(NASA-CR-158906) IMPROVED SONIC-BOX COMPUTER PROGRAM FOR CALCULATING TRANSONIC AERODYNAMIC LOADS ON OSCILLATING WINGS WITH THICKNESS (Lockheed-Georgia Co., Marietta.) 128 p HC A07/MF A01 CSCL 01A G3/02

N79-10998

Unclas
36970

NASA CR-158906

IMPROVED SONIC-BOX COMPUTER PROGRAM FOR CALCULATING TRANSONIC AERODYNAMIC LOADS ON OSCILLATING WINGS WITH THICKNESS

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

Page

SUMMARY 1

INTRODUCTION 1

SYMBOLS 3

MAJOR VARIABLES IN COMPUTER PROGRAM 5

EQUATIONS USED IN THE COMPUTER PROGRAM 8

  Small Perturbation Potential Equation 8
  Pressure Coefficient 11
  Generalized Aerodynamic Force Coefficient 11

DATA FITTING METHODS 12

  Cubic Spline 12
    Method 12
    Use 13
  Spline-Surface 14
    Method 14
    Use 15

PROGRAM FLOW 16

FUNCTION OF SUBROUTINES 18

PROGRAM DIMENSIONS 24

RESULTS 24

  Aspect Ratio 1.5 Delta Wing 25
    Zero Thickness 25
    Nonzero Thickness 25
  Aspect Ratio 2.0 and 4.0 Delta Wings with Zero Thickness 26
  Cropped Delta Wing with Zero Thickness 28
  Arrow Wing with Zero Thickness 28
  Rectangular Wing 29
    Zero Thickness 29
    Nonzero Thickness 30

REMARKS AND RECOMMENDATIONS 30

Page

REFERENCES 33

APPENDIX: COMPUTER PROGRAM 74

Input Guide 74

Sample Case 77

Input 77
Output 80

Computer Program Listing 83

IMPROVED SONIC-BOX COMPUTER PROGRAM FOR CALCULATING TRANSONIC AERODYNAMIC LOADS ON 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 which has previously been extended by Ruo and Theisen to account for the effects of wing thickness. The latter work is further extended herein to improve computational efficiency and accuracy and to account for swept trailing edges. Account for the non-uniform flow caused by finite thickness is made by application of the local linearization concept with appropriate coordinate transformation. A brief description of each computer routine and the applications of cubic spline and the spline-surface data fitting techniques in the program are given, and the method of input is shown in detail. Sample calculations as well as a complete listing of the computer program 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. Sonic-box method uses a velocity-potential doublet as the basic solution to satisfy the linearized transonic flow, unsteady small-perturbation velocity-potential equation with the associated boundary conditions.

In reference 3, 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 the 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 the 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 predetermined polynomial surface for the subsequent calculation of the unsteady pressure and the generalized aerodynamic force coefficients. For wings with a rapidly changing Mach number distribution or wing motions in certain deflection mode shapes, the polynomial surface is not adequate because of its inaccuracy.

The computer program described in this report 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 present computer code allows 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. It also uses a different scheme from that of reference 3 for calculation of the generalized aerodynamic force coefficients for wings with finite thickness to improve the efficiency of the program and the accuracy of the numerical results. This is done by performing the force coefficient integration in the transformed space where the velocity potential is computed rather than map the velocity potential from the transformed space to the physical space before performing the integration as in reference 3. Additionally, the contribution from the partial boxes along the swept leading edge has been included to improve convergence and accuracy of the numerical results with a smaller total number of boxes in representing the wing planform.

The wing is assumed to have zero mean angle of attack, and the same mean Mach number distribution on upper and lower surfaces, and to perform motions symmetric with respect to the root chord. These, however, are not limited by the basic concept.

The regular output of the present computer code is the generalized aerodynamic force coefficients. In addition to these, the values of downwash, velocity potential, pressure coefficient and Mach number at each box center may be printed out. Also the portion of the dimensionless area representing the actual wing planform in each box and the information of the arrangement of the boxes used in the computation as well as the velocity influence coefficients may be printed out. Since no provision is made in the computer code to smooth any computed data, the pressure coefficient must be used with caution.

SYMBOLS

a_0, a_1, a_2	constants
B_{ij}	area of ij-th box on wing (i-th spanwise column and j-th chordwise row)
b	reference length (dimension = L)
C_p	pressure coefficient
d_j	x-direction distance between two adjacent points used in the spline-curve fit, equation (14)
e, exp	exponential function
f(x,y)	wing deflection mode shape
$G_1(x,y)$,$G_2(y)$	leading edge adjustment terms, see equation (10)

$H(x,y)$	dependent variable represented by a cubic spline or spline-surface, see equations (14) and (15)
h_j	constant
I	integrated value, see equation (13)
i	$\sqrt{-1}$
k	reduced frequency, $\omega b/U_\infty$
L	unit of length
L_{ij}	generalized aerodynamic force coefficient
$M(x,y)$	local Mach number
M_j	coefficients of cubic spline
N	number of points used in data fitting
r_i	dimensionless distance between two points, $\sqrt{(x-x_i)^2+(y-y_i)^2}$, (reference length = b)
S	dimensionless wing planform area of full wing (reference area = b^2)
t	dimensionless time (reference time = b/U_∞)
T	unit of time
U_∞	reference velocity (freestream), (dimension = L/T)
$w(x,y)$	dimensionless downwash (reference velocity = U_∞)
x,y,z	dimensionless Cartesian coordinates (reference length = b)
$x_{\ell e}(y)$	dimensionless x-coordinate of wing leading edge (reference length = b)
$x_{te}(y)$	dimensionless x-coordinate of wing trailing edge (reference length = b)
y_{max}	dimensionless maximum semi-span (reference length = b)
θ_{ij}	phase angle of L_{ij}
τ	maximum thickness to chord ratio
$\varphi_o(x,y)$	dimensionless velocity potential of doublet
$\psi(x,y)$	downwash at point (x,y) due to doublet of unit strength at the origin, see equation (6)
ω	angular velocity (dimension = radian/T)
$\overline{(\ )}$	superscript denotes the dimensional quantity of the corresponding dimensionless variable shown inside the parentheses
$\tilde{(\ )}$	superscript denotes the quantity in the transformed space of the corresponding variable shown inside the parentheses

MAJOR VARIABLES IN COMPUTER PROGRAM

Some variables used in the computer program are described briefly in this section. The subroutine name in the parentheses following the description of a variable indicates where its value is generated or defined. The number appearing in the parentheses indicates the value of the variable for the corresponding condition described. In the following, N=1 and N=2 indicate, respectively, the real and the imaginary parts of a variable; NEW=1 and NEW=2 indicate, respectively, the zero thickness and nonzero thickness cases; i ($i \leq MB$) is related to the chordwise coordinate while j ($j \leq MB$) is related to the spanwise coordinate.

A(N,j,i)	influence coefficient, the integral in equation (6); the upwash at the center of a box caused by a unit doublet distribution over another box separated from the former by (j-1) number of boxes in spanwise direction and (i-1) number of boxes in chordwise direction, (POT2)
AMA(j,i,NEW)	Mach number at center of ij-th box on physical wing (NEW=1), Mach number at a point on physical wing corresponding to the center of ij-th box on the transformed wing (NEW=2), (SHAPE)
AR(j,i,NEW)	area of wing in ij-th box, (PLNFM)
AREA	area of the physical, full wing planform, (PLNFM)
CK	reduced frequency, (MAIN)
CTE(j)	Mach number on the physical wing at a point corresponding to the trailing edge of the j-th chordwise row of the transformed wing, (SHAPE)
D	length of box side, (MAIN)
DA(k)	input data, (DATRD)
DH	one-half of the length of box side, (MAIN)
DI	maximum number (real) of boxes in streamwise direction (same as L integer), (MAIN)
EDG(NEW,j)	value of leading edge adjustment term G_2 in equation (10) at j-th chordwise row, (SHAPE)

IEDG	flag to identify whether the *most outboard section* of the leading edge is parallel (1) or is not parallel (0) to the freestream, (SHAPE)
IR	computer read-unit number, (MAIN)
IW	computer write-unit number, (MAIN)
JMAK(NEW)	number of boxes on the wing along spanwise direction, (SHAPE)
JMAX(NEW)	number of boxes on wing plus wake along spanwise direction, (SHAPE)
KSFD(M)	number of points used in spline-surface fit of wing deflections in M-th mode, (DRED)
KSFM	number of points used in the spline-surface fit of Mach number, (MRED)
L	maximum number (integer) of boxes in streamwise direction, (MAIN)
M	deflection mode number, $M \leq MD$, (MAIN)
MB	maximum number of boxes in streamwise direction corresponding to the dimension of those subscripted variables related to the box distribution, (MAIN)
MD	maximum number of deflection modes allowed, corresponding to the dimension of those subscripted variables related to deflection mode, (MAIN)
ML(NEW,i)	number of boxes, including partial boxes, in i-th spanwise column, (PLNFM)
MLC(NEW,k,i)	sequential number of first (k=1) and last (k=2) wing box in the i-th spanwise column, (PLNFM)
MLT(NEW,k,j)	sequential number of first (k=1) and last (k=2) wing box in the j-th chordwise row, (SHAPE)
MLW(NEW,i)	sequential number of first wake box next to the wing box in i-th spanwise column, (PLNFM)
NB	number of points to be used in the spline-surface fit, corresponding to the dimension of those subscripted variables related to the spline-surface, (MAIN)
NEDGI	maximum number of points allowed to describe the input leading edge of the wing, (MAIN)
NEW	index for physical (1) or transformed (2) wings, (MAIN)

NM	NB+3, (MAIN)
NS(NEW)	number of leading edge points on the wing semispan, including point at apex only if the wing is pointed,(SHAPE)
NSMAX	maximum number of leading-edge points (including apex of pointed wing) that can be used to approximate leading edge of transformed wing corresponding to the dimension of those subscripted variables related to the leading-edge points used in computation, (MAIN)
NST(NEW)	number of trailing-edge points including the point at wing root, $NST(1) \leq NTDGI-1$, (SHAPE)
NTDGI	maximum number of points allowed to describe the input trailing edge of the wing, (MAIN)
S(N,j,i)	downwash at the center of ij-th box, equation (8), (WVAL), and velocity potential at the center of ij-th box, (BOXPO)
SFDH(k,M,NEW)	known wing deflection at those points used in spline-surface fit of wing deflections in M-th mode; or, the coefficients in wing-deflection, spline-surface expression for M-th mode, $k \leq NM$, (DRED)
SFDX(k,M) SFDY(k,M,NEW)	x and y coordinates of points used in spline-surface fit of wing deflections in M-th mode, $k \leq NB$, (DRED)
SFMH(k)	known Mach number at those points used in spline-surface fit of Mach number; or, the coefficients in Mach number spline-surface expression, $k \leq NM$, (MRED)
SFMX(k) SFMY(k)	x and y coordinates of points used in spline-surface fit of Mach number, $k \leq NB$, (MRED)
T(k,ℓ)	dimensioned temporary storage, $k \leq NM$ and $\ell \leq (NM+1)$
XEDG(k) YEDG(k)	x and y coordinates of those points used to describe the wing leading edge in the computations, $k \leq NSMAX$, (SHAPE)
XEDGI(k) YEDGI(k)	x and y coordinates of those input points used to describe the wing leading edge in physical space, $k \leq NEDGI$, (SHAPE)
XLE(NEW,j) XTE(NEW,j)	x-coordinate of leading and trailing edge along j-th chordwise row, (PLNFM)
XTDG(k) YTDG(k)	x and y coordinates of those points used to describe the wing trailing edge in the computations, (SHAPE)

XTDGI(k) YTDGI(k)	x and y coordinates of those input points used to describe the trailing edge in physical space, k ≤ NTDGI, (SHAPE)
XX(i)	x-coordinate of i-th spanwise column, (SHAPE)
YMAX(NEW)	maximum y dimension of the wing, (SHAPE)
YY(j)	y-coordinate of j-th chordwise row, (SHAPE)

EQUATIONS USED IN THE COMPUTER PROGRAM

In this section, some equations and expressions used in the present version of sonic-box method computer program are listed. Additional information on the derivation of these equations can be found in the standard text books or numerous other publications (e.g., refs. 1, 4, and 6).

Small Perturbation Potential Equation

The governing equation of the unsteady part of the harmonically oscillating thin wing in a sonic flow field can be written as

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

and

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

When M = 1 everywhere, equation (1) reduces to the familiar linearized equation. However, if the variation of local Mach number is taken into account, local linearization may be applied to obtain the following equation valid in a small region

$$\tilde{\varphi}_{o_{\tilde{y}\tilde{y}}} + \tilde{\varphi}_{o_{\tilde{z}\tilde{z}}} - (2ik\tilde{\varphi}_{o_{\tilde{x}}} - k^2\tilde{\varphi}_o) = 0 \tag{2}$$

where

$$\begin{aligned} \tilde{x} &= x \\ \tilde{y} &= My \\ \tilde{z} &= Mz \\ \tilde{\varphi} &= M\varphi \end{aligned} \tag{3}$$

Equation (2) is a linear equation with constant coefficients and is the same as the familiar linearized equation with M=1 (eq.(1)). Therefore, the problems of wing with finite thickness can be solved in the transformed space with a method suitable for the problems of zero thickness wing in the physical space. The sonic-box method (ref. 1) has been chosen for the present implementation. A solution to the linearized form of equation (1), when M=1, can be written as (ref. 1)

$$\varphi_o(x,y,z) = \begin{cases} 0, & x \le 0 \\ \dfrac{ik}{2\pi}\dfrac{z}{x^2}\exp\left[-\dfrac{ik}{2}\left(x + \dfrac{y^2+z^2}{x}\right)\right], & x>0 \end{cases} \tag{4}$$

and in the wake

$$\varphi_o(x,y,0) = \varphi_{o_{te}} \exp[-ik(x-x_{te})],\ x>x_{te} \text{ and } z = 0, \tag{5}$$

which represents a doublet at the origin of coordinates pulsating with reduced frequency k. An expression for downwash at any point on a wing lying approximately in the xy-plane due to a unit doublet at the coordinate origin can be obtained by taking partial differentiation of equation (4) with respect to z and letting z go to zero. The approximate solution to equation (2) with a known downwash distribution on a lifting surface, which is approximated by a matrix of square boxes and the velocity potential is considered to be constant within each box and equal to that at the box center, can be expressed as:

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

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

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

$$= \frac{ik}{2}\frac{1}{(x_i-\xi)^2}\exp\left\{-\frac{1}{2}ik\left[(x_i-\xi) + \frac{(y_j-\eta)^2}{(x_i-\xi)}\right]\right\}; \tag{7}$$

$w(x_i,y_j) = \left(\frac{\partial\varphi_o}{\partial z}\right)_{\text{at }(x_i,y_j)}$ = downwash at the center of box B_{ij}.

Downwash is related to the deflection mode shape in physical space as

$$\frac{\partial \varphi_o}{\partial z} = \frac{\partial f}{\partial x} + ikf \tag{8a}$$

and in the transformed space as

$$\frac{\partial \tilde{\varphi}_o}{\partial \tilde{z}} = \frac{1}{M}\left(\frac{\partial \tilde{f}}{\partial \tilde{x}} + ik\tilde{f}\right). \tag{8b}$$

Since the left-hand sides of equations (8a) and (8b) are identical, the downwash at $(\tilde{x},\tilde{y})$ on the transformed wing is unchanged from that at the corresponding point, (x,y), on the physical wing.

It is not possible to arrange the center of the box to follow the wing leading edge with the sonic-box method. Before the pressure coefficient and the generalized aerodynamic force coefficients are calculated, it is recommended to adjust the computed velocity potential near the leading edge to conform to the theoretically known leading edge condition (e.g., $\varphi_{\ell e} = 0$ for a swept (subsonic) leading edge). This adjustment is expressed as

$$\varphi'_o = G_1 \cdot G_2 \cdot \varphi_o \ , \tag{9}$$

where G_1 and G_2, depending solely on the shape of the wing leading edge, are:

$$\begin{aligned}
&x - x_{\ell e} \geq \Delta x: \quad G_1 = 1.0 \\
&x - x_{\ell e} < \Delta x: \quad G_1 = \sqrt{\frac{x - x_{\ell e}}{\Delta x}} \left\{ \sqrt{\frac{x + x_{\ell e}}{2x_{\ell e} + \Delta x}} \right\}^m , \quad \begin{cases} m=0 \text{ for } y \neq 0 \text{ at } x=0 \\ m=1 \text{ for } y=0 \text{ at } x=0 \end{cases} \\
&y_{max} - y \geq \Delta y: \quad G_2 = 1.0 \\
&y_{max} - y < \Delta y: \quad G_2 = \left\{ \sqrt{\frac{y_{max}^2 - y^2}{\Delta y (2y_{max} - \Delta y)}} \right\}^n , \quad \begin{cases} n=0 \text{ for } \frac{dy}{dx} \neq 0 \text{ at } x=1 \\ n=1 \text{ for } \frac{dy}{dx} = 0 \text{ at } x=1 \end{cases}
\end{aligned} \tag{10}$$

where

$x_{\ell e}(y)$ = x coordinate of wing leading edge at y = constant

y_{max} = y coordinate of the wing tip

Δx = range of x for which leading-edge adjustment is applied

Δy = range of y for which leading-edge adjustment is applied.

The selection of the range for leading-edge adjustment is arbitrary depending on how extensively one wants to impose the leading-edge property on the calculated velocity potential. For example, if one choses $\Delta x = x_{te} - x_{\ell e}$ and $\Delta y = y_{max}$, the resulting adjustment becomes identical to that used in earlier versions of sonic-box method computer program. Presently, both Δx and Δy are set to the length of the box side used in computation. These relationships are depicted in figure 2.

Pressure Coefficient

The unsteady pressure coefficient for the harmonically oscillating thin wing is obtained from Bernoulli's equation as

$$\begin{aligned} C_{P_o} &= -2(\varphi_{o_x} + ik\varphi_o), && \text{on the wing} \\ C_{P_o} &= 0, && \text{in the wake} \end{aligned} \tag{11}$$

and

$$C_P(x,y,z,t) = C_{P_o}(x,y,z)e^{ikt}.$$

Generalized Aerodynamic Force Coefficient

The generalized aerodynamic force coefficient as defined in reference 6 is written as

$$L_{ij} = \frac{8}{S}\iint_S (\varphi_{o_x} + ik\varphi_o)_i \, f_j \, dxdy \tag{12a}$$

$$= \frac{8}{S}\iint_{\tilde{S}} \frac{1}{M^2} (\tilde{\varphi}_{o_{\tilde{x}}} + ik\tilde{\varphi}_o)_i \, \tilde{f}_j \, d\tilde{x}d\tilde{y}. \tag{12b}$$

An integration by parts is performed to cast the integral in equation (12) into the following form:

$$I = \int_{y} [(\varphi_o f)_{x=x_{te}} - \int_{x} (f_x + ikf)\varphi_o dx]dy, \quad x,y \text{ on } S, \tag{13a}$$

$$\tilde{I} = \int_{\tilde{y}} [(\frac{1}{M2}\tilde{\varphi}_o\tilde{f})_{\tilde{x}=\tilde{x}_{te}} - \int_{\tilde{x}} \frac{1}{M2}(\tilde{f}_{\tilde{x}} + ik\tilde{f})\tilde{\varphi}_o d\tilde{x}]d\tilde{y}, \quad \tilde{x},\tilde{y}, \text{ on } \tilde{S}. \tag{13b}$$

Equation (12b) is integrated in the transformed space using equation (13b).

DATA FITTING METHODS

The polynomial surface method of data fitting was used in the earlier versions of sonic-box computer program (refs. 1, 2, and 3) to fit the wing deflection mode, Mach number, velocity potential and pressure coefficient.

In the present program, two types of data fitting are used. One is a two-dimensional cubic spline fitting and the other is a three-dimensional spline-surface fitting. The former is based on the concept of thin elastic beam and the latter on the thin elastic plate. Since they use different forms of expression, the two-dimensional fitting can not be obtained directly from the three-dimensional fitting by setting the coefficient of the terms involving the extra independent variable to zero as it is usually done when the polynomial fitting is used.

Cubic Spline

Method. - It has been demonstrated over the years that the draftsman's spline has distinct advantage over other methods of fairing curve through a set of predetermined points (e.g., reference 7). The basic equation can be written as

$$\begin{aligned} H = {} & \frac{M_{j-1}}{6}[(x_j - x^3) + d_j^2\,(x - x_{j-1}) - d_j^3] + \frac{M_j}{6d_j}[(x - x_{j-1}^3) - d_j^2(x - x_{j-1})] \\ & + H_{j-1} + T_j\,(x - x_{j-1}), \qquad \text{for } x_{j-1} \le x \le x_j \text{ and } j = 2,N \end{aligned} \tag{14}$$

where $d_j = x_j - x_{j-1}$

H = value of dependent variable at point x

H_j = value of dependent variable at point x_j

M_j = coefficients of cubic spline

N = total number of predetermined points

$T_j = (H_j - H_{j-1})/d_j$

x_j = coordinate of the predetermined points where H is known

The values of M_j's are computed with the N known values of H at x_j by requiring the slope at each of these points to have a value computed from both sides of the point. This provides only (N-2) equations for N unknowns, thus the following two additional equations are assumed:

$$M_1 = M_N = 0.$$

Use. - Cubic spline fitting is used in the computation of pressure coefficient and generalized aerodynamic force coefficients according to equations (11) and (13), respectively.

In the computation of pressure coefficient distribution on both physical and transformed wings, the known values of the velocity potential at the center of wing boxes, in the respective spaces, along each chordwise row are fitted with a cubic spline to calculate the chordwise derivative of the velocity potential at the box centers. The velocity potential distribution on the transformed wing is also fitted along each spanwise column to facilitate the interpolation of the velocity potential at the center of wing boxes along the same column on the physical wing.

In the computation of aerodynamic force coefficients, the integration is performed in the space where the velocity potential distribution is calculated and in the sequence of chordwise and then spanwise directions. The cubic spline is used to linearly extrapolate both the velocity potential at the trailing edge of each chordwise row for the chordwise integration and the integrated value of the chordwise integration at the root chord for the spanwise integration. Both chordwise and spanwise integrations are performed with the implied cubic-spline fitting as shown in equation (16).

