TO 38
      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)
      GO TO 39
   38 CONTINUE
      TEMP=(AMA(J,I)+AMA(N,K))/DD
      YM1=(YY(J)+YY(N))*TEMP
      YM2=(YY(J)-YY(N))*TEMP
      YM2=ABS(YM2)
      N1=YM1
      N2=YM2
      YM1=YM1-FLOAT(N1)
      YM2=YM2-FLOAT(N2)
      N1=N1+1
      N2=N2+1
      IF (N1.GT.LIM2) GO TO 138
      ARN10=A(1,N1,K1)
      AIN10=A(2,N1,K1)
      IF (N1+1.GT.LIM2) GO TO 238
      ARN11=A(1,N1+1,K1)
      AIN11=A(2,N1+1,K1)
      GO TO 338
  138 ARN10=A(1,LIM2,K1)
      AIN10=A(2,LIM2,K1)
  238 ARN11=A(1,LIM2,K1)
      AIN11=A(2,LIM2,K1)
  338 CONTINUE
      ARN20=A(1,N2,K1)
      AIN20=A(2,N2,K1)
      ARN21=A(1,N2+1,K1)
      AIN21=A(2,N2+1,K1)
      A1=ARN10+(ARN11-ARN10)*YM1+ARN20+(ARN21-ARN20)*YM2
      A2=AIN10+(AIN11-AIN10)*YM1+AIN20+(AIN21-AIN20)*YM2
      A1=A1*AMA(N,K)
      A2=A2*AMA(N,K)
   39 CONTINUE
      WT=1.0
      IF (N.GE.MLC(2,K).AND.MLC(2,K).NE.0) WT=AR(N,K)
      S(1,J,I)=S(1,J,I)-(A1*S(1,N,K)-A2*S(2,N,K))*WT
   40 S(2,J,I)=S(2,J,I)-(A2*S(1,N,K)+A1*S(2,N,K))*WT
   45 CONTINUE
   47 CONTINUE
C     SETTING UP MATRIX FOR SIMULTANEOUS EQUATIONS
   50 DO 52 J=1,JL
      DO 52 K=1,JL
      N1=J+K
      N2=IABS(J-K)+1
      WT=1.0
      IF (K.GE.MLC(2,I).AND.MLC(2,I).NE.0) WT=AR(K,I)
C
      IF(J.EQ.K) WT=1.0
C
      T(1,J,K)=(A(1,N1,1)+A(1,N2,1))*WT
      T(2,J,K)=(A(2,N1,1)+A(2,N2,1))*WT
   52 CONTINUE
C                                               SUBTRACTION OF CONTRIBUTION
C                                                FROM WAKE BOXES -- S(WING)=
C                                                S(WING)-T(WAKE)*PHI(WAKE)
      IF(JW.EQ.0) GC TO 60
      DO 56 J=JS,JN
      DO 55 N=JSW,JNW
      S(1,J,I)=S(1,J,I)-T(1,J,N)*S(1,N,I)+T(2,J,N)*S(2,N,I)
   55 S(2,J,I)=S(2,J,I)-T(1,J,N)*S(2,N,I)-T(2,J,N)*S(1,N,I)
```

```
   56 CONTINUE
      IF (MLW(I).LT.0) GO TO 60
C                                                 RE-POSITION ELEMENTS OF T
      DO 59 N=1,JE
      NW=N+JW
      DO 58 J=1,JE
      MW=J+JW
      DO 58 K=1,2
   58 T(K,J,N)=T(K,MW,NW)
   59 CONTINUE
C
C     SOLUTION OF EQUATIONS
   60 CONTINUE
C
      IF(JRITE.EQ.0) GO TO 72
      WRITE(IW,167) I
      DO 70 K=1,2
      N1=1
      N2=N1+3
   66 DO 67 J=1,JE
   67 WRITE(IW,169) (J,N,T(K,J,N),N=N1,N2)
      IF(N2.GE.JE) GO TO 68
      N1=N2+1
      N2=N1+3
      GO TO 66
   68 CONTINUE
      DO 69 J=1,JE
      WRITE(IW,170) K,J,I,S(K,J,I)
   69 CONTINUE
   70 CONTINUE
  167 FORMAT(1H0,5X,I2,14H-TH SPANWISE COLUMN)
  168 FORMAT(1H1,10X,64HCOEFFICIENT MATRIX (A) OF  (A)*(X)=(B)  FOR VELO
     $CITY POTENTIAL (X) ALONG EACH COLUMN//)
  169 FORMAT(8X,4(2X,1H(,I2,1H,,I2,2H) ,E13.6))
  170 FORMAT(9X,2HS(,I2,1H,,I2,1H,,I2,2H)=,E13.6)
   72 CONTINUE
C
      K = MSIMEC(NB,JE,I,T,S(1,JS,I))
      IF(K.NE.1) GO TO 95
C                                                 COMPUTE WAKE POTENTIALS --
C                                                 PHI(TE)*EXP(-IK*(X-XTE))
      IF(X.LE.(AMI-D)) GO TO 85
      JS=1
   75 Y=YY(JS)
      IF (X.LT.(XTE(JS)-DH)) GO TO 82
      IF (X.GE.(XTE(JS)+DH)) GO TO 82
      IF (X.GT.XTE(JS)) GO TO 76
      PTR=S(1,JS,I)
      PTI=S(2,JS,I)
      KK=1
      XB=X+D
      GO TO 77
   76 P=(X-XTE(JS))/D
      PTR=(1.0-P)*S(1,JS,I)+P*S(1,JS,I-1)
      PTI=(1.0-P)*S(2,JS,I)+P*S(2,JS,I-1)
      KK=0
      XB=X
   77 CONTINUE
   80 IF(XB.GT.1.0) GO TO 82
      XW=(XB-XTE(JS))*CK
      IKW=I+KK
      S(1,JS,IKW)=PTR*COS(XW)+PTI*SIN(XW)
      S(2,JS,IKW)=PTI*COS(XW)-PTR*SIN(XW)
      KK=KK+1
      XB=XB+D
      GO TO 80
   82 JS=JS+1
      IF(JS.LE.JL) GO TO 75
   85 CONTINUE
   90 II=II+1
```

```
      RETURN                                                            BOXPO     211
   95 WRITE (IW,140)                                                    BOXPO     212
      STOP                                                              BOXPO     213
  100 FORMAT(1H1,20X,42HINFLUENCE COEFFICIENT (REAL AND IMAGINARY)///)  BOXPO     214
  110 FORMAT(1H0,3X,I3,38H-BOX SEPARATION IN CHORDWISE DIRECTION)       BOXPO     215