Spline-Surface

Method. - In reference 8, it is shown that a spline-surface can fit a three-dimensional distribution of data better than the polynomial surface. The expression of the spline-surface can be written as:

$$H = a_0 + a_1x + a_2y + \sum_{j=1}^{N} h_j r_j^2 \ \ell n r_j^2, \tag{15}$$

where

$r_j^2 = (x-x_j)^2 + (y-y_j)^2$

a_0, a_1, a_2, h_j = coefficients of spline-surface

H = value of dependent variable at (x,y)

N = total number of predetermined points

(x_j, y_j) = coordinates of the predetermined points where H is known.

The valus of a_0, a_1, a_2, and h_j's are computed with the N known values of H at (x_j, y_j) plus the following three additional relations:

$$\sum_{j=1}^{N} h_j = 0$$

$$\sum_{j=1}^{N} h_j x_j = 0$$

$$\sum_{j=1}^{N} h_j y_j = 0.$$

This yields (N+3) equations with (N+3) unknowns.

The partial derivatives of equation (15) with respect to x, y can be easily obtained as follows:

$$\frac{\partial}{\partial x} H(x,y) = a_1 + 2 \sum_{j=1}^{N} h_j(x-x_j) \cdot \ell n r_j^2$$

$$\frac{\partial}{\partial y} H(x,y) = a_2 + 2 \sum_{j=1}^{N} h_j(y-y_j) \cdot \ell n r_j^2$$

Use. - There are five sets of discrete points on the lifting surface which in general do not coincide:

1. (x,y) on physical wing where Mach number is known
2. (x,y) on physical wing where modal deflection is known
3. (x,y) of box centers on physical wing, or
   $(\tilde{x},\tilde{y})$ of box centers on transformed wing
4. $(\tilde{x},\tilde{y})$ on transformed wing corresponding to (x,y) on physical wing where modal deflection is known
5. (x,y) on physical wing corresponding to box centers of transformed wing.

In the process of solving equation (6) for the velocity potential, spline-surface fitting is used to facilitate the interpolation of quantities and calculation of their gradients among these sets of points. The known local Mach number distribution on physical wing is surface-fitted for the interpolation of local Mach number on the physical wing at the box centers (for print-out purpose only), at those points where the modal deflection is specified, and at those points corresponding to the box centers and the trailing-edge points of transformed wing (for use in the evaluation of the integrals in eq. (13b)). The modal deflection points in the physical space are then transformed into transformed space according to equation (3) for nonzero-thickness wing. Note that the modal deflection $f(x,y)$ is unchanged in the transformed space, i.e.. $\tilde{f}(\tilde{x},\tilde{y}) = f(x,y)$. The modal deflection, in the physical space for zero-thickness wing and in the transformed space for nonzero-thickness wing, is surface-fitted to facilitate the evaluation of the modal deflection and its chordwise derivative at box centers and the modal deflection at trailing edge as required in equation (8) for downwash and in equation (13) for force coefficient in the respective spaces.

When the modal deflection of the physical wing is expressed in a functional form, a number of points (i.e., box centers) on the transformed wing are selected for specifying the modal deflection and the corresponding points on the physical wing are then located with the aid of the surface-fitted Mach number. The modal deflection at each of these points on the physical wing is then calculated with the known expression and this, in turn, is the value at each of the corresponding points on the transformed wing.

PROGRAM FLOW

The flow chart of the program SBOXR is presented in figure 1. The function of those controlling variables appeared in the flow chart is as follows:

Variable	Value	Function
NEW	1	Indicates case without thickness effect
	2	Indicates case with thickness effect
DA(26)	0	Indicates first frequency for a wing
	1	Indicates additional frequency for the same wing
DA(50)	0	Calculates cases with and without thickness effects
	1	Calculates case without thickness effects only
	2	Calculates case with thickness effects only
K M N	≤DA(28)	Counter of the modes of deflection
DA(28)	≤ MD	Total number of modes to be considered

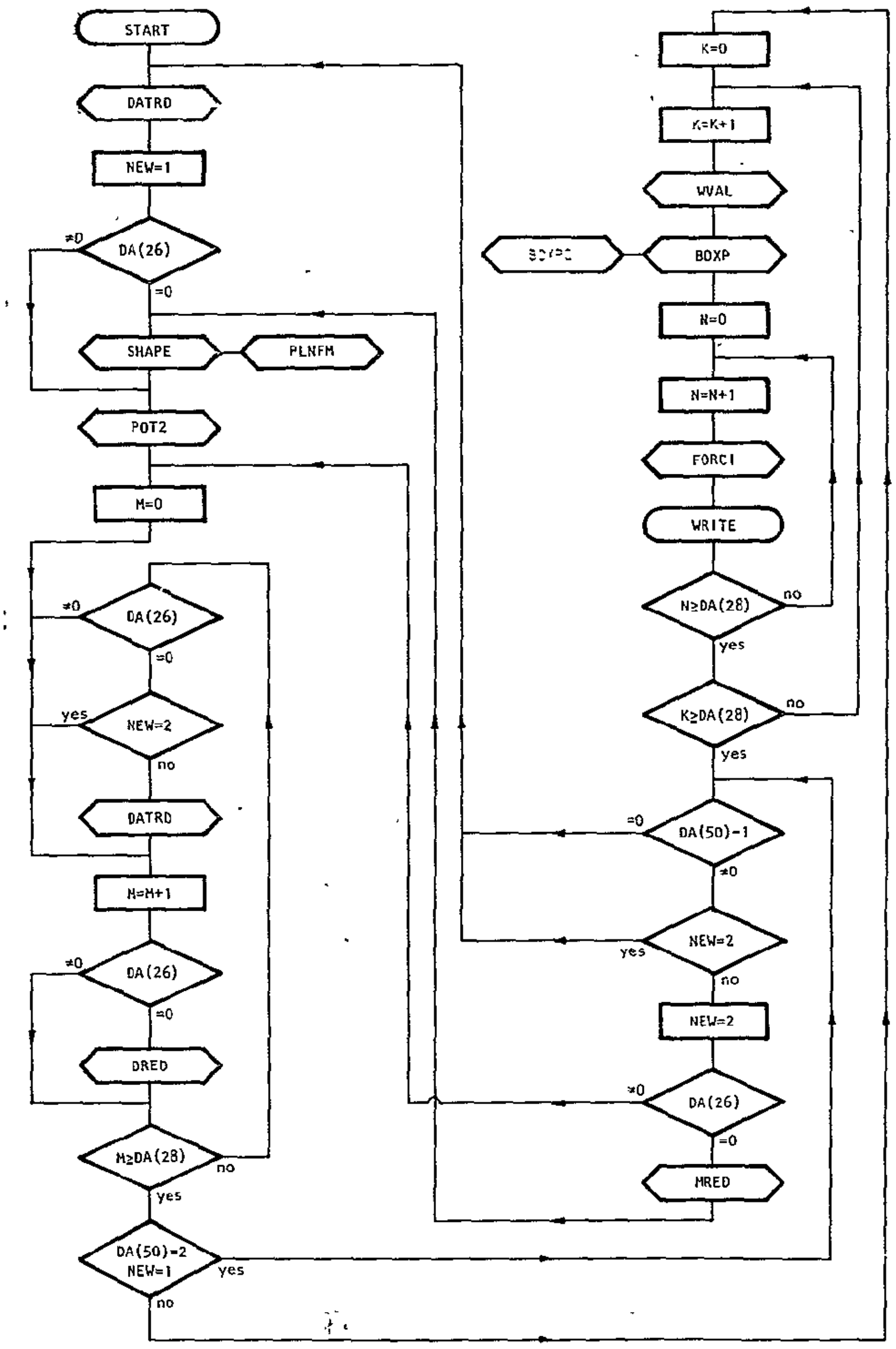


Figure 1. - Flow chart of program SBOXR.

FUNCTION OF SUBROUTINES

A brief description of the function of each subroutine is given in this section. Some equations are noted here for the purpose of identifying the steps in the sequence of obtaining the final results, i.e., the generalized aerodynamic force coefficients. It is suggested to refer to reference 1 for the derivations of these equations. The freestream flow is in the positive x direction and the apex of wing is at the origin. The spanwise direction is designated by y. Since the plane passing through the centerline chord and perpendicular to the wing planform is the plane of symmetry of wing geometry and motion, the input of wing geometry, deflection, and steady-flow Mach number distribution needs to be made only for the portion of the wing where $x \geq 0$ and $y \geq 0$ (see figure 3). It is assumed that the camber, twist, and mean angle of attack of the wing are all small; and the difference in local Mach number at the corresponding points of the upper and lower wing surfaces can be neglected. As the computer program is presently formulated, the motion is limited to symmetric modes. This limitation is imposed by the expression used in the present program to evaluate the influence coefficient of upwash due to a unit doublet distribution.

In the following, the variable name used in the computer program to represent a certain quantity mentioned in the description of a subroutine is indicated in the parentheses following such quantity, also, whenever a variable NEW appears as a subscript to another variable it is implied that the operation is applicable to either physical (NEW=1, without thickness effect) or transformed (NEW=2, with thickness effect) wings.

SBOXR - a controlling routine; also performs the final calculation of the generalized aerodynamic force coefficient for the full wing as given in equation (12), and prints out the results.

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

SHAPE - wing geometry routine; the main objectives of this routine are as follows:

(1) approximates the wing planform of the physical or tranformed wing with a grid of square boxes and calculates the number of boxes (ML(NEW,i)) along each spanwise column (XX(i))

(2) finds the number sequence of the leading (MLT(NEW,1,j)) and trailing edge (MLT(NEW,2,j)) wing boxes along each chordwise row (YY(j))

(3) finds the number sequence of the first (MLC(NEW,1,i)) and the last (MLC(NEW,2,i)) wing boxes along each spanwise column (XX(i))

(4) computes the wing tip adjustment term (EDG(NEW,i)) along each spanwise column (XX(i))

(5) finds the Mach number at each wing box center on the physical wing (AMA(j,i,1)) and at those points on the physical wing corresponding to the box centers (AMA(j,i,2)) and the trailing edge (XTE(NEW,j), CTE(j)) along the chordwise row (YY(j)) of the transformed wing

(6) redistributes the number of leading edge (XEDG(k), YEDG(k)) and trailing edge (XTDG(k), YTDG(k)) segments by subdividing the corresponding input data (XEDGI(k), YEDGI(k) and XTDGI(k), YTDGI(k)) for the physical wing so that the transformed wing can be better approximated

(7) performs the coordinate transformation of the wing with thickness effect with the aid of the known Mach number distribution (SFMX(k), SFMY(k), SFMH(k), KSFM) on the physical wing, using the relations shown in equation (3); and also checks the possible creation of an artificial wake in the transformed wing resulting from an uneven distortion of the wing tip

PLNFM - wing geometry routine called from subroutine SHAPE; the main functions of this routine are as follows:

(1) computesactual wing area in each box (AR(j,i,NEW))

(2) computes the x-coordinates of the wing leading edge (XLE(NEW,j)) and trailing edge (XTE(NEW,j)) along each chordwise row (YY(j))

(3) calculates the full wing planform area (AREA) of the physical wing

(4) finds the total number of boxes (ML(NEW,i)) including the wing as well as the wake along each spanwise column (XX(i))

(5) finds the number sequence of the first (MLC(NEW,1,i)) and the last (MLC(NEW,2,i)) wing boxes along each spanwise column (XX(i)) in which the actual wing planform occupies at least 50% of the box area.

(6) identifies, along each spanwise column (XX(i)), the number sequence of the first wake box (MLW(NEW,i)) of the wing-tip wake (MLW(NEW,i)<0) and the last wake box of the wing-root wake (MLW(NEW,i)>0)

POT2 - evaluates the integral in the expression relating the downwash and the velocity potential (equation (6)). The integral in equation (6) is integrated over the surface of each box $B_{i'j'}$ centered at $(x_{i'},y_{j'})$ once for every frequency, and the result (A(N,j,i)) is used in the calculation of the velocity potential (S(N,j,i)) in both physical and transformed spaces for all modes under consideration at the same frequency.

DRED - fits the wing deflection for each mode (M) of motion in both physical and transformed space (SFDX(ℓ,M), SFDY(ℓ,M,NEW), SFDH(ℓ,M,NEW), KSFD(M)) with a spline-surface of the form shown in equation (15) expressed in terms of the dimensionless coordinates. For the physical wing, the wing deflection in each mode can be given as input either in the form of the coefficients (SFDH(ℓ,M,1)) of an expression $H = a_0+a_1\bar{x}$ (KSFD(M)=0) or deflection (SFDH(ℓ,M,1)) at a number of predetermined points (SFDX(ℓ,M), SFDY(ℓ,M,1), KSFD(M)>0). For the transformed wing, the wing deflection in a particular mode is obtained either directly from the input data points for the physical wing (KSFD(M)>0) or indirectly from a number of selected points (KSFD(M)>0) on the physical wing of known deflection of form $h = a_0+a_1x$. The deflection (SFDH(ℓ,M,NEW)) and the chordwise coordinate (SFDX(ℓ,M)) of a point on the transformed wing are unaltered from those of the corresponding point on the physical wing, but its spanwise coordinate (SFDY(ℓ,M,NEW)) in the transformed space is modified by a factor of the Mach number (SFMX(k), SFMY(k), SFMH(k), KSFM) from that of the physical wing (see equations (3) and (8)).

SLCT - called from subroutine DRED; selects a maximum of NB number of points from the input data on the physical (NEW=1) or transformed (NEW=2) wing for the spline-surface fitting of the wing deflection.

WVAL - calculates the complex (real N=1, imaginary N=2) downwash (S(N,j,i,)) at the center of each box (ij-th) for a particular mode (M) and reduced frequency (CK) using the known wing deflection spline-surface (SFDX(ℓ,M), SFDY(ℓ,M,NEW), SFDH(ℓ,M,NEW), KSFD(M)) for the planar wing in physical (NEW=1) or transformed (NEW=2) space.

BOXP - incorporates the leading edge adjustment, equation (9), to the computed velocity potential (S(N,j,i)); computes pressure coefficient, equation (11), when a print out of its value is desired. To compute the pressure coefficient for the wing with thickness, the velocity potential on the physical wing is found by transforming the known velocity potential on the transformed wing with the known coordinates and Mach number on the physical wing.

BOXPO - called from subroutine BOXP, solves a set of simultaneous equations, equation (6), relating the complex velocity potential (S(N,j,i)) and the known complex downwash of a particular mode (M) at the center of each box (ij-th) to obtain the velocity potential of the zero thickness wing in the physical and the transformed spaces. Since the disturbance is considered not to travel upstream, the calculation is carried out for one spanwise column at a time starting from the most forward column. The number of upstream columns influencing a particular column is controlled through the input (DA(49)). If a wake is encountered, the velocity potential in the wake along a chordwise row (YY(j)) is computed using equation (5) for various chordwise stations; the velocity potential at the trailing edge (XTE(NEW,j)) is interpolated from the velocity potential computed for the last wing box and the first wake box along the same chordwise row by considering the first wake box to be a wing box.

FORCI - performs the integration of the generalized aerodynamic force coefficient, equation (12a) or (12b), according to whether the wing is in the physical or transformed space. At each box center on the wing, the values of f and f_x as required in equation (13) are obtained from the fitted spline-surface, equation (15) and φ_o is known. For wing with thickness, the Mach number at each box center on the transformed wing (eq. (13b)) is interpolated from the spline-surface representing Mach number distribution on the physical wing. The integration in x-direction along each chordwise row (YY(j)) is carried out by computing the value of the integrand at a number of points along a particular row and then completing the integration with the quadrature expression shown in equation (16). The integration in y-direction is accomplished in the same manner. For the wing with thickness effect, the procedure of evaluating the integral is the same except the involvement of the Mach number at those points on the physical wing corresponding to the box centers (AMA(j,i,2)) and trailing edge (CTE(j)) on the transformed wing.

MRED - fits the mean steady-state Mach number distribution (SFMX(k), SFMY(k), SFMH(k), KSFM) over the physical wing with a spline-surface, equation (15), expressed in terms of the dimensionless coordinates. The input for the Mach number can be either in the form of the coefficients of an expression Mach number = $a_0+a_1\bar{x}$ (KSFM=0), or as tabulation (SFMH(k)) of the Mach number, or the pressure coefficient at a number of predetermined points (SFMX(k), SFMY(k), KSFM>0) on the physical wing.

INTGL - called from subroutine FORCI; carries out the evaluation of the integral in equation (13) by, first, fitting the values of the integrand at a set of predetermined points with a cubic spline, equation (14), and then by using the following quadrature expression:

$$I = \frac{1}{2}\sum_{j=2}^{N} d_j \left[- \frac{1}{12} d_j^2 \left(M_{j-1}+M_j\right) + \left(H_{j-1}-H_j\right)\right]. \tag{16}$$

Equation (16) sums the sectionally integrated area under a cubic spline (equation (14)); which fits N values of the integrand, between two adjacent points, j-1 and j.

TRIDI - called from subroutine INTGL; solves for the roots of a set of equations whose coefficient matrix is a tri-diagonal matrix.

SPLN1 - provides a cubic-spline fit for a variable at a given set of points, from which the interpolated values of the dependent variable and its derivative may be obtained using subroutine SPLN2; also inserts an additional point between every successive pair of input points in the original set to increase the number of points for cubic-spline fitting and performs the specified number of smoothing iterations using subroutine SMOOTH.

SPLN2 - called from subroutine SPLN1; interpolates the value of the dependent variable and calculates its derivative at the set of desired locations.

SPISET - called from subroutine SPLN1; computes the slope of the dependent variable using the cubic spline method at those points where the values of the dependent variable are known.

SMOOTH - called from subroutine SPLN1; replaces the value of the dependent variable at each of the predetermined points with the new value computed by passing through a five-point least-squares cubic before the slope at these points is computed.

CHLSKY - called from subroutine SMOOTH; solves for the roots of a set of simultaneous equations whose coefficient matrix is a symmetric matrix.

SURF1 - spline-surface fitting routine; with the known values of the dependent variable at a number of points of known coordinates, this routine finds the coefficients of the spline-surface associated with those known points of the form shown in equation (15).

SURF2 - interpolating routine; finds the value of the dependent variable and its partial derivatives at a specified point based on the fitted spline-surface.

CIN - calculates sine and cosine integrals.

MSIMER - finds solutions of the simultaneous real number equations.

MSIMEC - finds solutions of the simultaneous complex number equations.

PROGRAM DIMENSIONS

The present computer program is dimensioned to handle a maximum of 30 (MB) boxes either in chordwise or spanwise direction in approximating one-half of the wing planform. The maximum numbers of leading and trailing edge segments are, respectively 7 (NEDGI-1) and 2 (NTDGI-1). It can handle up to 3 (MD) wing deflection mode shapes. The maximum number of points used in the spline-surface fitting is 100 (NB). These limitations can easily be increased by changing the dimensions of the corresponding variables in the computer program.

The maximum number of input points to describe the wing deflection and the thickness effect expressed in terms of the steady-state, either the Mach number distribution or pressure coefficient distribution is presently dimensioned 100 (NB). This number can be increased to 128 by activating the unused portion of the input data array, DA, from locations 501 to 700 and by changing the value of KP, the starting location for the thickness effect input points, from 701 to 613 in subroutine MRED and the value of NB from 100 to 128 in program SBOXR. The dimensions of the variables used in the spline-surface fitting need to be changed according to the new value of NB.

RESULTS

Sample calculations were made using the present program as well as other versions of sonic-box computer program. It was found necessary to use a unit-box increment instead of a multi-box increment in the convergence test due to the inherent property of the box method. Since no smoothing is applied to any computed values in the present program, the pressure coefficient calculated should be used with caution. In order to use it, one needs to put the pressure coefficient distribution through a smoothing process such as the built-in smoothing routine which was not utilized, however, for the results presented in this section. 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.

Wings considered in the sample calculations are delta, cropped delta, arrow and rectangular wings and are given separately below.

Aspect Ratio 1.5 Delta Wing

Zero Thickness. - Shown in figure 4 is the convergence with respect to the number of boxes along the root chord of the generalized force coefficients due to plunge (mode 1) and pitch about its apex (mode 2) at a reduced frequency 0.2 for an aspect ratio 1.5 delta wing. The results obtained from the present and the other sonic-box computer programs given in references 1 and 2 are plotted in the same figure for comparison. The program of reference 3 generated the same results as that of reference 1. The results of the present program appear to show more clearly the converging trend at a fewer number of boxes used in the computation than the other sonic-box programs.

The computed and the smoothed velocity potentials along a chordwise row, y = 0.0166667 from root chord, of the aspect ratio 1.5 delta wing due to plunge at a reduced frequency 0.2 from reference 3 using 30 boxes along the root chord are plotted in figure 5. The results from the programs of references 1 and 2 show the similar pattern of velocity potential distribution. In these programs, a smoothing of the computed velocity potential distribution is applied to obtain the distribution for the subsequent computation of the pressure and force coefficients. The velocity potential distribution over the same wing at the same conditions using the present program is shown in figure 6. In the present illustration, no smoothing is used; thus, the velocity potentials used in the computation differ from the computed values only near the leading edge where the leading edge adjustment is applied.

Nonzero Thickness. - Figure 7 shows the variation of the force coefficients due to plunge (mode 1) and pitch about its apex (mode 2) as function of the reduced frequency for the aspect ratio 1.5 delta wing computed with the present program using 30 boxes along the root chord. The results with thickness were calculated with the assumption that the wing has an elliptic cross-section in a plane perpendicular to the chordwise axis and the steady-state pressure coefficient distribution used as input was obtained from the method given in reference 9. The distribution of Mach number at the box centers interpolated from the fitted spline-surface is plotted in figure 8. The Mach number distribution in spanwise direction for this wing is supposed to be constant but the interpolated value has deteriorated near the trailing edge. Since the error in magnitude is not large, the effect on the integrated results shown in figure 7 may not be very serious. As it was observed in reference 4, the present results also show that the thickness effect on the force coefficients of a delta wing is not very large. However, the effect on flutter speed can be significant (ref. 10).

Aspect Ratio 2.0 and 4.0 Delta Wings with Zero Thickness

Convergence of the force coefficient due to pitch, $f = -0.00617 + 0.017446\,\bar{x}$ (b = 0.5 ft.), of aspect ratio 2.0 delta wing at a reduced frequency 0.106 is shown in figure 9 and that due to plunge (mode 1) and pitch about its apex (mode 2) of aspect ratio 4.0 delta wing at a reduced frequency 0.1 is shown in figure 10. In figure 9, the present results are compared with that given in reference 5. The results of references 1 and 3 are included in figure 10 for comparison. According to the calculations performed for the delta wings, the results of the present program show a better and faster converging trend than that of the other versions of the sonic-box method computer program.

The computed velocity potential distribution without smoothing, due to plunge ($f = 1.0$) at a reduced frequency 0.1 of aspect ratio 4.0 delta wing from the program of reference 1 is plotted in figure 11, and that from the present program is plotted in figure 12. The numerical values of these two sets of results are not very different. The fluctuation of the values of velocity potential along the chordwise row as observed in figure 5 is not observed in figure 11. This is due to the difference in the box arrangement along the swept leading edge of the delta wings.

The convergence plots of the force coefficients of a delta wing as function of the number of boxes along the root chord always show fluctuations; it does not increase or decrease monotonically toward the converged values. This is an inherent property of the box-method.

As shown in figure 13, the box arrangement along the swept leading edges can only approximate that of the actual wing. The number of boxes in a spanwise column does not increase uniformly from the nose to the tail of the delta wing. For example, the aspect ratio 1.5 delta wing has 3 spanwise columns with the same number of boxes, followed by another 3 columns with the same number of boxes and followed by 2 columns with the same number of boxes and this sequence, 3-3-2, is repeated toward the trailing edge of the wing. For the aspect ratio 2.0 and 4.0 delta wings, the sequences are, respectively, 2-2 and 1-1.

From the plots shown in figure 5, one notices that the fluctuations of the computed velocity potential along a chordwise row is caused by the sequence of increase of the number of boxes in the spanwise columns from the nose toward the tail of the wing. The plots in figure 5 clearly show that the values of the computed velocity potential along a chordwise row increase whenever the number of boxes in a spanwise column increases and remains the same or decreases whenever the number of boxes remains unchanged. This is the way the fluctuation is formed.

The fluctuation of the computed velocity potential along the chordwise row of the aspect ratio 4.0 delta wing is not observed. This is due to the fact that the number of boxes in the spanwise column increases by one uniformly from the leading edge to the trailing edge, and the magnitude of the computed velocity potential only increases in value.

The fluctuation of the force coefficient as function of the number of boxes along the root chord shown in figure 9 is probably caused by the number of boxes in the spanwise column at the trailing edge of the aspect ratio 2.0 delta wing. The box number sequence for this wing is 2-2 and the last spanwise column may end at the middle or end of the sequence depending on whether an even or an odd number of boxes along the root chord is chosen. The example shown in figure 5 ends at the middle of the sequence.

The similar data from the present program are shown in figures 6 and 12. Except for the leading edge adjustment, the results presented in these figures are the computed values unsmoothed. The fluctuation of the computed velocity potential along the chordwise row is greatly reduced. This is achieved by including the contributions from the partial boxes along the leading edges.

Cropped Delta Wing with Zero Thickness

Shown in figure 14 is the convergence of the force coefficients due to plunge (mode 1), f = 1.0, and pitch (mode 2), $f = -0.4034 + 1.832\ \bar{x}$, (b = 0.5833 ft.), at reduced frequency 0.095 with respect to the number of boxes along the root chord of the aspect ratio 1.5 cropped delta wing with taper ratio 0.143. The results computed from the programs of references 1 and 2 are also included for comparison. It is again observed that the results of the present program show a better converging trend than the other versions of the sonic-box computer program.

Arrow Wing with Zero Thickness

The convergence of the force coefficients due to plunge (mode 1) and pitch about its apex (mode 2) at a reduced frequency 0.5 of a pointed tip wing with aspect ratio 4.0 and leading edge sweep angle 56.33 degrees is shown in figure 15. The results computed from the present program and that of reference 2 are included in the figure. The convergence rate as well as the converged values of these two programs are about the same.

Distributions of the velocity potential due to plunge on the wing and in the wake calculated from the present program and that of reference 2 are, respectively plotted in figures 16 and 17. The reduced frequency is 0.5 and the number of boxes used in the chordwise direction is 30. The leading edge adjustment is included in the results shown in fiugre 16, but no smoothing. The results computed from the program in reference 2 shown in figure 17 are the smoothed values and the computed values along 1st, 5th, 10th and 15th chordwise rows. These computed velocity potentials exhibit the similar fluctuation in the chordwise direction as observed in the case of delta wing.

The velocity potential at the trailing edge of the arrow wing is interpolated from the velocity potentials computed at the centers of the last wing box and the first wake box along the same chordwise row. This wake box is assumed to be a wing box in the preliminary computation to facilitate the evaluation of the velocity potential at the trailing edge; this box is treated as a wake box thereafter. In reference 2, the velocity potential in the wake along a chordwise row is computed from the velocity potential at the trailing edge of the wing on the same row before the smoothing is applied. After the smoothing, the velocity potential at the trailing edge may not remain the same as the one originally used in the computation of the velocity potential in the wake. This, in turn, raises the question of the correctness of the velocity potential in the wake which was used in the computation of the velocity potential on the portion of the wing downstream of the wake in question. The problem area of the computer program in reference 2 described here is depicted in figure 17. In the present program, this does not become a problem because the computed velocity potentials in the chordwise direction are fairly smooth and no additional smoothing is applied.

Rectangular Wing

Zero Thickness. - The results for a rectangular wing of aspect ratio 2.0 performing plunge (mode 1) and pitch about its leading edge (mode 2) at reduced frequency 0.6 are presented in figures 18, 19, and 20. In figure 18, the convergence of the different sonic-box programs is shown. The convergence of the present program again showed a better trend than the other versions of sonic-box method computer program. Since the calculations were made only up to 20 boxes along the root chord, the converged values of the other programs were not yet attained. The difference in the values of force coefficients of the present method and that of reference 3 might have been due to the different data fitting techniques used in the programs. The different ways of treating the wing leading edge in these two programs do not have effect on the results for this wing because there is no partial box involved in the computation.

Nonzero Thickness. - The convergence of the present program and that of reference 3 for both zero and nonzero thickness is shown in figure 19. This rectangular wing has a biconvex airfoil section with a thickness ratio 0.0521 and the steady-state Mach number distribution on the wing was calculated with the method of reference 9. The thickness has greater effect on the rectangular wing than on the delta wing (fig. 7). This may be caused by the fact that the Mach number on the rectangular wing (fig. 21) deviates from unity more than that on the delta wing. For the nonzero thickness, in addition to the differences in data fitting technique and handling of wing leading edge, the method of computing the force coefficients is also different. In the present method, the force-coefficient integration is performed in the transformed space, whereas the method in reference 3 integrates in the physical space.

Variation of the force coefficients with respect to reduced frequency is plotted in figure 20. Both zero and nonzero thickness using the present program and that of reference 3 with 20 boxes along the root chord are shown. The results of reference 6 for the zero thickness wing are included in figures 20(c) and (d) for comparison. Agreement between the present results and that of reference 6 is better in the medium frequency range. This may be attributed to the different assumptions imposed on the parameters involving reduced frequency. The results of reference 6 are based on a theory for large values (greater than 0.8) of the product of aspect ratio and the square root of reduced frequency, whereas, the present program is based on the assumption of small values of the product of box-side length and reduced frequency. With the sonic-box method, one needs to use more boxes to represent the wing as the frequency increases.

REMARKS AND RECOMMENDATIONS

A sonic-box method computer program using the concept of local linearization to account for the effects of wing thickness in unsteady sonic flow is presented. This program uses a coordinate transformation to reduce the small perturbation, unsteady transonic velocity potential equation, which has variable coefficients, to a linear equation with constant coefficients and solves this boundary-value problem in the transformed space by the sonic-box method by regarding the mean local Mach number as a parameter.

Accuracy of this approach deteriorates when the variation of Mach number on the physical wing is large. In fact, the method fails when the lateral variation of Mach number becomes large enough to cause multivalued transformation which implies wing-surface fold-over in the transformed space. Moreover, difficulties can also occur when artifical "wakes" created by the transformation impinge upon downstream portion of the lifting surface. Specifically, these "wakes" exist if yM along leading edge (or tip) at a particular chordwise location is greater than that at another location further downstream. Existence of these artificial "wakes" causes no difficulty, however, unless a portion of the transformed wing further downstream protrudes into these "wakes". Therefore, the user of this program should examine the geometry of the transformed wing to determine if these conditions exist.

This program includes the contribution of the partial boxes along the wing leading edge in the calculation of velocity potential, thus, allows one to use a smaller number of boxes to represent the wing than other versions of sonic-box computer program do. The earlier polynomial-surface data-fitting technique has been replaced by the cubic spline and the spline-surface to improve the accuracy.

Since the box method itself is numerical in nature, the distribution of the calculated values is not always smooth, that is, some degree of data oscillation is unavoidable. This means that some smoothing of the computed data will be desirable not only because of the oscillation of the computed data but also because of the very nature of cubic spline and the spline-surface techniques. Even though a smoothing scheme has been included in the program, it was not utilized in the samples presented. The adequacy of this smoothing scheme has not been extensively tested and the differentiation of the computed values has been avoided in the calculation of generalized aerodynamic force coefficients. It is quite possible to lose the desired feature of the interpolating surface after the smoothing unless a proper control of the smoothing process is employed.

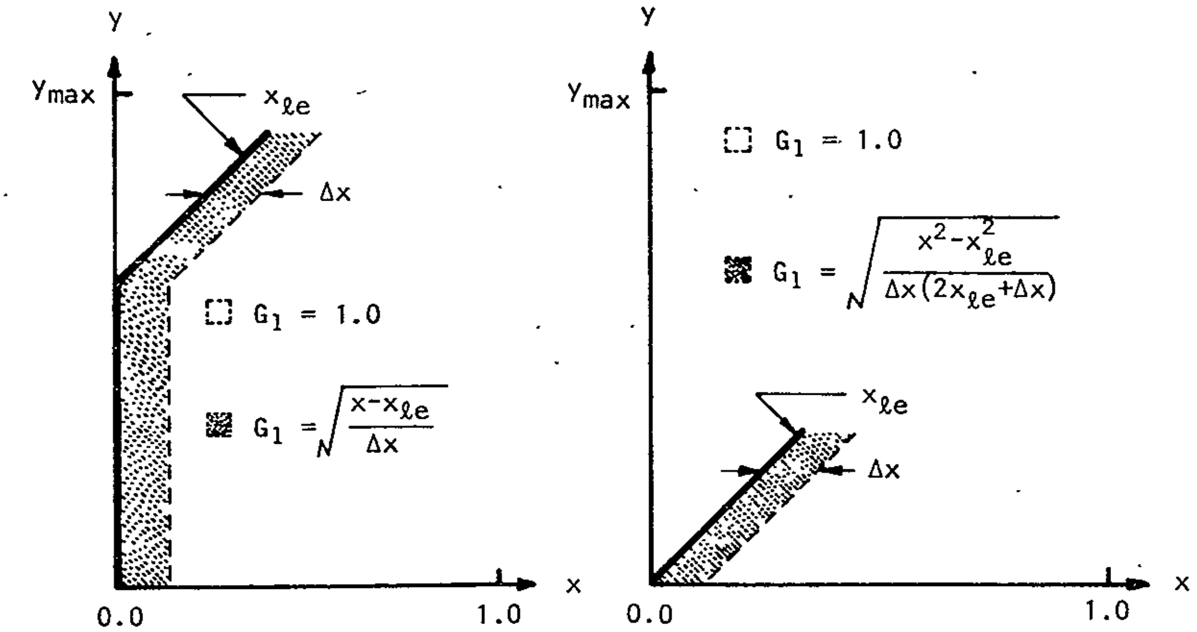
In the program, the vibration mode shape and Mach number are represented by spline-surface. The accuracy of the interpolating surface can be affected by the choice of the data points used to generate it. To facilitate an assessment of the accuracy of the fitting technique, nonzero thickness and flexible modes were not emphasized. Instead, zero thickness and rigid body modes were studied more extensively in the samples presented. Emphasis was placed on the consistency of the converging trend with the present program. According to the results presented, the thickness effect on the rectangular wing is quite significant, as much as 30%, but smaller for the delta wing. This is attributed to the difference in the Mach number variation for the two wings.

Since the accuracy of the present program depends on the data-fitting techniques used in the program, it would be most desirable to modify or change the fitting scheme so that the flexible mode shape and the Mach number distribution can be better represented and interpolated during the computation.

One method to accomplish this is to generate accurate spline-surface for Mach number and each mode shape from a separate program and use these as part of the input. Another method is to input Mach number and deflection of every mode at each box center on the physical wing without fitting the data with surfaces and use the cubic spline to facilitate the interpolation required during the computation. Either of these two methods undoubtly will increase the computation time, but the additional cost may be compensated with the use of a smaller number of boxes to represent the wing.

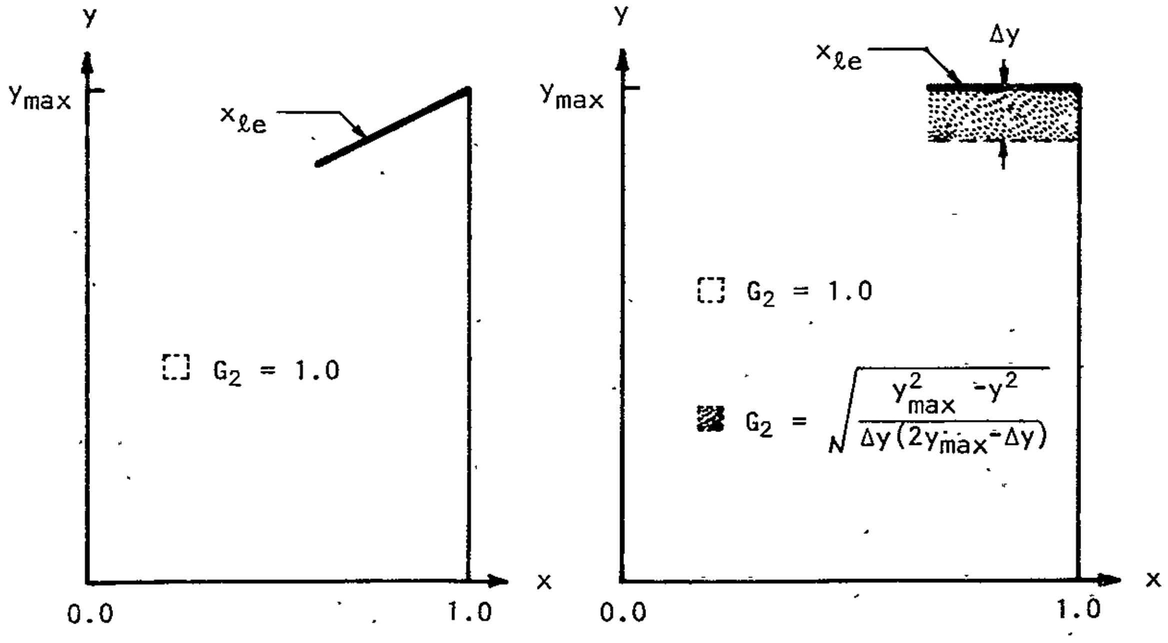
REFERENCES

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 Porgram). NASA CR-132477, Sept. 1974.
4. Ruo, S. Y.; and Theisen, J. G.: Calculation of Unsteady Transonic Aerodynamics for Oscillating Wings with Thickness. NASA CR-2259, June 1975.
5. Olsen, J. J.: Demonstration of a Transonic Box Method for Unsteady Aerodynamics of Planar Wings. AFFDL-TR-66-121, Oct. 1966, Air Force Flight Dynamics Lab., Wright-Patterson Air Force Base, Ohio.
6. Landahl, M. T.: *Unsteady Transonic Flow.* Pergamon Press, New York, 1961.
7. Desmarais, R. N.; and Bennett, R. M.: Computer Programs for Plotting Curves with Various Dashed-Line Sequences. NASA TM X-2465, Feb. 1972.
8. Harder, R. L.; and Desmarais, R. N.: Interpolation Using Surface Splines. *Journal of Aircraft,* Vol. 9, No. 2, February 1972, pp. 189-191.
9. 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.
10. Ruo, S. Y.; Yates, E. C., Jr.; and Theisen, J. G.: Calculation of Unsteady Transonic Aerodynamics for Oscillating Wings with Thickness. *Journal of Aircraft,* Vol. 11, No. 10, Oct. 1974, pp. 601-608.



(a) $y \neq 0$ at $x=0.0$

(b) $y=0$ at $x=0.0$



(c) $\frac{dy}{dx} \neq 0$ at $x=1.0$

(d) $\frac{dy}{dx} = 0$ at $x=1.0$

Figure 2. - Shapes of leading edge.