  120 FORMAT(1H ,5X,4(I5,1P2E13.6))                                     BOXPO     216
  140 FORMAT(1H010X,59HSOLUTION OF SIMULTANEOUS EQUATIONS FOR THE POTENT BOXPO     217
     1IAL FAILED)                                                       BOXPO     218
      END                                                               BOXPO     219
```

```
      SUBROUTINE FORCI(XX,YY,S,SFDX,SFDY,SFDH,KSFD,KSFS,XLE,XTE,YMAX      FORCI
     1                ,AMA,JMAX         ,MLT,L,NEW,MZ,MB,MD,NB,NM,CK,T)   FORCI
      DIMENSION MLT(2,MB),XLE(MB),XTE(MB),XX(MB),YY(MB),S(2,MB,MB)        FORCI
     1           ,SFDX(NB,MD),SFDY(NB,MD),SFDH(NM,MD),KSFD(MD),T(NM,1)    FORCI
     2           ,AMA(MB,MB)                                              FORCI
C                                                                         FORCI
      N=0                                                                 FORCI
C     NO SMOOTHING IN SUBROUTINE SMOOTH. (SPLN1)                          FORCI
      D=XX(2)-XX(1)                                                       FORCI
      MZ=KSFD(MZ)                                                         FORCI
      IF(KSFD(MZ).EQ.0.AND.NEW.EQ.2) MZ=KSFS                              FORCI
      DO 150 J=1,JMAX                                                     FORCI
      MLT1=MLT(1,J)                                                       FORCI
      NZ=MLT(2,J)-MLT1+2                                                  FORCI
      NQ=NZ+1                                                             FORCI
      K1=1                                                                FORCI
      K2=4                                                                FORCI
      KC=0                                                                FORCI
      T(1,1)=XTE(J)                                                       FORCI
   40 KK=MLT1                                                             FORCI
      T(1, 2)=XLE(J)                                                      FORCI
      T( 1,K2)=0.0                                                        FORCI
      KA=2                                                                FORCI
      KB=NZ                                                               FORCI
      IF(K1.EQ.3.OR.NZ.EQ.2) GO TO 45                                     FORCI
      IF(KC.EQ.2) GO TO 45                                                FORCI
C     EXCLUDE LEADING EDGE POINT  IF IT IS TOO CLOSE TO  FIRST BOX        FORCI
      IF (XX(KK)-XLE(J).GT.0.2*D) GO TO 45                                FORCI
      KA=3                                                                FORCI
      KB=NZ-1                                                             FORCI
      KC=1                                                                FORCI
   45 CONTINUE                                                            FORCI
      DO 50 K=KA,KB                                                       FORCI
      T(K, 2)=XX(KK)                                                      FORCI
      IF(K1.EQ.3) GO TO 50                                                FORCI
      T(K,K2)=S(K1,J,KK)                                                  FORCI
   50 KK=KK+1                                                             FORCI
      IF(KC.EQ.2) GO TO 51                                                FORCI
      IF(K1.EQ.3) GO TO 60                                                FORCI
      CALL SPLN1(1,T(1,1),T(1,8),T(1,9),KB,T(1, 2),T(1,K2),T(1,10),N)     FORCI
      T(NQ,K2)=T(1,8)                                                     FORCI
      IF(KC.NE.1) GO TO 51                                                FORCI
      KC=2                                                                FORCI
      GO TO 40                                                            FORCI
   51 CONTINUE                                                            FORCI
C                                                                         FORCI
      IF(NZ.GT.2) GO TO 52                                                FORCI
C     ADJUSTMENT FOR TRAILING EDGE VELOCITY POTENTIAL                     FORCI
      KK=MLT(2,J)                                                         FORCI
      DUM=SQRT((XTE(J)-XLE(J))/(XX(KK)-XLE(J)))                           FORCI
      T(NQ,K2)=DUM*T(NZ,K2)                                               FORCI
   52 CONTINUE                                                            FORCI
C                                                                         FORCI
      KC=0                                                                FORCI
      IF(K1.EQ.2) GO TO 55                                                FORCI
      K1=2                                                                FORCI
      K2=6                                                                FORCI
      GO TO 40                                                            FORCI
   55 K1=3                                                                FORCI
      GO TO 40                                                            FORCI
   60 T(NQ, 2)=XTE(J)                                                     FORCI
      IF (ABS(XLE(J)-XX(MLT1)).GT.1.E-05) GO TO 67                        FORCI
C     LEADING EDGE AND FIRST BOX COINCIDE                                 FORCI
      NZ=NZ-1                                                             FORCI
      NQ=NQ-1                                                             FORCI
      DO 65 K=2,NQ                                                        FORCI
      T(K, 2)=T(K+1, 2)                                                   FORCI
      T(K, 4)=T(K+1, 4)                                                   FORCI
      T(K, 6)=T(K+1, 6)                                                   FORCI
```

```
      IF(NEW.EQ.1) GO TO 65
      T(K,12)=T(K+1,12)
   65 CONTINUE
   67 CONTINUE
C
C     INTERPOLATE DEFLECTION AT BOX CENTER, LEADING AND TRAILING EDGES
      CALL SURFZ(YY(J),T(1, 2),1,NQ,2,T(1,13),T(1,14),DUM,SFDX(1,M2)
     $           ,SFDY(1,M2),SFDM(1,M2),NZ,2)
C     PERFORM CHORDWISE INTEGRATION
      DO 75 K=1,NQ
      T(K,15)= T(K,14)*T(K,4)+CK*T(K,13)*T(K,6)
   75 T(K,16)= T(K,14)*T(K,6)-CK*T(K,13)*T(K,4)
      CALL INTGL(T(1,2),T(1,15),T(J,17),T(J,16),T(1,19),2,NQ,NZ,NM)
      DUM=T(NQ,13)
      T(J,17)=T(NQ,4)*DUM-T(J,17)
      T(J,18)=T(NQ,6)*DUM-T(J,16)
  150 CONTINUE
C     PERFORM SPANWISE INTEGRATION
      NQ=JMAK+2
      K1=17
      K2=15
      T(1,1)=0.0
  165 DO 170 K=1,JMAK
  170 T(K, 2)=YY(K)
      NZ=JMAK
      CALL SPLN1(1,T(1,1),T(1,5),T(1,9),NZ,T(1,2),T(1,K1),T(1,10),N)
      T(NQ,K2)=0.0
      T( 1,K2)=T(1,6)
      DO 175 K=1,JMAK
  175 T(K+1,K2)=T(K,K1)
      IF(K1.EQ.18) GO TO 180
      K1=18
      K2=16
      GO TO 165
  180 CONTINUE
      NZ=JMAK+1
      T( 1,2)=0.0
      T(NQ,2)=YMAX
      DO 185 K=1,JMAK
  185 T(K+1,2)=YY(K)
C     WING TIP CORRECTION
      K1=0
      XD=T(NQ,2)-T(NZ-1,2)
      PR=T(NZ-1,15)
      PI=T(NZ-1,16)
      IF(T(NZ,2)+.5*D.LT.T(NQ,2)) GO TO 195
  190 DUM=SQRT((T(NQ,2)-T(NZ,2))/XD)
      T(NZ,15)=DUM*PR
      T(NZ,16)=DUM*PI
      IF(K1.EQ.1) GO TO 196
  195 IF (YY(JMAK)+1.05*D.GE.YMAX) GO TO 196
      K1=1
      NZ=NZ+1
      NQ=NQ+1
      T(NQ,2)=T(NQ-1,2)
      T(NZ,2)=T(NZ-1,2)+D
      T(NQ,15)=T(NQ-1,15)
      T(NQ,16)=T(NQ-1,16)
      GO TO 190
  196 CONTINUE
      IF(ABS(T(NQ,2)-T(NZ,2)).GE.1.E-05) GO TO 197
      NQ=NQ-1
      NZ=NZ-1
      T(NQ, 2)=T(NQ+1, 2)
      T(NQ,15)=T(NQ+1,15)
      T(NQ,16)=T(NQ+1,16)
  197 CONTINUE
C
      CALL INTGL(T(1,2),T(1,15),T(1,1),T(2,1),T(1,19),2,NQ,NZ,NM)
      RETURN
      END
```

```
      SUBROUTINE MRED(DA,T,NM,NB,KSFM,SFMX,SFMY,SFMH,IW,IPRINT)
C     SPLINE-SURFACE FIT OF MACH NUMBER
      DIMENSION DA(1),T(NM,1),SFMX(1),SFMY(1),SFMH(1)
      CONST=0.28571429
      KSFM=DA(97)
      IF(KSFM) 80,50,10
C
C     FITTING OF GIVEN PRESSURE/MACH TO A SPLINE-SURFACE
C
   10 IF(NB-KSFM) 80,15,15
   15 CONTINUE
      KP=701
      DO 30 IP=1,KSFM
      SFMX(IP)=DA(KP  )/DA(24)
      SFMY(IP)=DA(KP+1)/DA(24)
      SFMH(IP)=DA(KP+2)