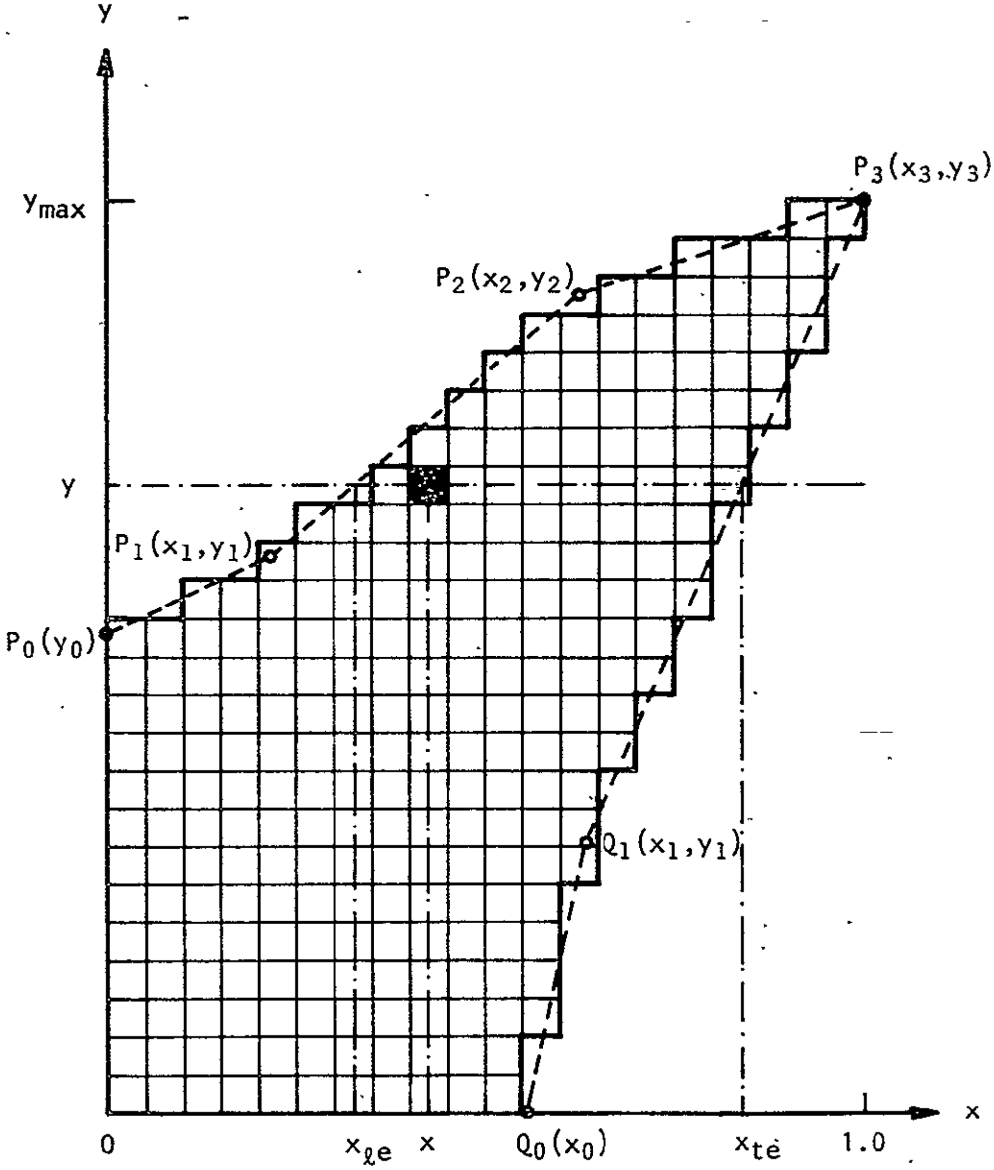
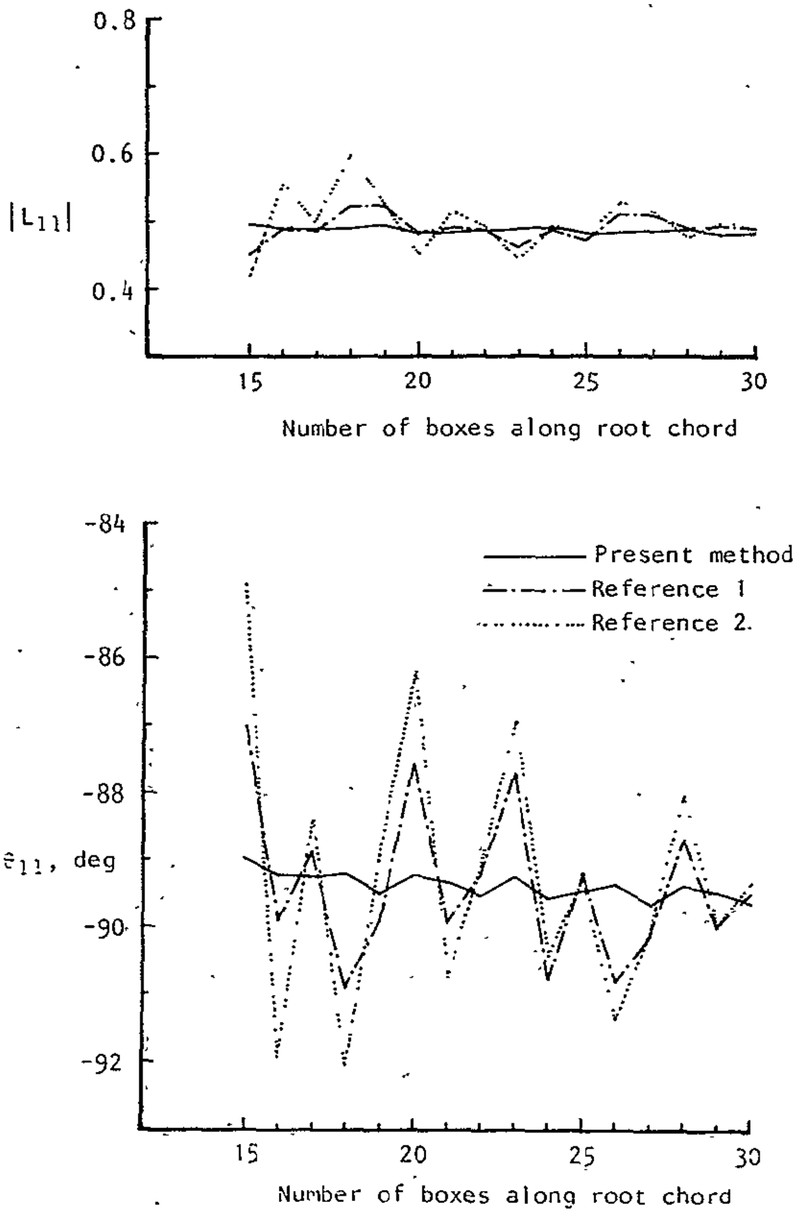
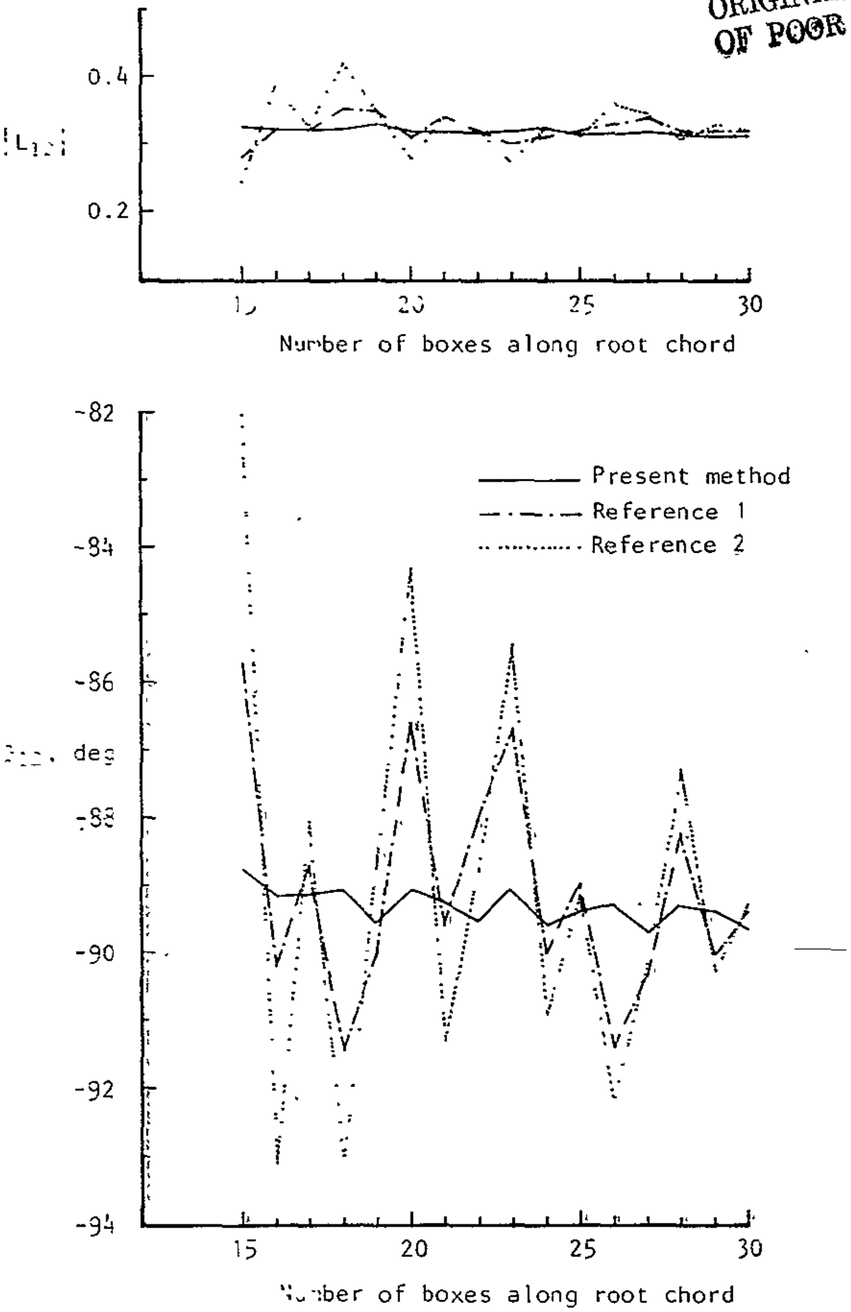


Figure 3. - Half wing geometry.



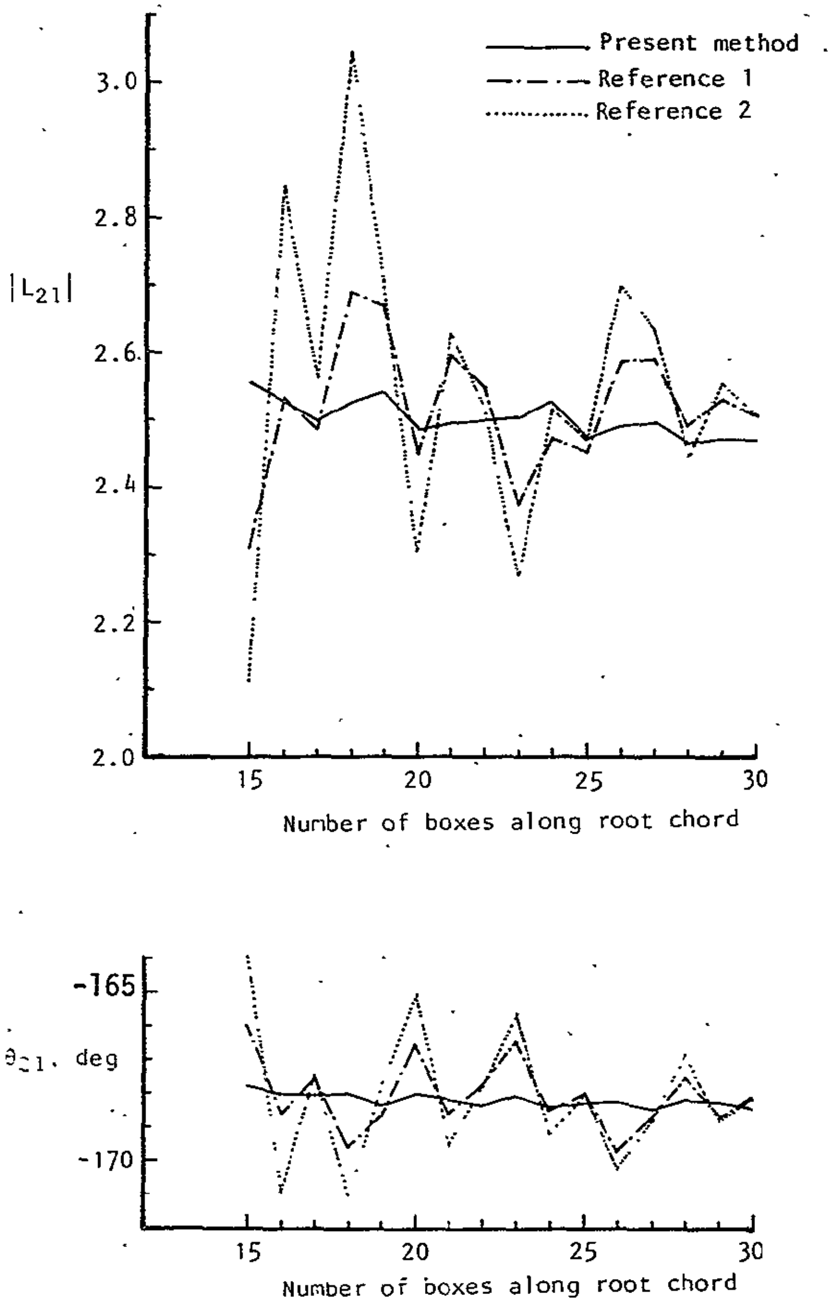
(a) Lift due to plunge.

Figure 4. - Convergence of force coefficients due to plunge and pitch for delta wing of aspect ratio 1.5 at k = 0.2. Zero thickness.



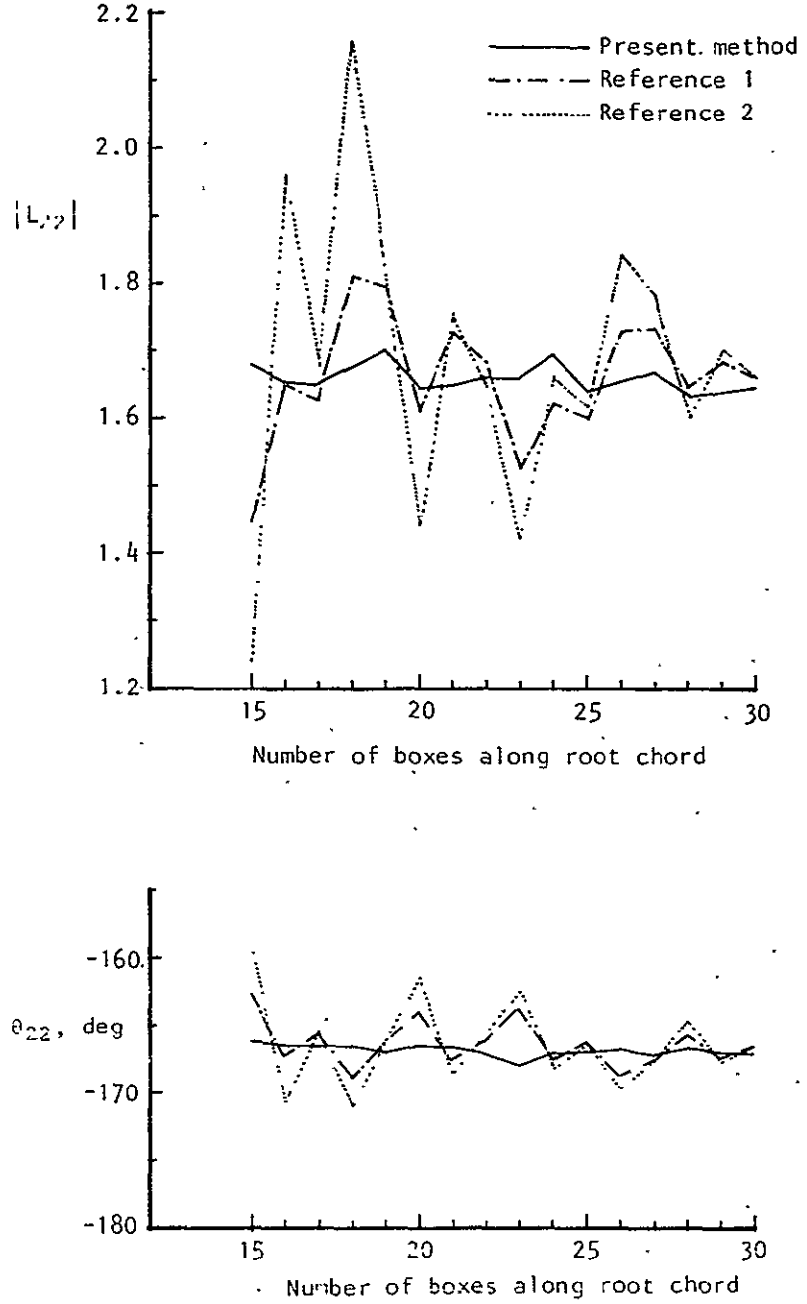
(b) Moment due to plunge.

Figure 4. - Continued.



(c) Lift due to pitch.

Figure 4. - Continued.



(d) Moment due to pitch.

Figure 4. - Concluded.

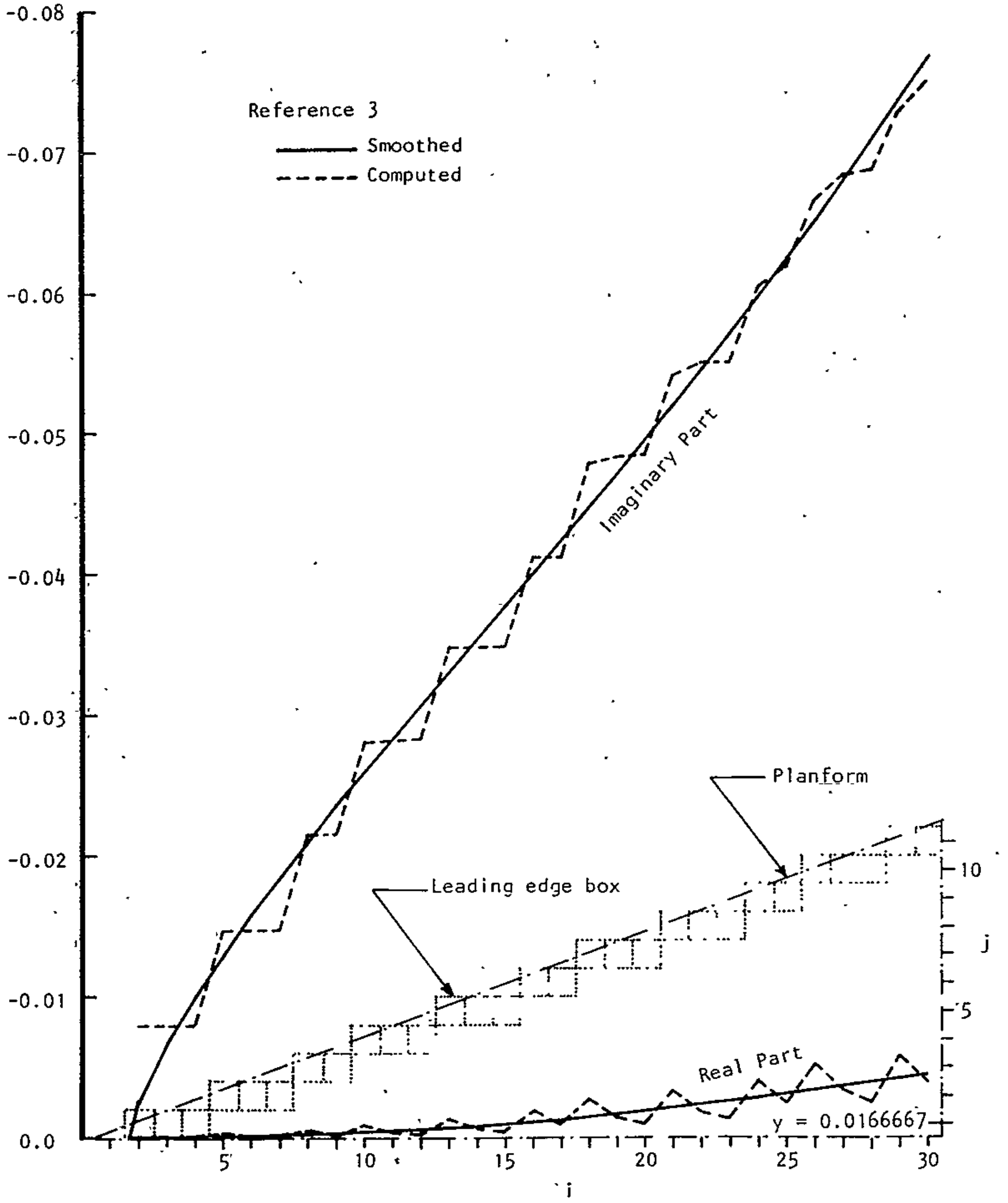
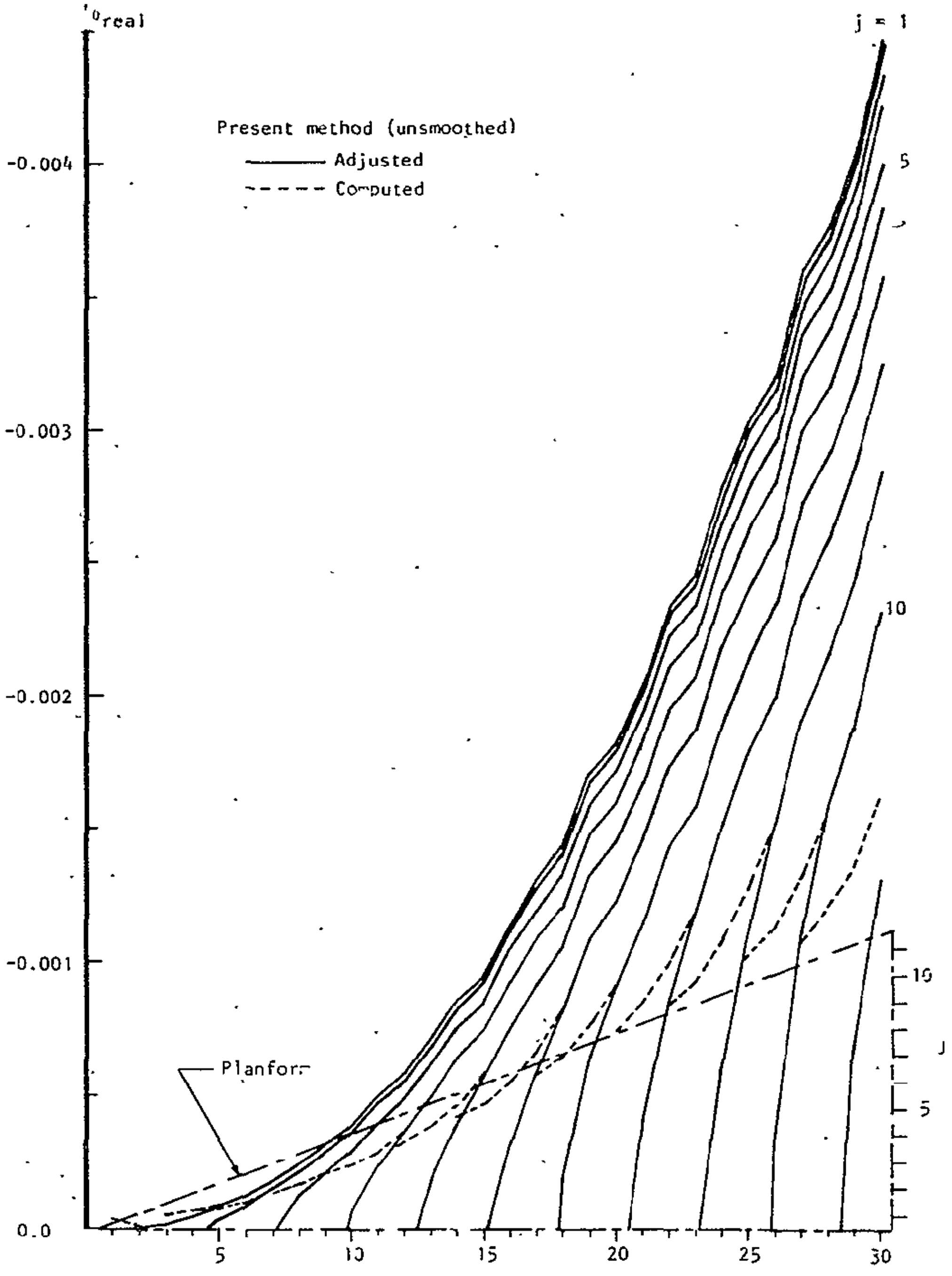
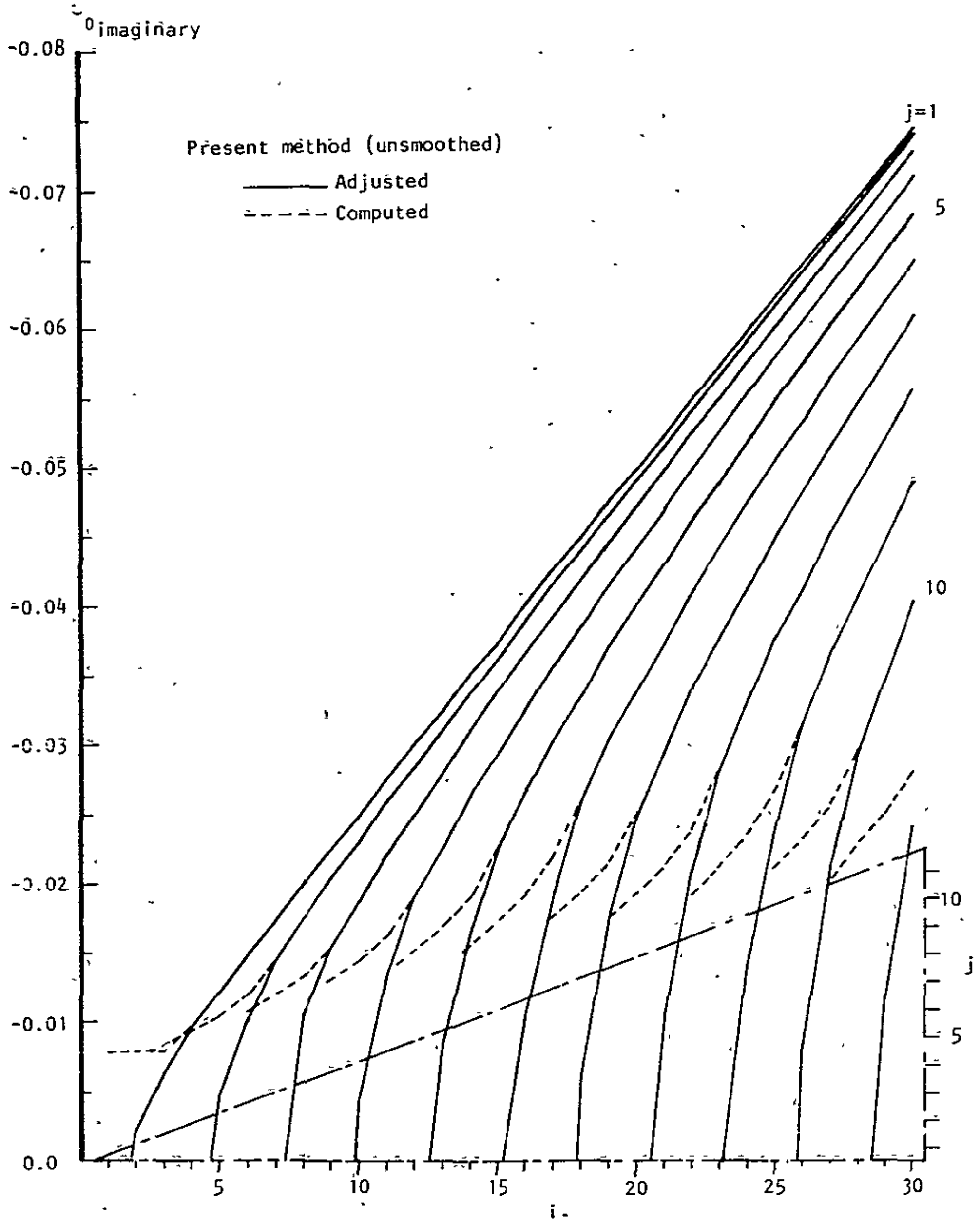


Figure 5. - Velocity potential along chordwise row at y = 0.0166667 due to plunge for delta wing of aspect ratio 1.5 at k = 0.2 with 30 boxes along root chord. Zero thickness.