C     DA(96)=1, INPUT DATA ARE PRESSURE COEFFICIENT
C     DA(96)=2, INPUT DATA ARE LOCAL MACH NUMBER
      IF(DA(96)-1.0) 75,20,25
C     CONVERT PRESSURE COEFFICIENT INTO LOCAL MACH NUMBER
   20 SFMH(IP)=SQRT(5.*(1.2/((1.+0.7*SFMH(IP))**CONST)-1.))
   25 CONTINUE
   30 KP=KP+3
C     SPLINE-SURFACE FITTING OF DATA
   40 CONTINUE
      IF(IPRINT.NE.0) WRITE(IW,100)
      CALL SURF1(NM,KSFM,T,SFMX,SFMY,SFMH,IPRINT)
      RETURN
C
C     PRESENTLY INPUT OF PRESSURE COEFFICIENT  IN
C          A POLYNOMIAL FORM IS NOT ALLOWED
C     THE FOLLOWING IS FOR MACH INPUT AS A POLYNOMIAL  M=A0+A1*X
   50 CONTINUE
      KSFM=0
      SFMH(1)=DA(71)
      SFMH(2)=DA(72)*DA(24)
      SFMH(3)=0.0
      GO TO 40
C
   75 IPR=96
      GO TO 85
   80 IPR=97
   85 WRITE
     C  (IW,110)IPR
      STOP
  100 FORMAT(1H0,10X,73HCOMPUTED  MACH(X,Y) = A0+A1*X+A2*Y+ SUM OF H(I)*
     $(R(I)**2)*(ALOG(R(I)**2)))
  110 FORMAT(1H0,10X,14HMRED--BAD DATA,I5)
      END
```

```
      SUBROUTINE INTGL(X,Y,VR,VI,S,N,NQ,NZ,NM)
C     INTEGRATION BASED ON SPLINE FUNCTION
      DIMENSION X(1),Y(NM,2),S(NM,1)
C     DEFINE L(J)
      DO 20 I=2,NQ
   20 S(I,1)=X(I)-X(I-1)
      IF (NQ.EQ.2) GO TO 50
C     DEFINE TRI-DIAGONAL COEFFICIENT MATRIX
      DO 25 I=2,NZ
      S(I,2)= S(I,1)/6.0
      S(I,3)=(S(I,1)+S(I+1,1))/3.0
   25 S(I,4)= S(I+1,1)/6.0
      S( 2,2)=0.0
      S(NZ,4)=0.0
C     DEFINE RIGHT-HAND-SIDE COLUMN MATRIX
      K=1
   35 DO 40 I=2,NQ
   40 S(I,5)=(Y(I,K)-Y(I-1,K))/S(I,1)
      DO 45 I=2,NZ
   45 S(I,6)= S(I+1,5)-S(I,5)
C     SOLVE FOR COEFFICIENTS OF SPLINE FUNCTION M(J)
      CALL TRIDI(2,NZ,S(1,2),S(1,3),S(1,4),S(1,6),S(1,7),S(1,8),S(1,9))
   50 CONTINUE
      S( 1,7)=0.0
      S(NQ,7)=0.0
      VI=0.0
      DO 60 I=2,NQ
      VI=VI+0.5*S(I,1)
     $          *(Y(I,K)+Y(I-1,K)-S(I,1)*S(I,1)*(S(I,7)+S(I-1,7))/12.)
   60 CONTINUE
      IF(K.EQ.2) RETURN
      VR=VI
      IF(N.EQ.1) RETURN
      K=2
      GO TO 35
      END
```

```
      SUBROUTINE TRIDI(K1,K3,A,B,C,D,V,E,F)
      DIMENSION A(1),B(1),C(1),D(1),V(1),E(1),F(1)
      IF (K3.NE.K1) GO TO 5
      V(K3)=D(K3)/B(K3)
      RETURN
    5 CONTINUE
      E(K1)=B(K1)
      F(K1)=D(K1)/E(K1)
      K2=K1+1
      DO 10 I=K2,K3
      E(I)=B(I)-A(I)*C(I-1)/E(I-1)
   10 F(I)=(D(I)-A(I)*F(I-1))/E(I)
      V(K3)=F(K3)
      K2=K3-K1
      DO 20 J=1,K2
      I=K3-J
   20 V(I)=F(I)-C(I)*V(I+1)/E(I)
      RETURN
      END
```

```
      SUBROUTINE SPLN1(NI,X,Y,DY,N,XX,YY,DYY,NSMOS)
C     X,Y,DY=INTERPOLATION INDEPENDENT, DEPENDENT VARIABLES, AND DY/DX
C     NI     =NO. OF INTERPOLATION POINTS
C     XX,YY =INPUT INDEPENDENT AND DEPENDENT VARIABLES
C     N      =NO. OF INPUT POINTS
C     DYY    =D(YY)/D(XX) FOR INPUT DATA
C     NSMOS =CONTROLS OF SMOOTHING AND PRE-INTERPOLATION
      DIMENSION X(1),Y(1),DY(1),XX(1),YY(1),DYY(1)
      N2=N
      IPRE=NSMOS/10
      NSMO=NSMOS-IPRE*10
      IF(IPRE.EQ.0) GO TO 30
C     STORE INPUT DATA FOR PRE-INTERPOLATION
      DO 20 J=1,N
      Y(J)=XX(J)
   20 DY(J)=YY(J)
      CALL SPISET(N,XX,YY,DYY,0.0,0)
C     PRE-INTERPOLATION
      DO 25 I=2,N
      I1=I+I-2
      I2=I1+1
      XX(I2)=Y(I)
      YY(I2)=DY(I)
      XX(I1)=0.5*(Y(I-1)+Y(I))
   25 CALL SPLN2(XX(I1),1,N,Y,DY,DYY,YY(I1),DUM,1)
      N2=2*N-1
   30 CONTINUE
      IF(NSMO.EQ.0) GO TO 40
C     SMOOTH INPUT DATA XX, YY
      CALL SMOOTH(N2,XX,YY,Y,NSMO)
C     INTERPOLATE Y AT X FROM XX, YY, DYY AND CALCULATE DY=D(Y)/D(X)
   40 CALL SPISET(N2,XX,YY,DYY,0.0,0)
      CALL SPLN2(X,NI,N2,XX,YY,DYY,Y,DY,2)
      RETURN
      END


      SUBROUTINE SPLN2(XP,NP,N,X,Y,D,SPF,SPD,K)
      DIMENSION X(1),Y(1),D(1),XP(1),SPF(1),SPD(1)
C     EVALUATES A NATURAL CUBIC SPLINE AND ITS FIRST DERIVATIVE USING
C     SLOPE ARRAY D CALCULATED BY SPISET AND USING THE INPUT DATA
C     ARRAYS X AND Y
      DO 10 J=1,NP
      IF(XP(J).LT.X(1).OR.N.EQ.1) GO TO 6
      DO 2 I=2,N
      IF(XP(J).LT.X(I)) GO TO 4
    2 CONTINUE
      SPF(J)=Y(N)+D(N)*(XP(J)-X(N))
      IF(K.EQ.1) GO TO 10
      SPD(J)=D(N)
      GO TO 10
    4 C1=1./(X(I)-X(I-1))
      C2=X(I)-XP(J)
      C3=XP(J)-X(I-1)
      C4=C2*C1
      C5=C3*C1
      SPF(J)=C5*C5*((1.+2.*C4)*Y(I)-C2*D(I))
     $        +C4*C4*((1.+2.*C5)*Y(I-1)+C3*D(I-1))
      IF(K.EQ.1) GO TO 10
      C6=2.*C2-C3
      C7=2.*C3-C2
      SPD(J)=C1*C1*(C3*(2.*(1.+C1*C6)*Y(I)-C6*D(I))
     $            -C2*(2.*(1.+C1*C7)*Y(I-1)+C7*D(I-1)))
      GO TO 10
    6 SPF(J)=Y(1)-D(1)*(X(1)-XP(J))
      IF(K.EQ.1) GO TO 10
      SPD(J)=D(1)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SPISET(N,X,Y,D,RMS,IFRMS)                              SPISET
      DIMENSION X(1),Y(1),D(1)                                          SPISET
      DATA JMAX,RL,R/20,.16666667,1.07179677/                           SPISET
      D(1)=0.                                                           SPISET
      IF(N.EQ.1) RETURN                                                 SPISET
      D(N)=0.                                                           SPISET
      AN=N                                                              SPISET
      DO 1 II=2,N                                                       SPISET
      I=N+2-II                                                          SPISET
      X(I)=X(I)-X(I-1)                                                  SPISET
    1 Y(I)=(Y(I)-Y(I-1))/X(I)                                           SPISET
      IF(N.EQ.2) GO TO 5                                                SPISET
      DO 2 II=3,N                                                       SPISET
      I=N+3-II                                                          SPISET
      D(I-1)=2.*(Y(I)-Y(I-1))/(X(I)+X(I-1))                             SPISET
      Y(I)=1.5*D(I-1)                                                   SPISET
    2 X(I)=0.5*X(I-1)/(X(I)+X(I-1))                                     SPISET
      DO 3 J=1,JMAX                                                     SPISET
      DO 3 I=3,N                                                        SPISET
    3 D(I-1)=R*(Y(I)-X(I)*D(I-2)-(0.5-X(I))*D(I))-(R-1.0)*D(I-1)        SPISET
      DO 4 I=3,N                                                        SPISET
      X(I)=X(I-1)*(0.5/X(I)-1.)                                         SPISET
    4 Y(I)=0.33333333*Y(I)*(X(I)+X(I-1))+Y(I-1)                         SPISET
    5 SAVE=Y(2)-RL*X(2)*(2.*D(1)+D(2))                                  SPISET
      DO 6 I=2,N                                                        SPISET