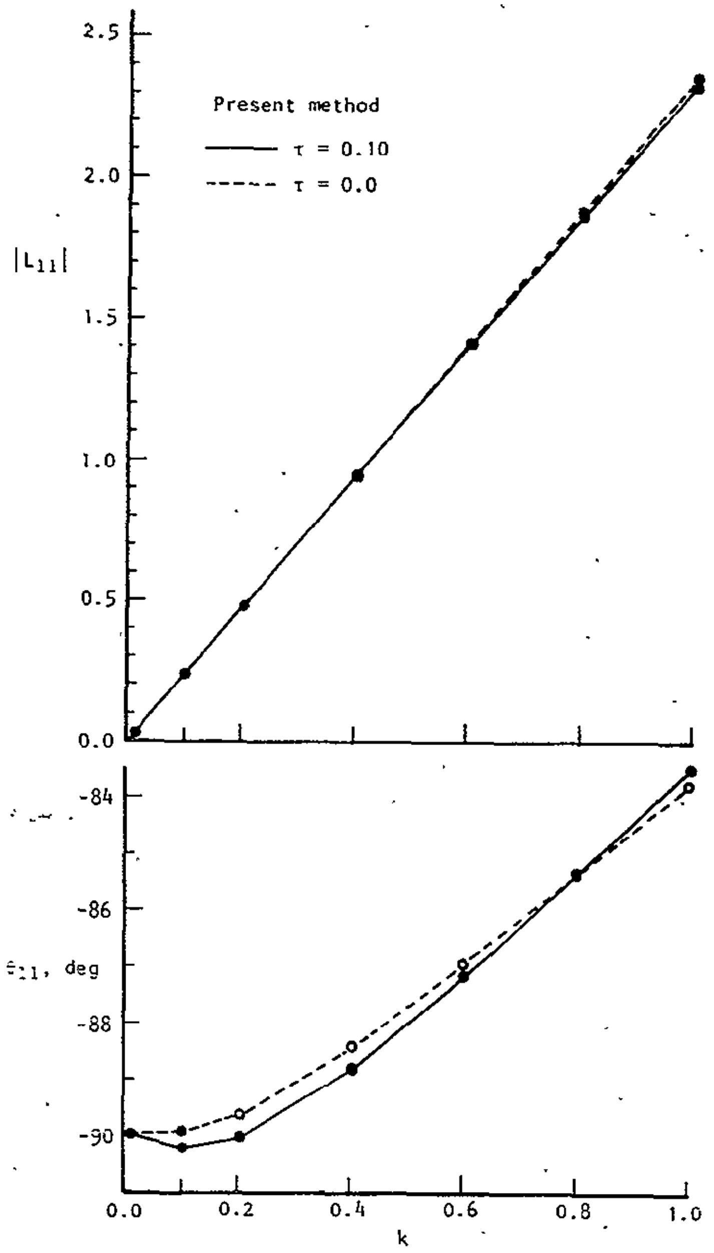
(a) Real part

Figure 6. - Velocity potential along chordwise rows due to plunge for delta wing of aspect ratio 1.5 at k = 0.2 with 30 boxes along root chord. Zero thickness.



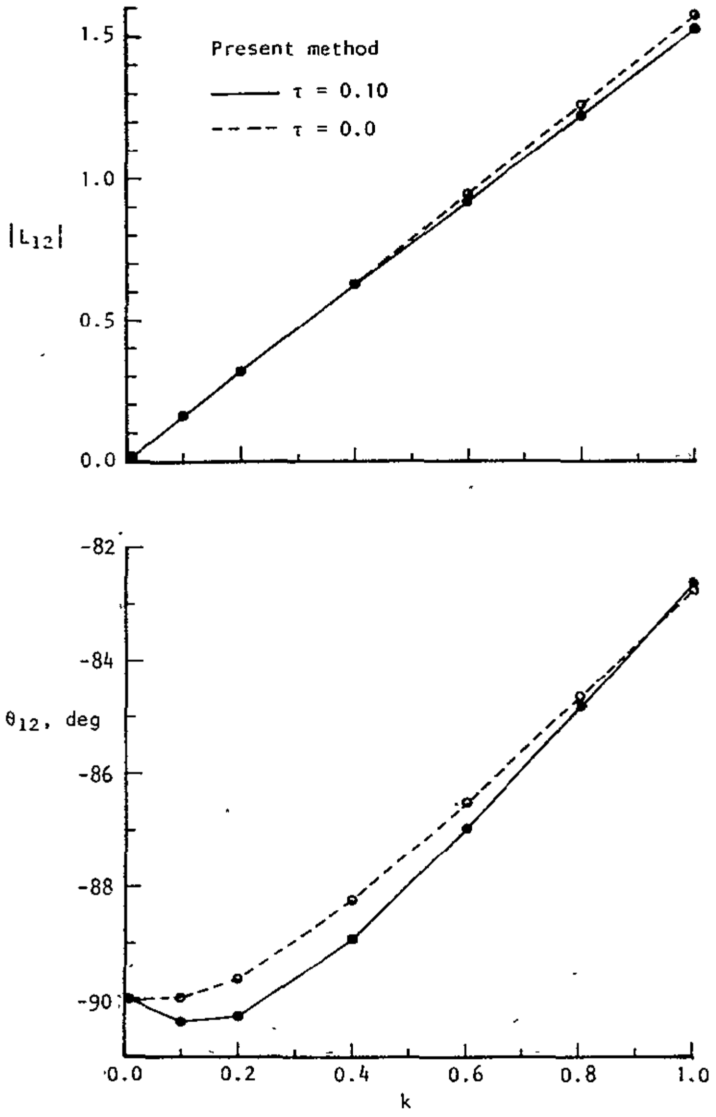
(b) Imaginary part

Figure 6. - Concluded.



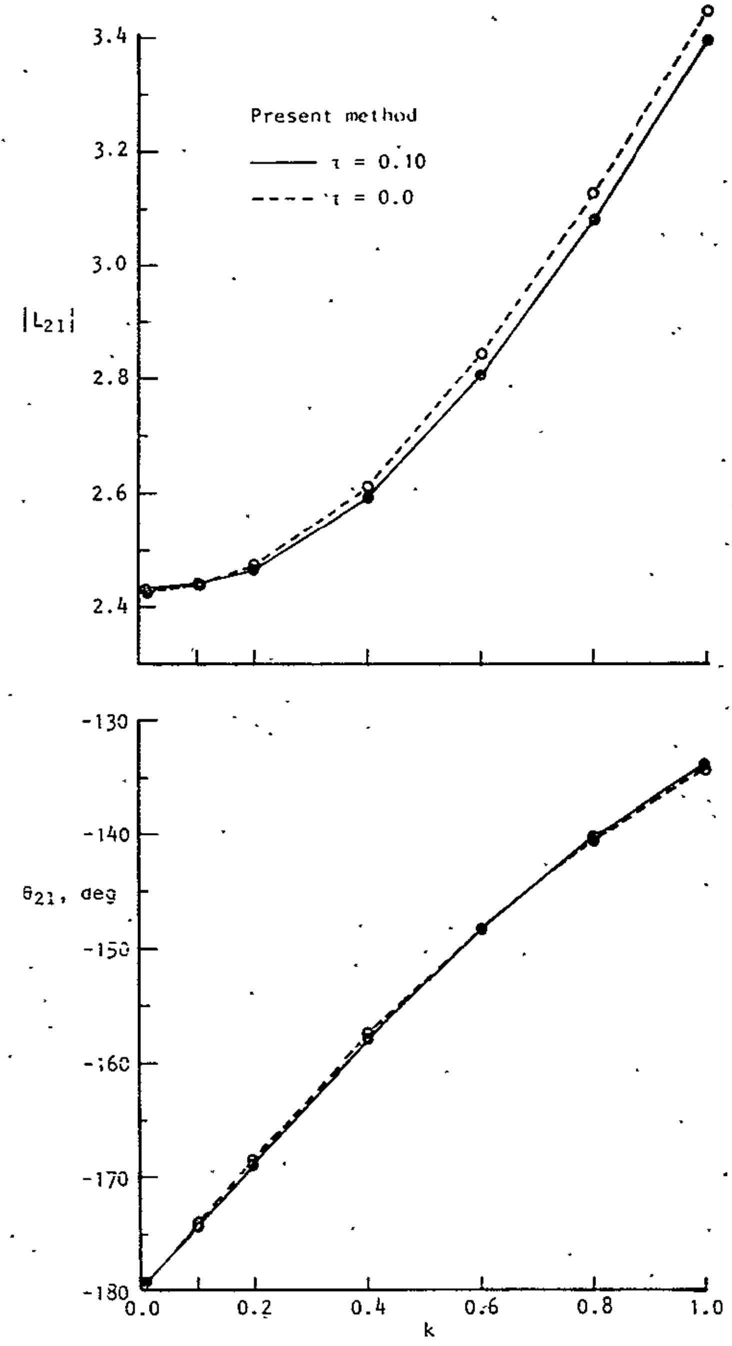
(a) Lift due to plunge.

Figure 7. - Force coefficients due to plunge and pitch for delta wing of aspect ratio 1.5 with 30 boxes along root chord.



(b) Moment due to plunge.

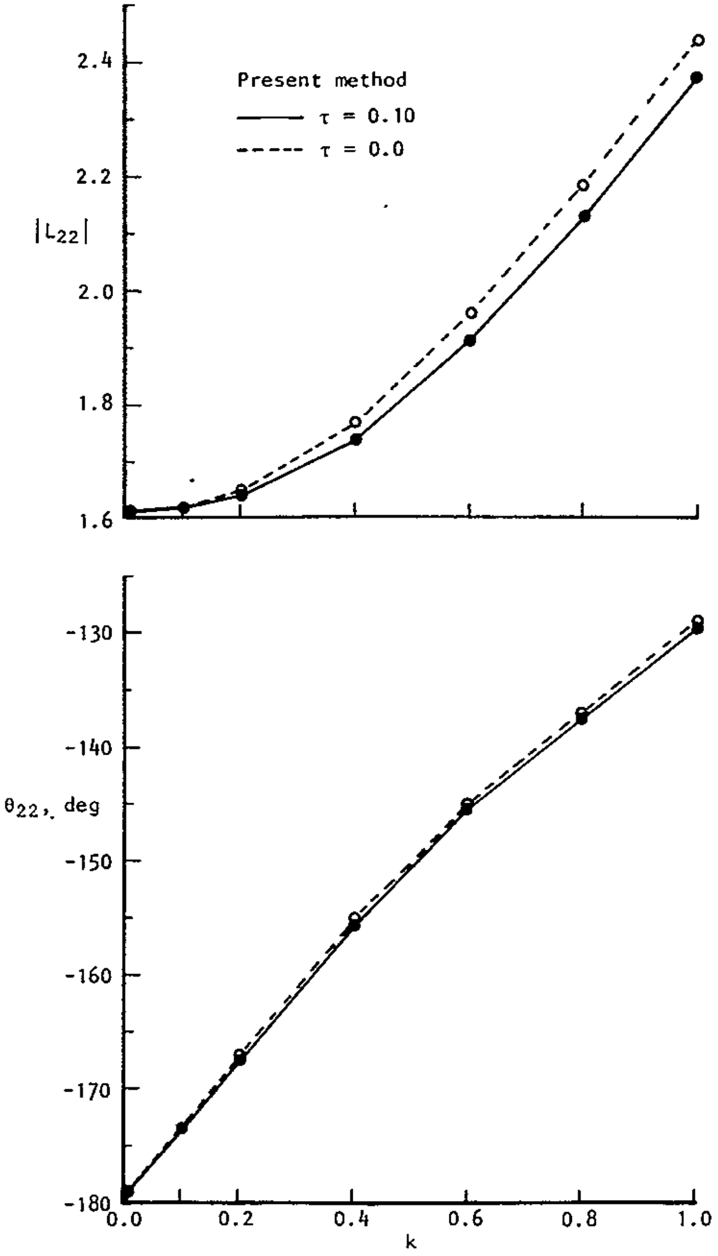
Figure 7. - Continued.



ORIGINAL PAGE IS
OF POOR QUALITY

(c) Lift due to pitch.

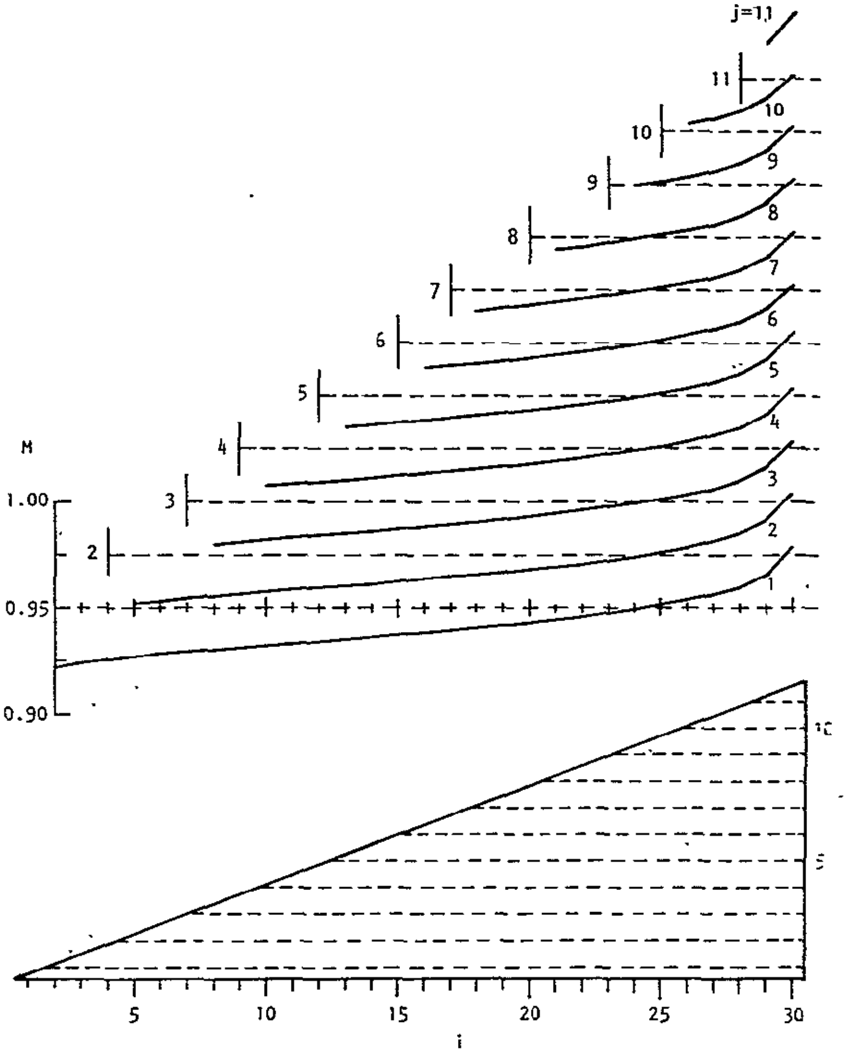
Figure 7. - Continued.



(d) Moment due to pitch.

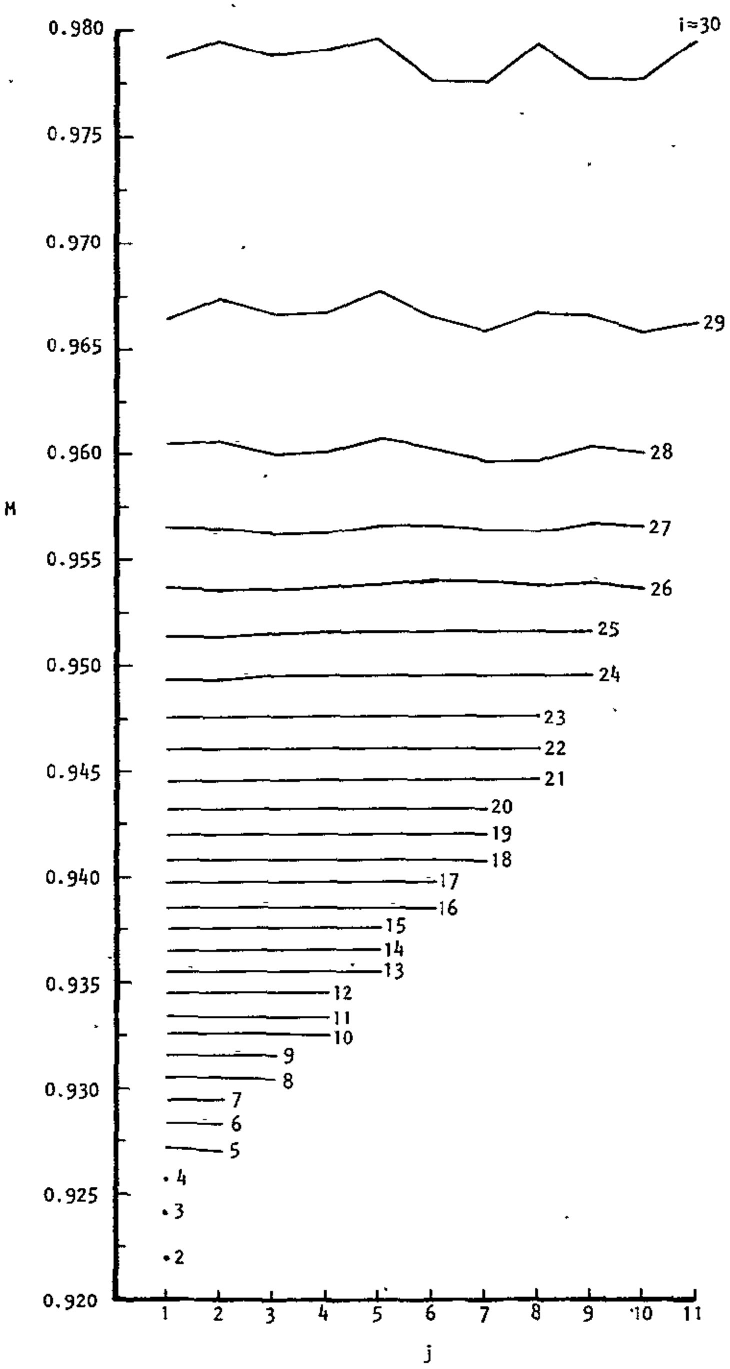
Figure 7. - Concluded.

ORIGINAL PAGE IS
OF POOR QUALITY



(a) Chordwise distribution.

Figure 8. - Interpolated Mach number for $\tau = 0.10$ delta wing of aspect ratio 1.5.



(b) Spanwise distribution.

Figure 8. - Concluded.