      C=RL*X(I)                                                         SPISET
      S1=Y(I)-C*(2.*D(I-1)+D(I))                                        SPISET
      S2=Y(I)+C*(D(I-1)+2.*D(I))                                        SPISET
      D(I-1)=0.5*(SAVE+S1)                                              SPISET
    6 SAVE=S2                                                           SPISET
      D(N)=SAVE                                                         SPISET
      IF(IFRMS.NE.1) GO TO 8                                            SPISET
      RMS=0.                                                            SPISET
      IF(N.EQ.2) GO TO 8                                                SPISET
      DO 7 I=3,N                                                        SPISET
      C=2.*D(I-1)                                                       SPISET
    7 RMS=RMS+((C+D(I-2)-3.*Y(I-1))/X(I-1)+(D(I)+C-3.*Y(I))/X(I))**2    SPISET
      RMS=2.*SQRT(RMS/AN)                                               SPISET
    8 DO 9 I=2,N                                                        SPISET
      Y(I)=Y(I)*X(I)+Y(I-1)                                             SPISET
    9 X(I)=X(I)+X(I-1)                                                  SPISET
      RETURN                                                            SPISET
      END                                                               SPISET
```

```
      SUBROUTINE SMOOTH (N,X,Y,T,NSMOS)
C     THE Y ARRAY IS SMOOTHED BY A LOCAL FIVE POINT LEAST SQARES
C     CUBIC WEIGHTED BY W
      DIMENSION X(1),Y(1),T(1)
      IF(N.LT.5) RETURN
      DO 10 NS=1,NSMOS
      T(1)=NS
      AN=N
      S=(T(1)*(X(N)-X(1))/AN)**2
      DO 4 L=1,N
      K=MIN0(N-4,MAX0(1,L-2))
      K4=K+4
      DO 1 I=1,20
    1 T(I)=0.
      DO 3 M=K,K4
      W=1./(S+(X(L)-X(M))**2)
      R=1.0
      DO 3 I=1,4
      I4=I-4
      RR=1.0
      DO 2 J=1,4
      J4=4*J+I4
      T(J4)=T(J4)+R*RR*W
    2 RR=RR*X(M)
      T(I+16)=T(I+16)+R*Y(M)*W
    3 R=R*X(M)
      CALL CHLSKY(T,4,T(17),1,4)
      M=L-((L-1)/5)*5
      IF(L.GT.5) Y(L-5)=T(M+20)
      T(M+20)=0.
      R=1.0
      DO 4 J=1,4
      T(M+20)=T(M+20)+R*T(J+16)
    4 R=R*X(L)
      I4=N-5
      DO 5 L=1,5
      ML=M+L-((M+L-1)/5)*5
      J4=I4+L
    5 Y(J4)=T(ML+20)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CHLSKY(A,N,B,M,NX)                                     CHLSKY
      DIMENSION A(NX,1),B(NX,1)                                         CHLSKY
C     CHOLESKY DECOMPOSITION IS USED TO SOLVE THE MATRIX EQUATION  AX=B CHLSKY
C     WHERE THE COEFFICIENT MATRIX, A, IS SYMMETRIC. ON OUTPUT X IS     CHLSKY
C     STORED IN B                                                       CHLSKY
      IF(N.EQ.1) GO TO 6                                                CHLSKY
      DO 2 I=2,N                                                        CHLSKY
      I1=I-1                                                            CHLSKY
      DO 2 J=I,N                                                        CHLSKY
      DO 2 L=1,I1                                                       CHLSKY
    2 A(I,J)=A(I,J)-A(L,I)*A(L,J)/A(L,L)                                CHLSKY
      DO 5 K=1,M                                                        CHLSKY
      DO 3 I=2,N                                                        CHLSKY
      I1=I-1                                                            CHLSKY
      DO 3 L=1,I1                                                       CHLSKY
    3 B(I,K)=B(I,K)-A(L,I)*B(L,K)/A(L,L)                                CHLSKY
      DO 4 I=2,N                                                        CHLSKY
      I1=I-1                                                            CHLSKY
      DO 4 L=1,I1                                                       CHLSKY
      NI=N-I1                                                           CHLSKY
      NL=N+1-L                                                          CHLSKY
    4 B(NI,K)=B(NI,K)-A(NI,NL)*B(NL,K)/A(NL,NL)                         CHLSKY
      DO 5 I=1,N                                                        CHLSKY
    5 B(I,K)=B(I,K)/A(I,I)                                              CHLSKY
      RETURN                                                            CHLSKY
    6 A(1,1)=1./A(1,1)                                                  CHLSKY
      DO 7 L=1,M                                                        CHLSKY
    7 B(1,L)=A(1,1)*B(1,L)                                              CHLSKY
      RETURN                                                            CHLSKY
      END                                                               CHLSKY
```

```
      SUBROUTINE SURF1(NM,N,T,ABX,ABY,ABH,IRITE)                        SURF1
C     FIT DATA (N POINTS) BROUGHT THROUGH ABX,ABY,ABH                   SURF1
C     T -- TEMPORARY ARRAY FOR SPLINE-SURFACE FITTING                   SURF1
C     2 -- PROPERTY ARRAYS FOR INPUT POINTS TO BE FITTED                SURF1
C          ABX(I) - INDEPENDENT VARIABLE X                              SURF1
C          ABY(I) - INDEPENDENT VARIABLE Y                              SURF1
C          ABH(I) - COMES IN AS DEPENDENT VARIABLE OF X AND Y           SURF1
C                   GOES OUT AS COEFFICIENTS OF SPLINE-SURFACE          SURF1
      DIMENSION T(NM,1),ABX(1),ABY(1),ABH(1)                            SURF1
      IW=6                                                              SURF1
C                                                                       SURF1
      NP1=N   +1                                                        SURF1
      NP2=NP1+1                                                         SURF1
      NP3=NP2+1                                                         SURF1
      NP4=NP3+1                                                         SURF1
      IF(N.EQ.0) GO TO 13                                               SURF1
      DO 2 I=1,N                                                        SURF1
      T(I,NP2)=ABX(I)                                                   SURF1
      T(I,NP3)=ABY(I)                                                   SURF1
    2 T(I,NP4)=ABH(I)                                                   SURF1
      DO 4 I=1,N                                                        SURF1
      T(I,I)=0.                                                         SURF1
      T(I,NP1)=1.0                                                      SURF1
      T(NP1,I)=1.0                                                      SURF1
      T(NP2,I)=T(I,NP2)                                                 SURF1
    4 T(NP3,I)=T(I,NP3)                                                 SURF1
      NM1=N-1                                                           SURF1
      DO 6 I=1,NM1                                                      SURF1
      IP1=I+1                                                           SURF1
      DO 6 J=IP1,N                                                      SURF1
      XX=T(I,NP2)-T(J,NP2)                                              SURF1
      YY=T(I,NP3)-T(J,NP3)                                              SURF1
      H=XX*XX+YY*YY                                                     SURF1
      T(I,J)=H*ALOG(H)                                                  SURF1
    6 T(J,I)=T(I,J)                                                     SURF1
      DO 8 I=1,3                                                        SURF1
      IPN=I+N                                                           SURF1
      DO 8 J=1,4                                                        SURF1
      JPN=J+N                                                           SURF1
    8 T(IPN,JPN)=0.                                                     SURF1
      K=MSIMER(NM,NP3,1,T(1,1),T(1,NP4))                                SURF1
      IF(K.EQ.1) GO TO 9                                                SURF1
      WRITE(IW,220)                                                     SURF1
      STOP                                                              SURF1
    9 CONTINUE                                                          SURF1
C     STORE (N+3)-COEFFICIENT IN ARRAY  ABH                             SURF1
      DO 12 I=1,NP3                                                     SURF1
   12 ABH(I)=T(I,NP4)                                                   SURF1
   13 IF(IRITE) 18,18,14                                                SURF1
   14 WRITE(IW,200)                                                     SURF1
      WRITE(IW,120) (ABH(I),I=NP1,NP3)                                  SURF1
      IF(N.EQ.0) GO TO 18                                               SURF1
      DO 16 I=1,N                                                       SURF1
   16 WRITE(IW,110) I,ABH(I),ABX(I),ABY(I)                              SURF1
   18 CONTINUE                                                          SURF1
      RETURN                                                            SURF1
  110 FORMAT(10X,I5,1P3E14.7)                                           SURF1
  120 FORMAT(62X,1P3E14.7)                                              SURF1