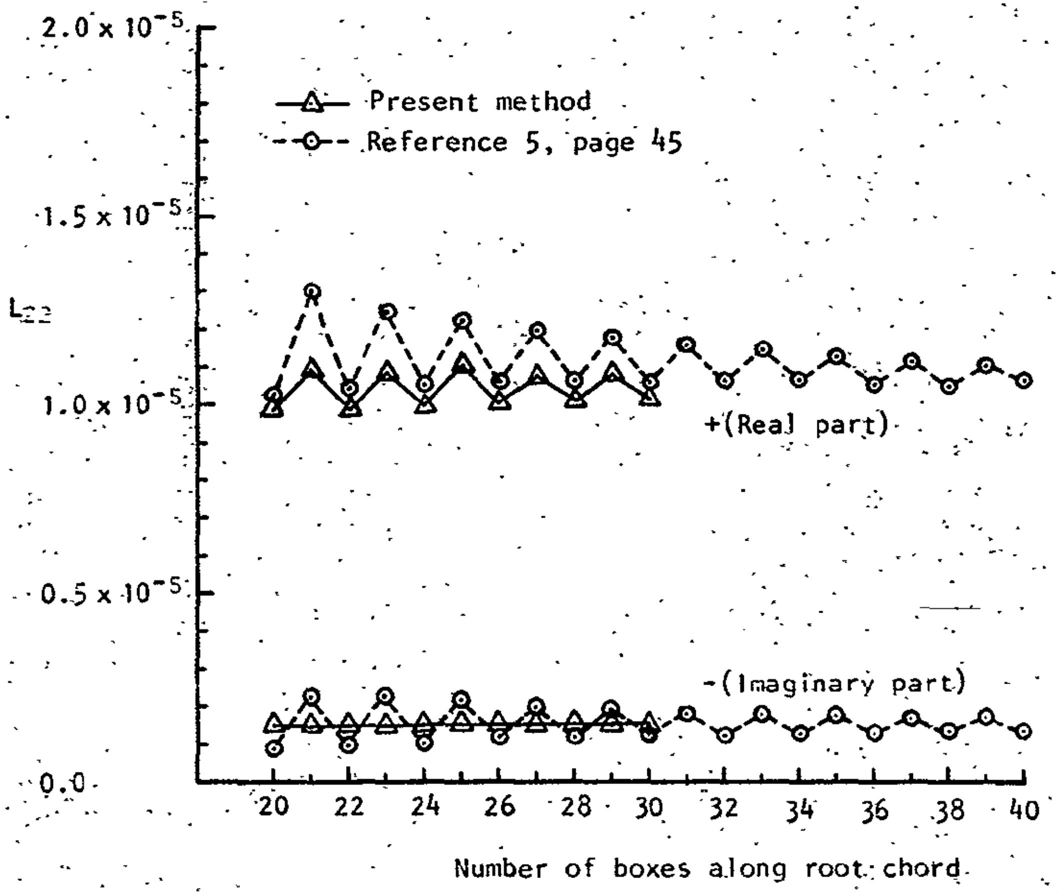
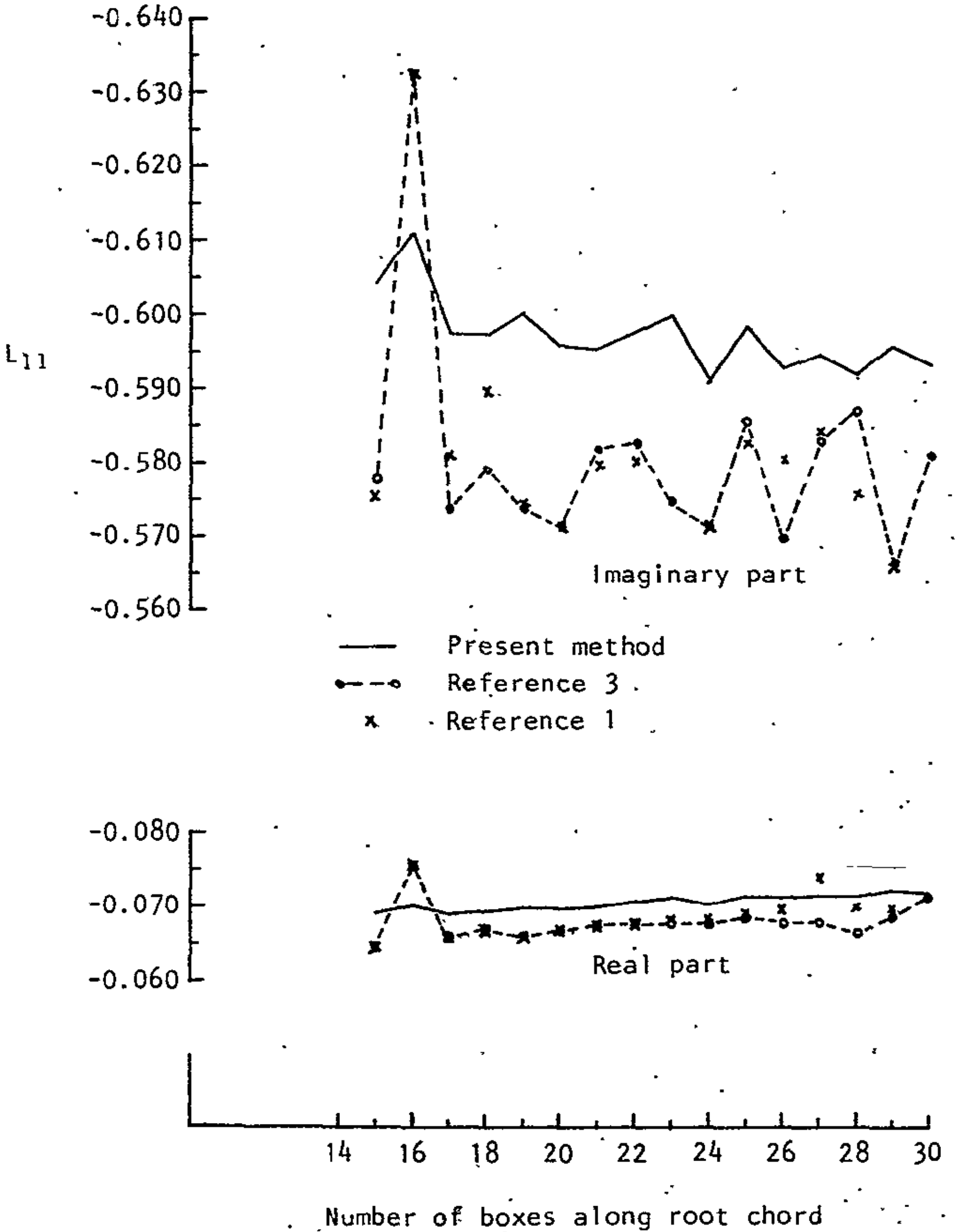
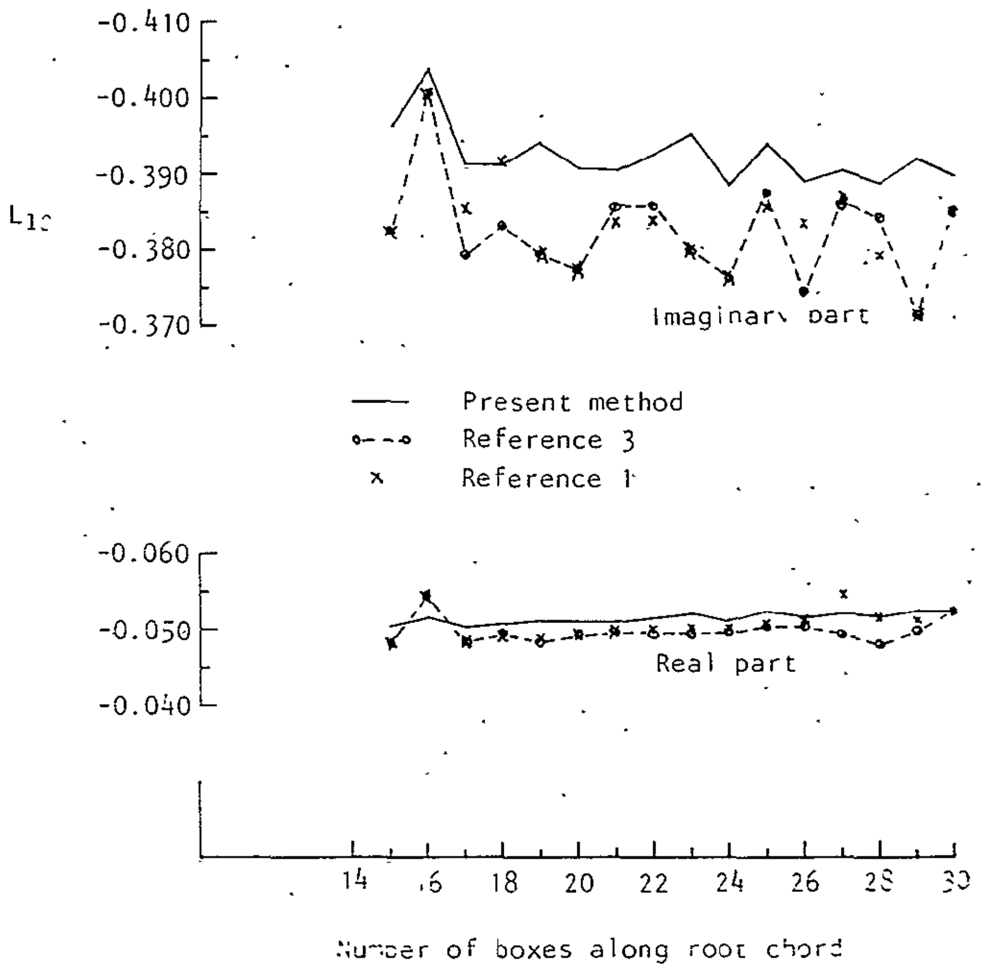


Figure 9. - Convergence of moment due to pitch for delta wing of aspect ratio 2.0 at k = 0.106. Zero thickness.



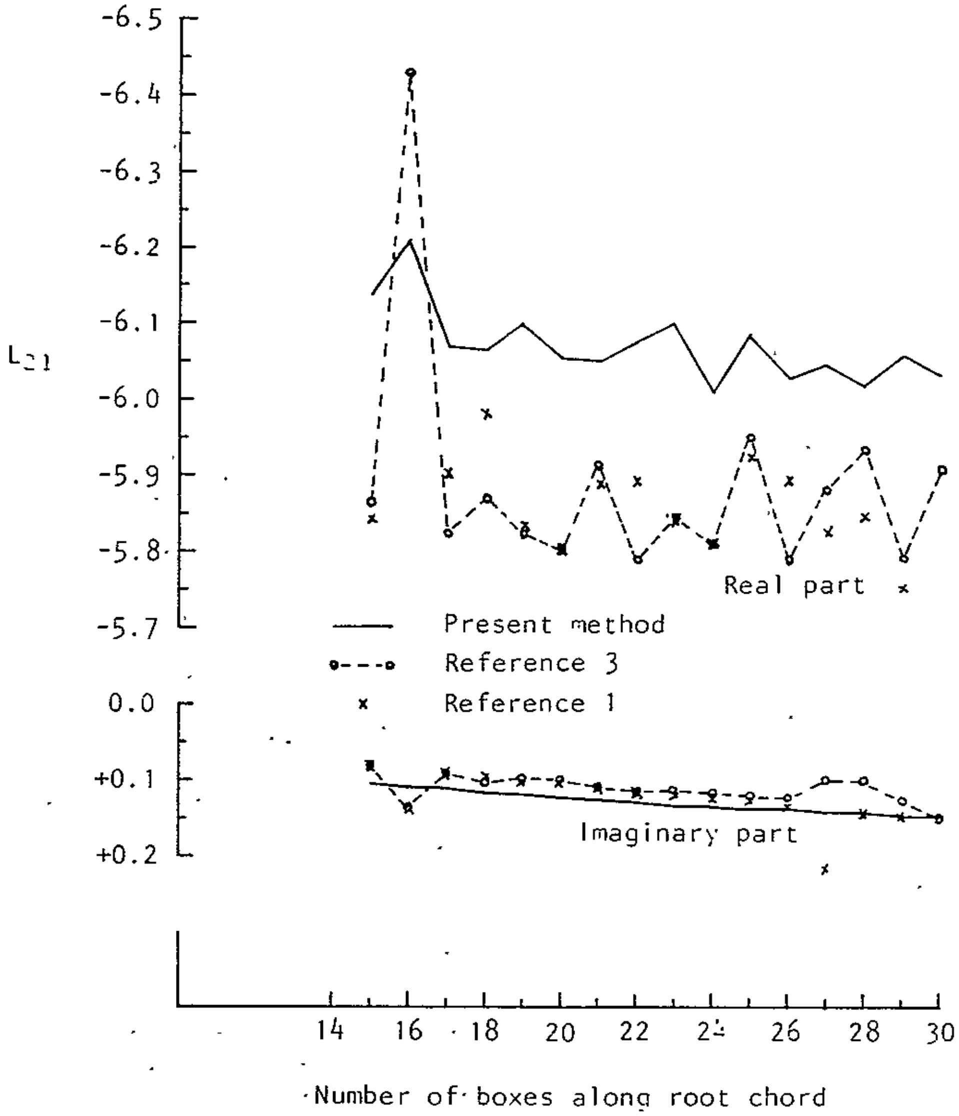
(a) Lift due to plunge.

Figure 10. - Convergence of force coefficients due to plunge and pitch for delta wing of aspect ratio 4.0 at $k = 0.1$. Zero thickness.



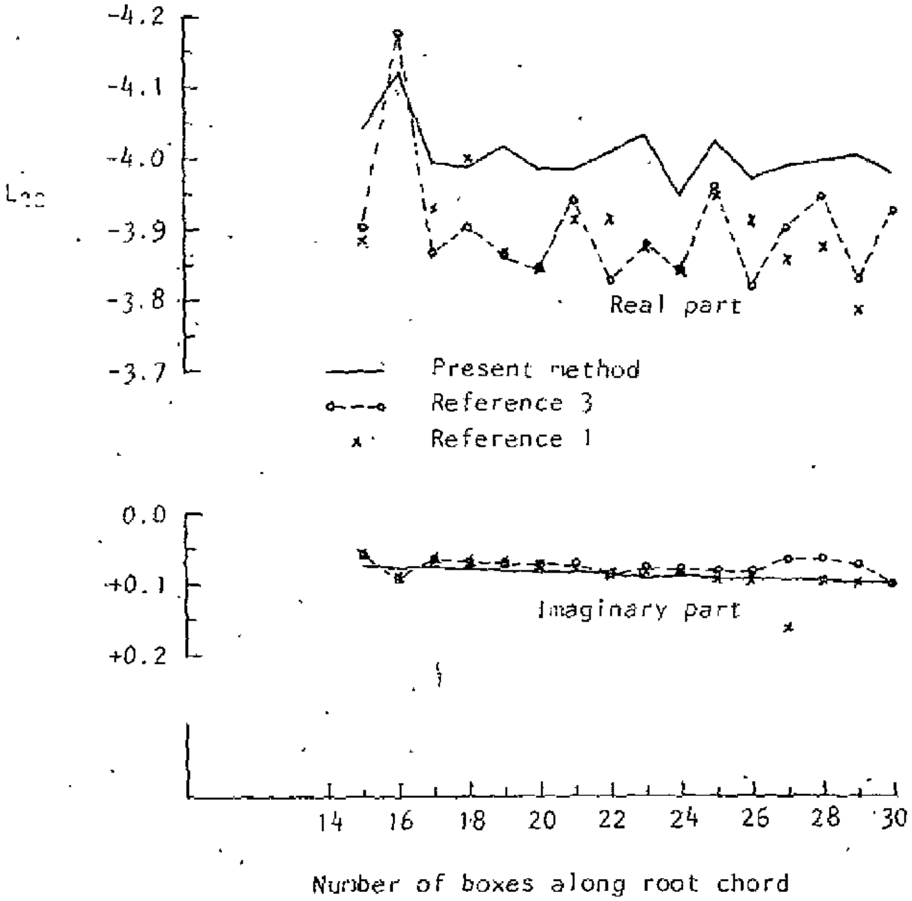
(b) Moment due to plunge.

Figure 10. - Continued.



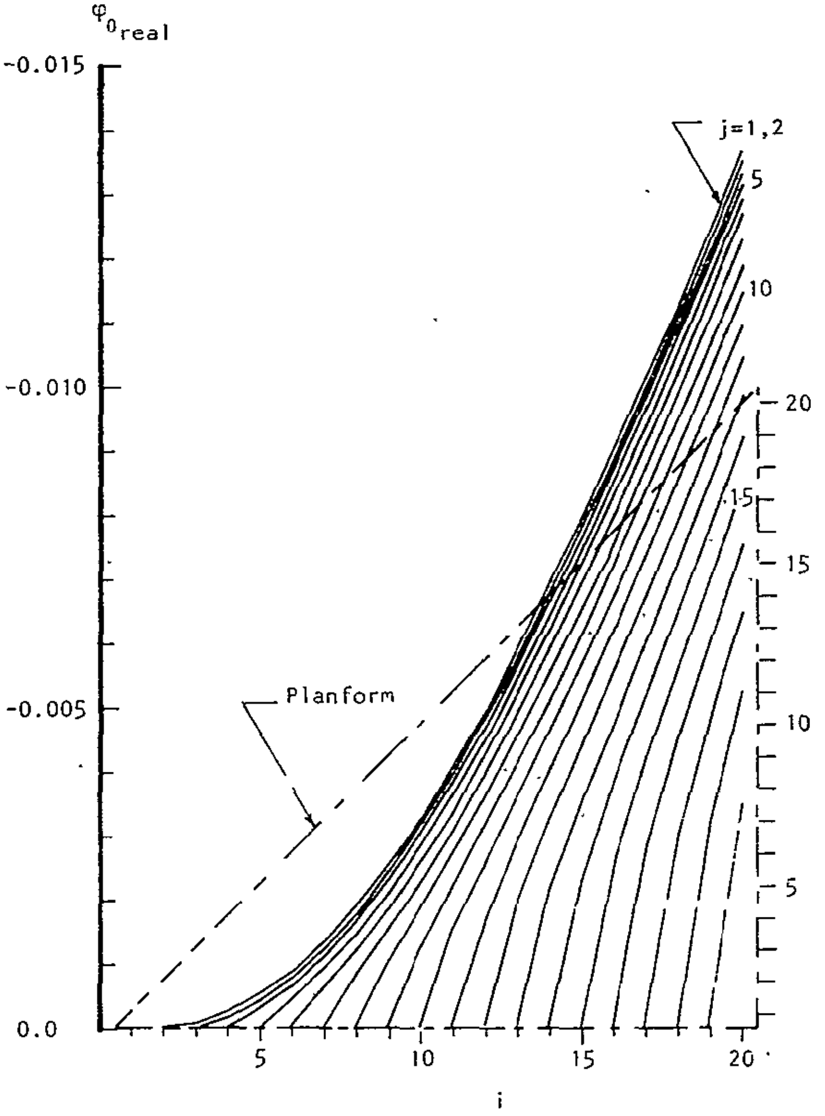
(c) Lift due to pitch.

Figure 10. - Continued.



(d) Moment due to pitch.

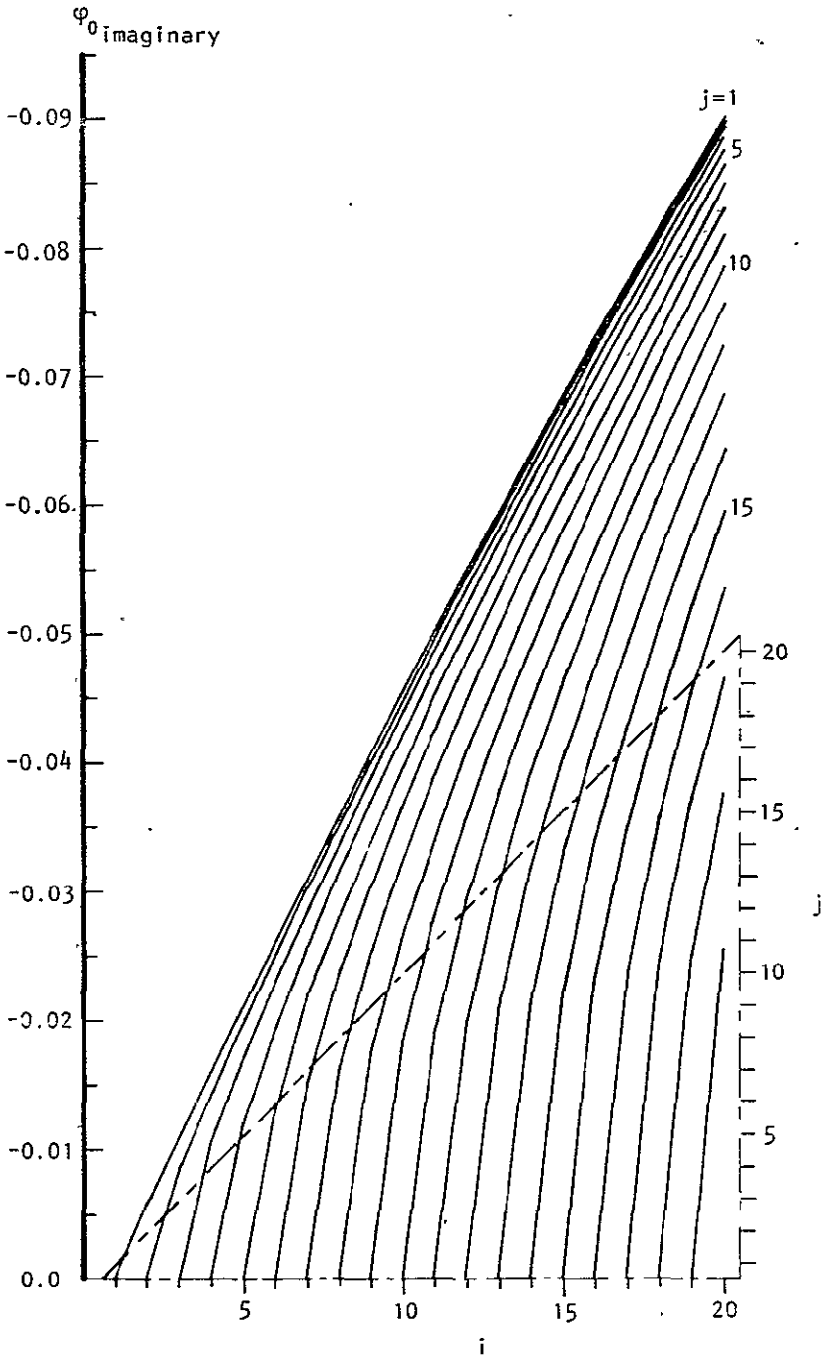
Figure 10. - Concluded.



(a) Real part.