  200 FORMAT(1H0,20X,37HWHERE R(I)**2=(X-X(I))**2+(Y-Y(I))**2/          SURF1
     11H0,10X,54H(IN DIMENSIONLESS COORDINATES - DISTANCE/CHORD LENGTH)/ SURF1
     21H0,13X,1HI,3X,4HH(I),10X,4HX(I),10X,4HY(I),                      SURF1
     315X,2HA0,12X,2HA1,12X,2HA2/)                                      SURF1
  220 FORMAT(5X,26HDID NOT CONVERGE IN  SURF1//)                        SURF1
      END                                                               SURF1
```

```
      SUBROUTINE SURF2(Z1,Z2,J1,J2,MXY,VALU,VLUX,VLUY,XI,YI,HI,N,K)
C     COMPUTE VALUE OF SPLINE-SURFACE FITTED DATA AT A POINT (X,Y)
C     Z1,Z2- COORDINATES OF THE POINT WHERE THE FITTED VALUE IS SOUGHT
C     VALU - FITTED VALUE SOUGHT
C     VLUX - GRADIENT OF FITTED VALUE IN X
C     VLUY - GRADIENT OF FITTED VALUE IN Y
C     XI,YI,HI - ARRAYS FOR KNOWN PROPERTIES IN SPLINE-SURFACE FORM
C     N -- NUMBER OF POINTS IN XI, YI ARRAYS
C     MXY=0  X=Z1(J), Y=Z2(J) WHERE J=J1,J2
C     MXY=1  X=Z1(1), Y=Z2(J) WHERE J=J1,J2
C     MXY=2  Y=Z1(1), X=Z2(J) WHERE J=J1,J2
      DIMENSION XI(1),YI(1),HI(1)
      DIMENSION Z1(1),Z2(1),VALU(1),VLUX(1),VLUY(1)
      NP1=N  +1
      NP2=NP1+1
      NP3=NP2+1
      IF(MXY.EQ.1) X=Z1(1)
      IF(MXY.EQ.2) Y=Z1(1)
      DO 40 J=J1,J2
      IF(MXY.EQ.1) Y=Z2(J)
      IF(MXY.EQ.2) X=Z2(J)
      IF(MXY.NE.0) GO TO 10
      X=Z1(J)
      Y=Z2(J)
   10 CONTINUE
      IF (K-2) 13,12,11
   11 VLUY(J)=HI(NP3)
   12 VLUX(J)=HI(NP2)
   13 VALU(J)=HI(NP1)+HI(NP2)*X+HI(NP3)*Y
      IF(N.EQ.0) GO TO 40
      DO 30 I=1,N
      TX=X-XI(I)
      TY=Y-YI(I)
      H = TX*TX+TY*TY
      HA=0.
      IF(H.GT.0.) HA=ALOG(H)
      HB=2.*(1.+HA)*HI(I)
      IF (K-2) 23,22,21
   21 VLUY(J)=VLUY(J)+HB*TY
   22 VLUX(J)=VLUX(J)+HB*TX
   23 VALU(J)=VALU(J)+HI(I)*H*HA
   30 CONTINUE
   40 CONTINUE
      RETURN
      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
C     FOR ABS(X) LESS THAN 1 A SERIES EXPANSION IS USED
C
    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
C     FOR ABS(X) GREATER THAN 1 APPROXIMATIONS OF HASTINGS ARE USED
C
    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
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      FUNCTION MSIMER(M,N,L,A,B)
      DIMENSION A(M,1),B(M,1)
      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.1.E-7) 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.1.E-7) 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(N,M,L,A,B)
      DIMENSION A(M,1),B(M,1)
      COMPLEX A,B,G
      DO 30 I = 1,N
      C = 0.0
      DO 10 J = 1,N
   10 C=AMAX1(C,ABS(REAL(A(I,[illegible](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.1.E-7) 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.1.E-7) 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
```

1. Report No. NASA CR-158907	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle SONIC-BOX METHOD EMPLOYING LOCAL MACH NUMBER FOR OSCILLATING WINGS WITH THICKNESS	5. Report Date September 1978 6. Performing Organization Code
7. Author(s) S. Y. Ruo	8. Performing Organization Report No. LG78ER0226
9. Performing Organization Name and Address Lockheed-Georgia Company 86 South Cobb Drive Marietta, Georgia 30063	10. Work Unit No. 11. Contract or Grant No. NAS1-13613
12 Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D. C. 20546	13. Type of Report and Period Covered Contractor Report 14. Sponsoring Agency Code

15 Supplementary Notes

Technical Monitor, Dr. E. Carson Yates, Jr., Aeroelasticity Branch, Structures and Dynamics Division, NASA-Langley Research Center, Hampton, Virginia 23365
Final Report

16. Abstract

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 for planar wing which has previously been extended to include the effects of the swept trailing edge and the thickness of the wing. Account for the non-uniform flow caused by finite thickness is made by application of the local linearization concept. The thickness effect, expressed in terms of the local Mach number, is included in the basic solution to replace the coordinate transformation method used in the earlier work. Calculations were made for a delta wing and a rectangular wing performing plunge and pitch oscillations, and the results were compared with those obtained from other methods. An input guide and a complete listing of the computer code are presented.

17. Key Words (Suggested by Author(s)) Unsteady Transonic Flow Lifting Surface Theory Transonic Potential Flow Aircraft Aerodynamics	18. Distribution Statement Unclassified - Unlimited STAR Category: 02

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 71	22. Price* $5.25

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