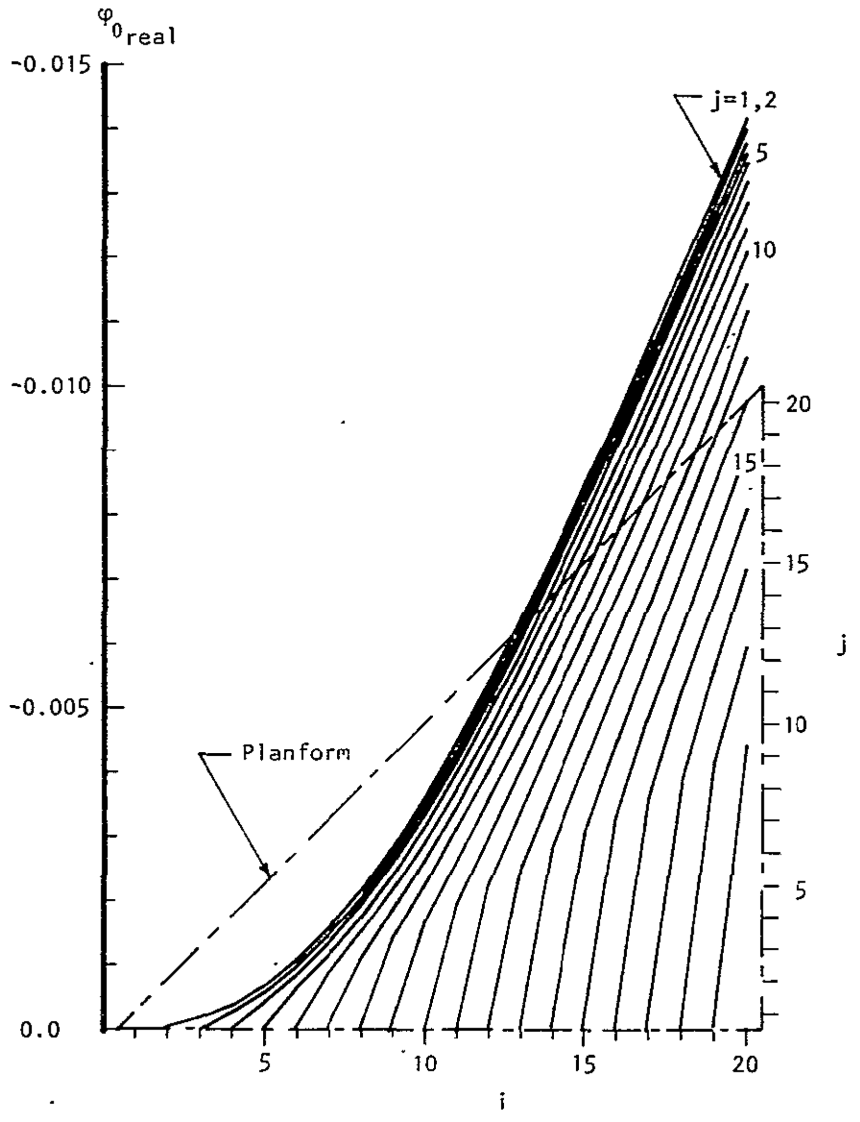
Figure 11. - Velocity potential along chordwise rows due to plunge for delta wing of aspect ratio 4.0 at $k = 0.1$ with 20 boxes along root chord -- reference 1. Zero thickness.

ORIGINAL PAGE IS
OF POOR QUALITY



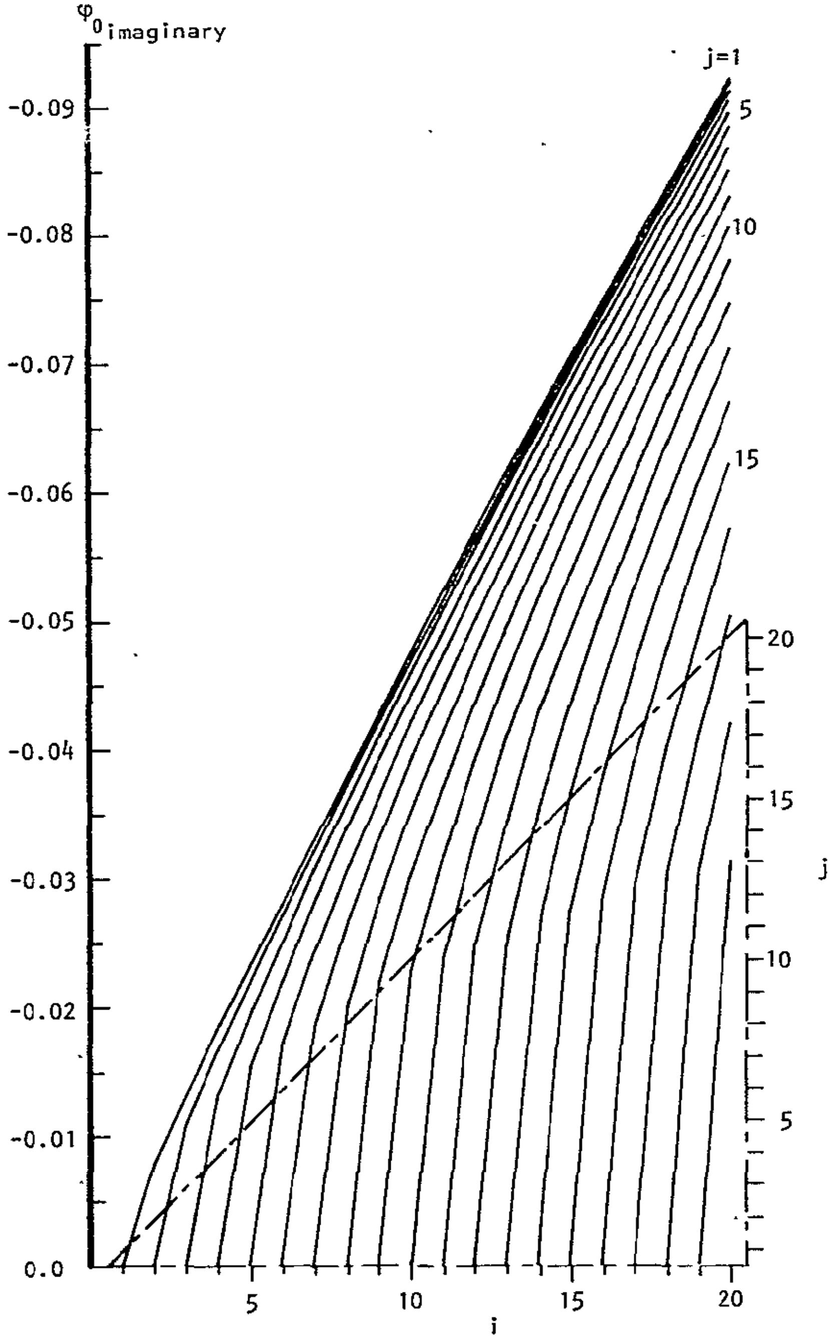
(b) Imaginary part.

Figure 11. - Concluded.



(a) Real part.

Figure 12. - Velocity potential along chordwise rows due to plunge for delta wing of aspect ratio 4.0 at $k = 0.1$ with 20 boxes along root chord -- present method. Zero thickness.



(b) Imaginary part.

Figure 12. - Concluded.

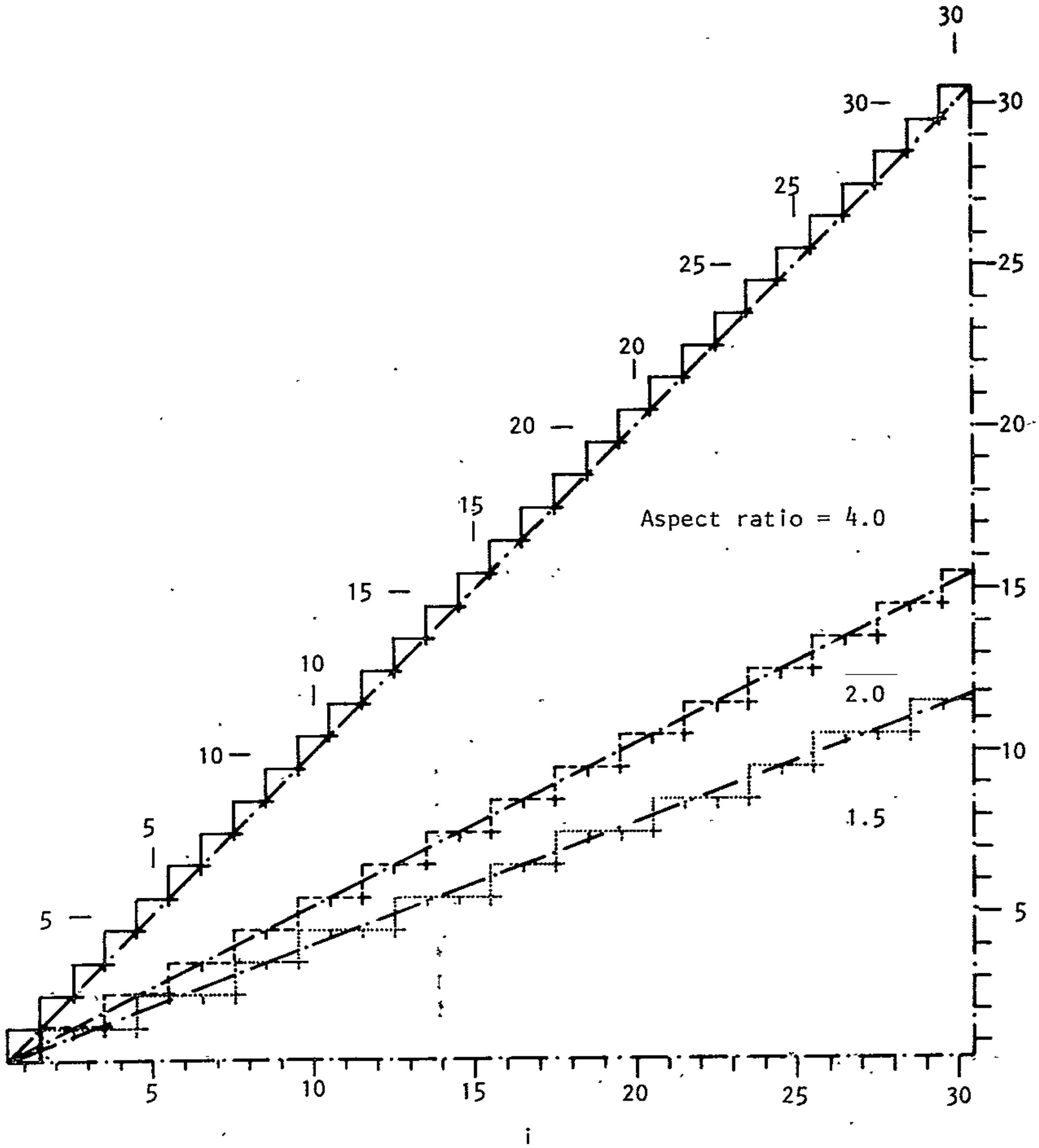


Figure 13. - Box arrangement along swept leading edge of delta wing.

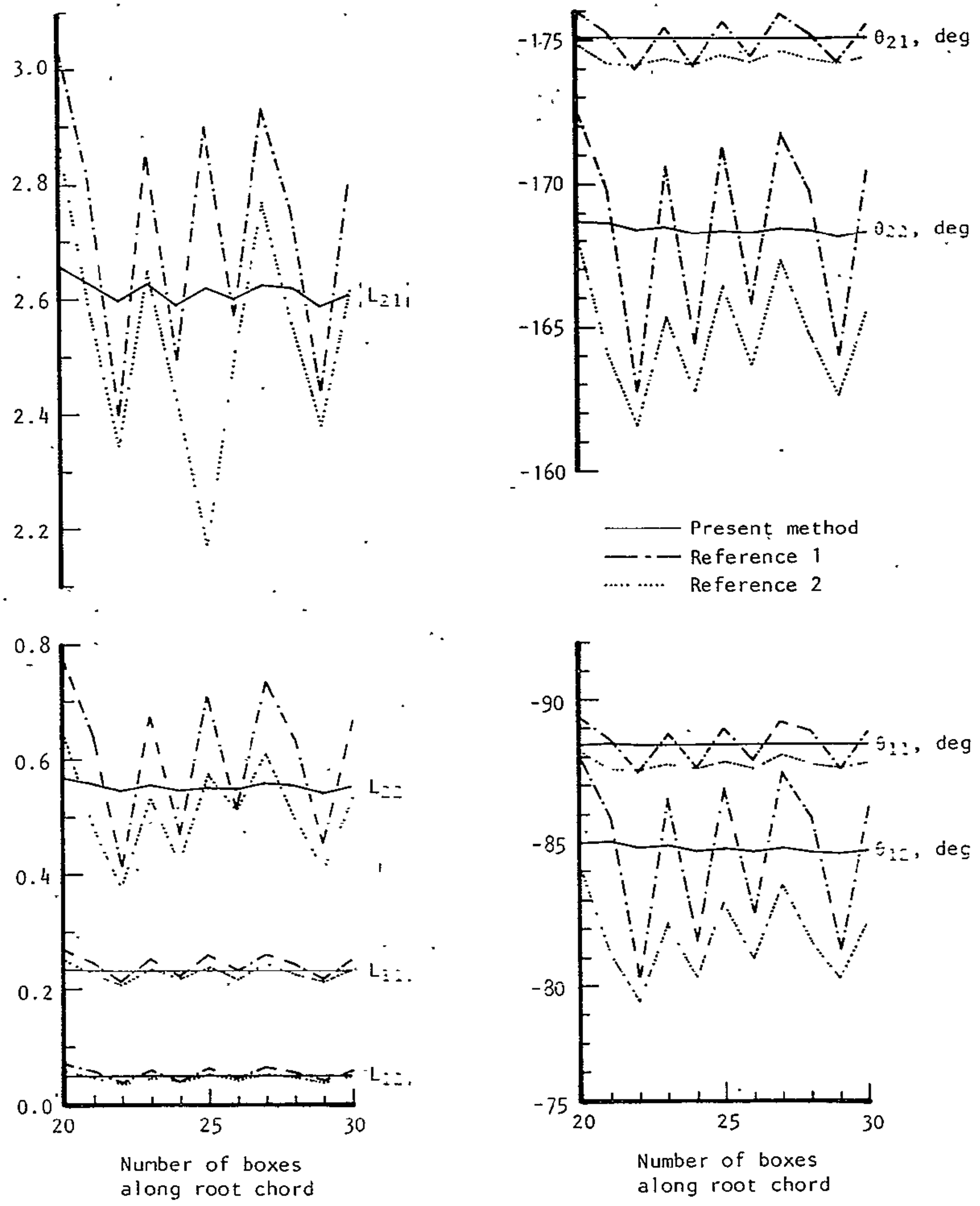
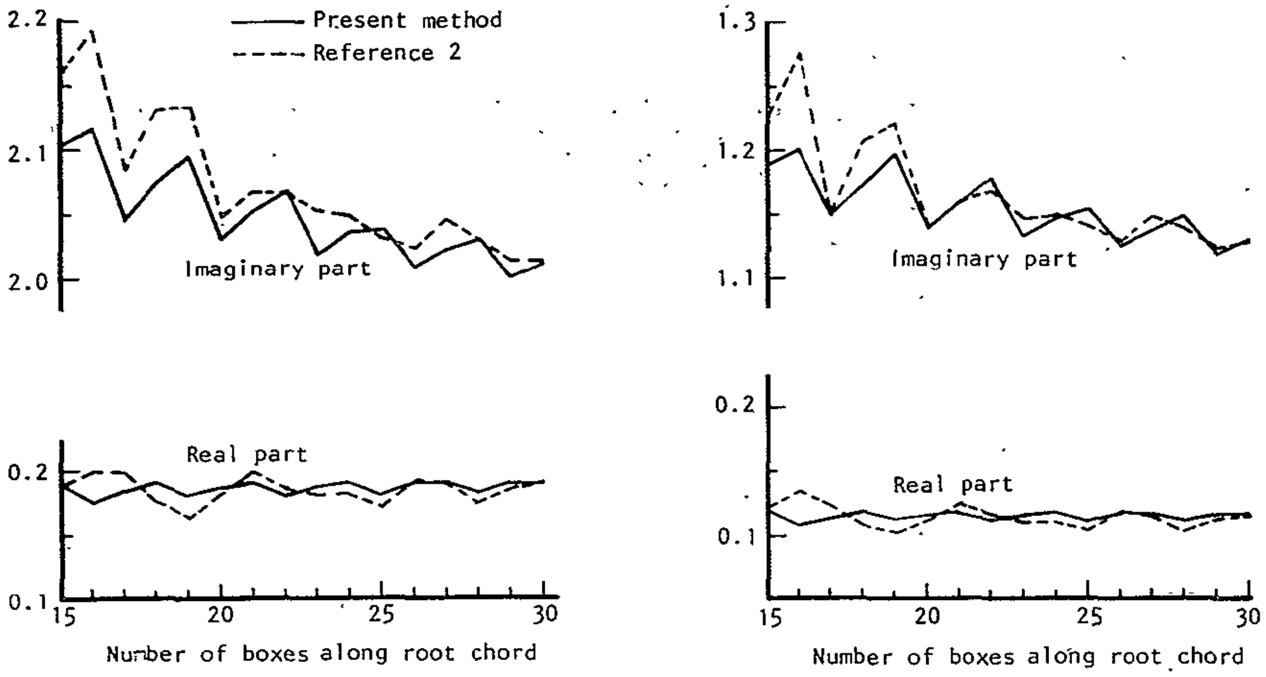
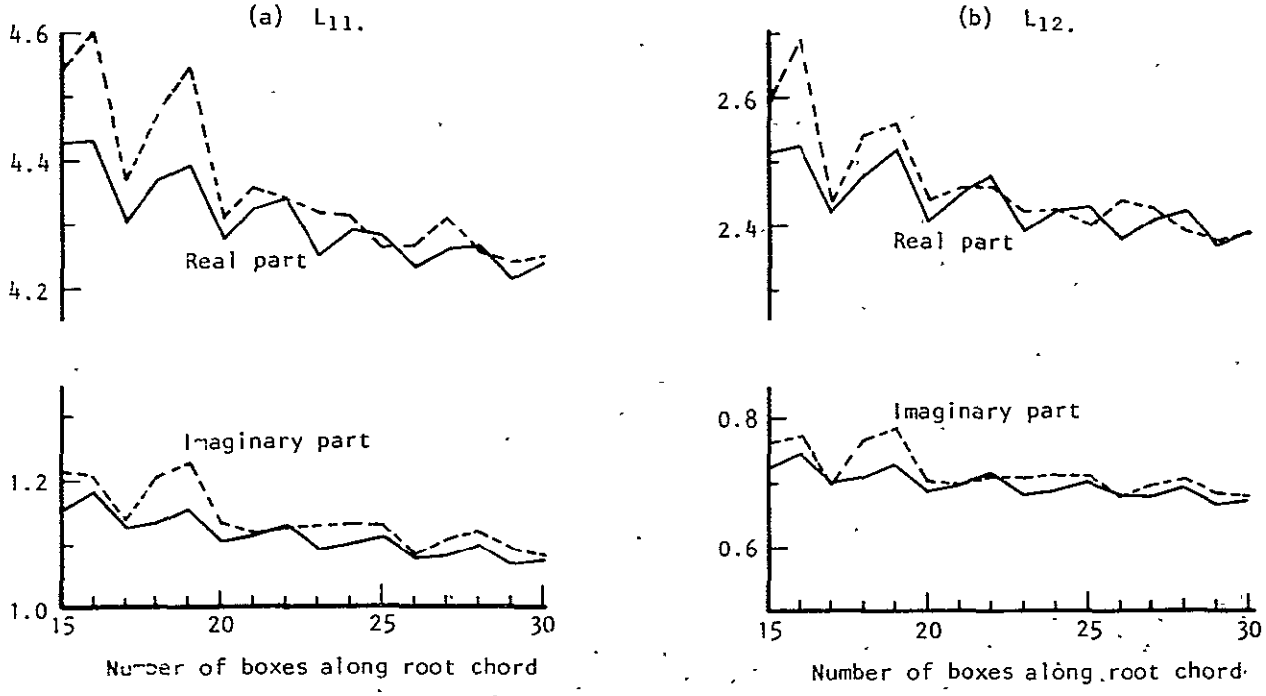


Figure 14. - Convergence of force coefficients due to plunge (mode 1) and pitch (mode 2) for cropped delta wing of aspect ratio 1.5 and taper ratio 0.143 at k = 0.095. Zero thickness.



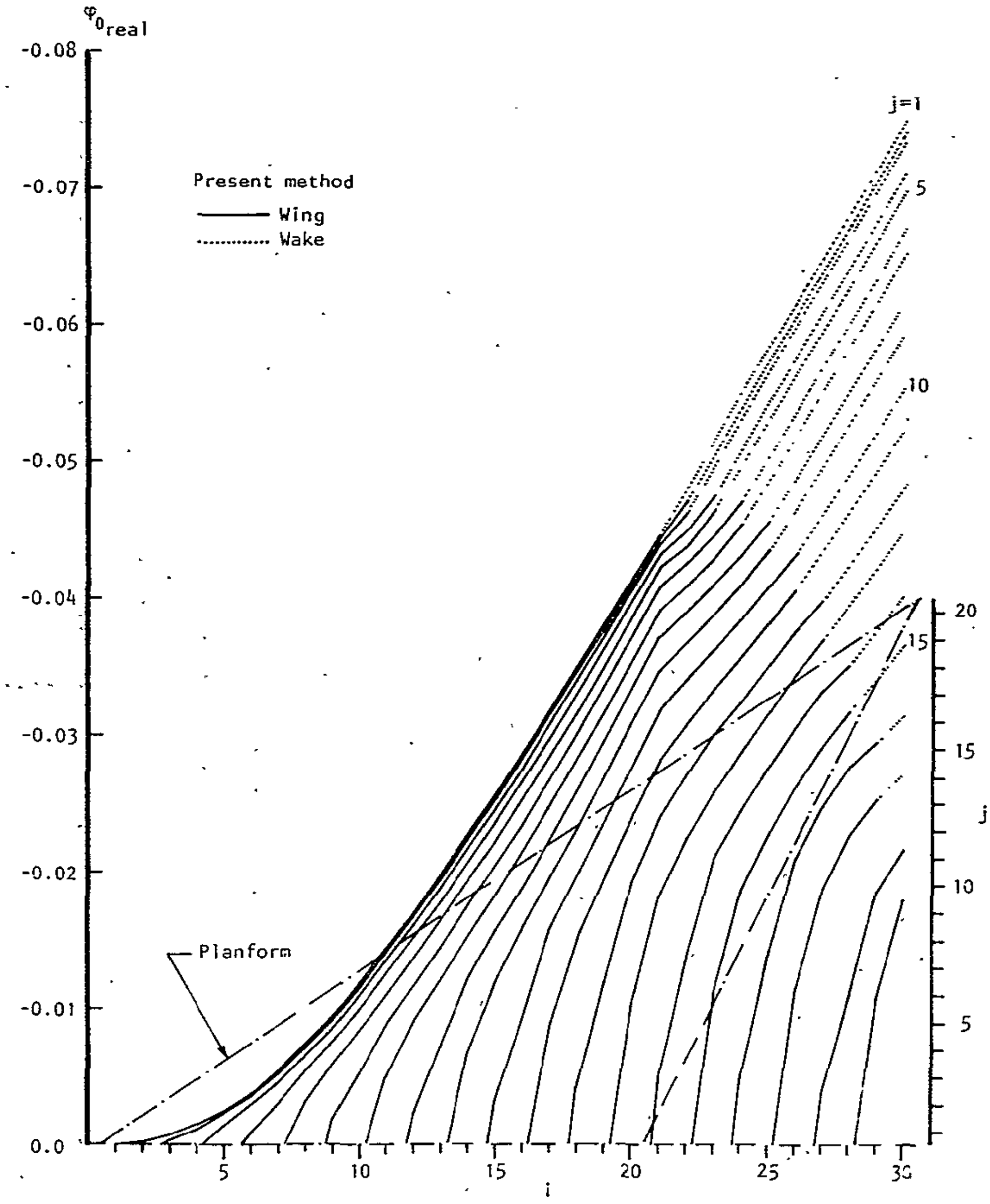
(a) L_{11}. (b) L_{12}.



(c) L_{21}. (d) L_{22}.

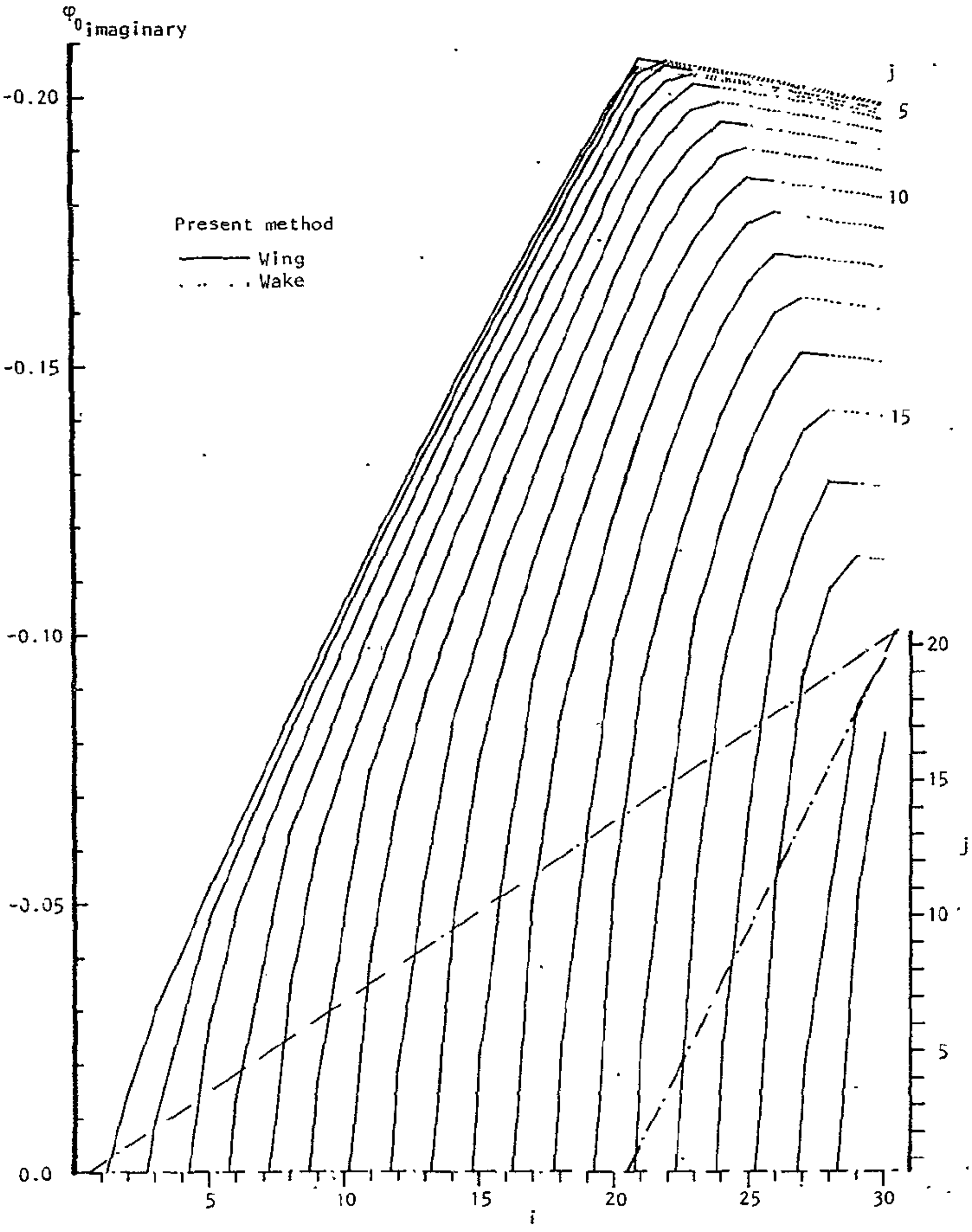
Figure 15. - Convergence of force coefficients due to plunge (mode 1) and pitch (mode 2) for arrow wing of aspect ratio 4.0 and leading edge sweep angle 56.33 degrees at $k = 0.5$. Zero thickness.

ORIGINAL PAGE IS
OF POOR QUALITY



(a) Real part.

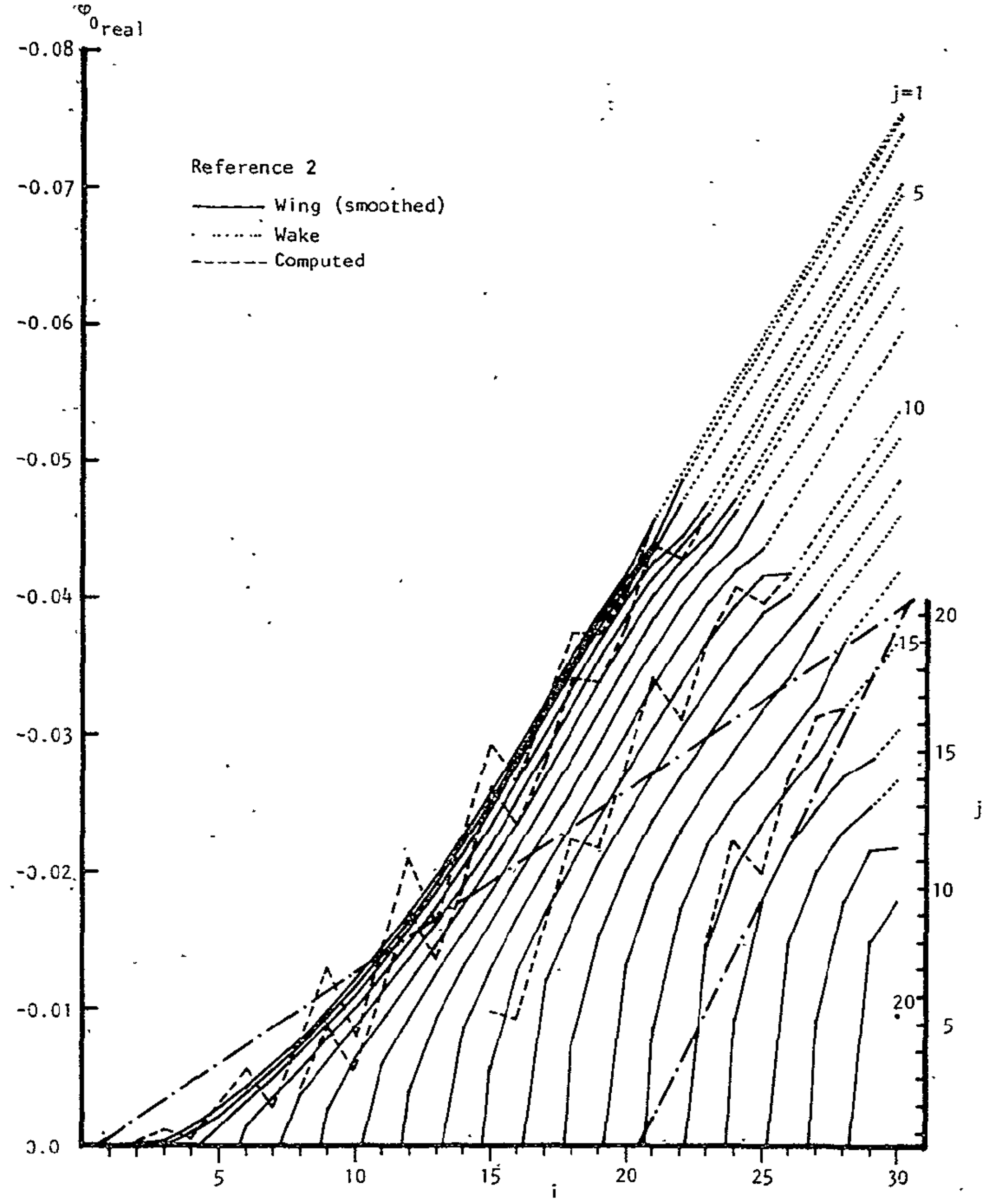
Figure 16. - Velocity potential along chordwise rows due to plunge for arrow wing of aspect ratio 4.0 and leading edge sweep angle 56.33 degrees at k = 0.5 -- present method. Zero thickness.



(b) Imaginary part.

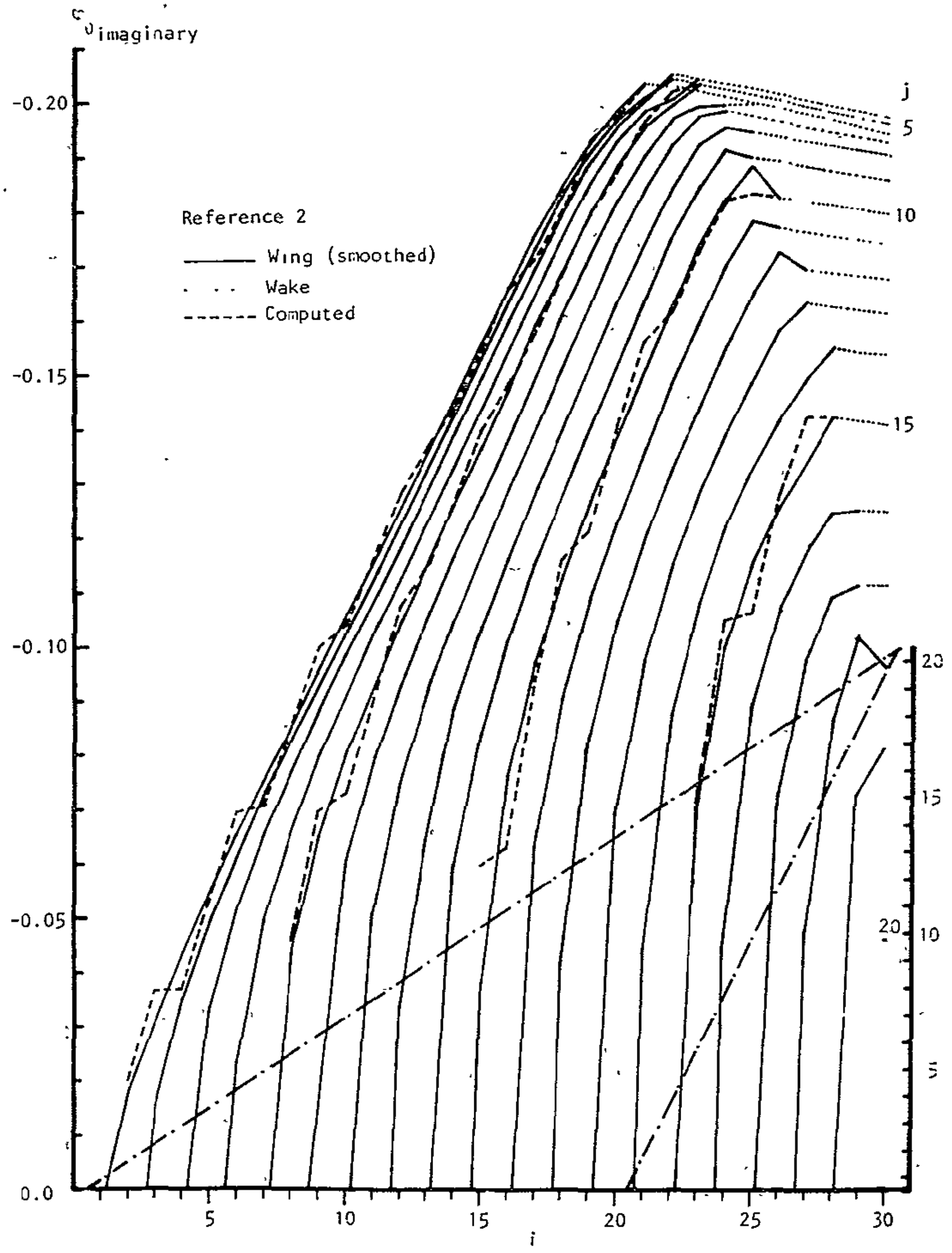
Figure 16. - Concluded.

ORIGINAL PAGE IS
OF POOR QUALITY



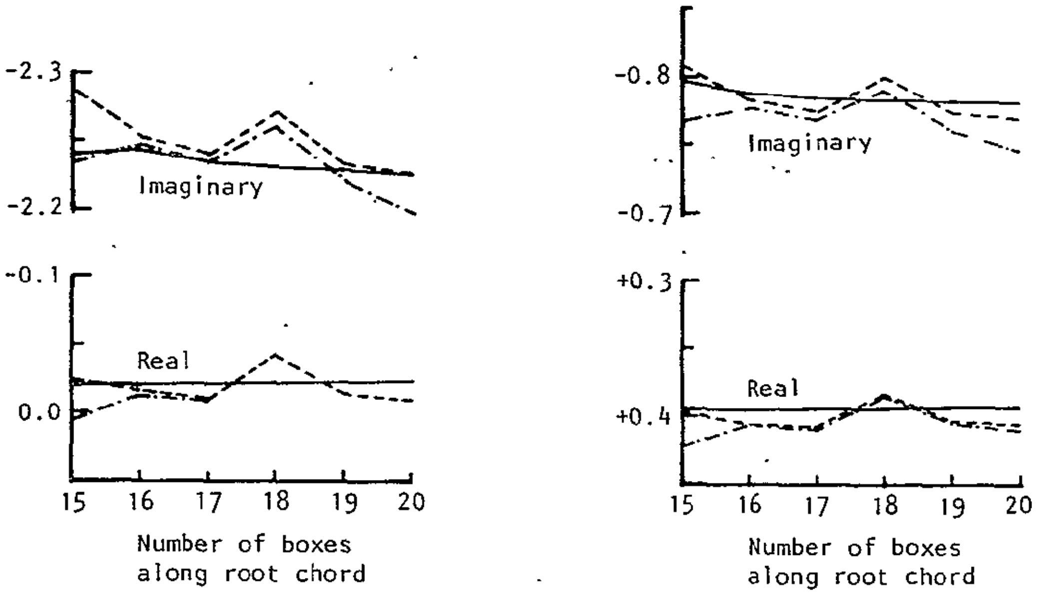
(a) Real part.

Figure 17. - Velocity potential along chordwise rows due to plunge for arrow wing of aspect ratio 4.0 and leading edge sweep angle 56.33 degrees at k = 0.5 -- reference 2. Zero thickness.



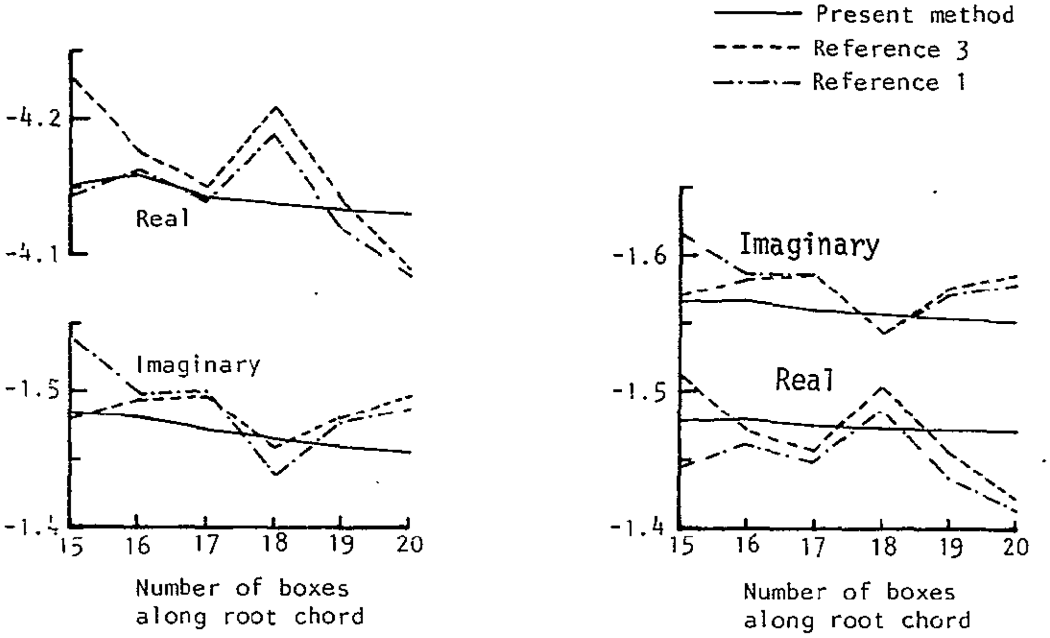
(b) Imaginary part.

Figure 17. - Concluded.



(a) L_{11}.

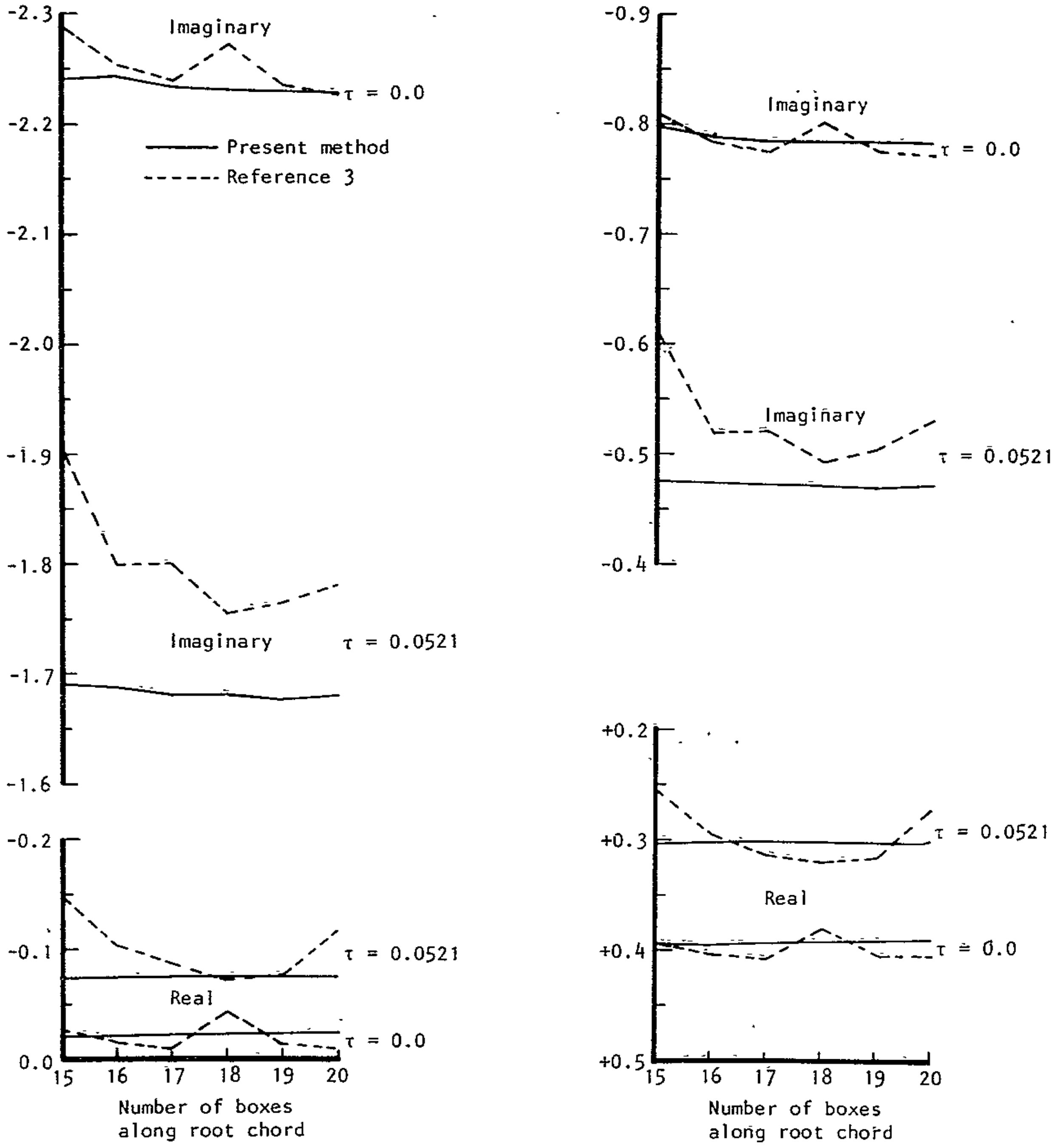
(b) L_{12}.



(c) L_{21}.

(d) L_{22}.

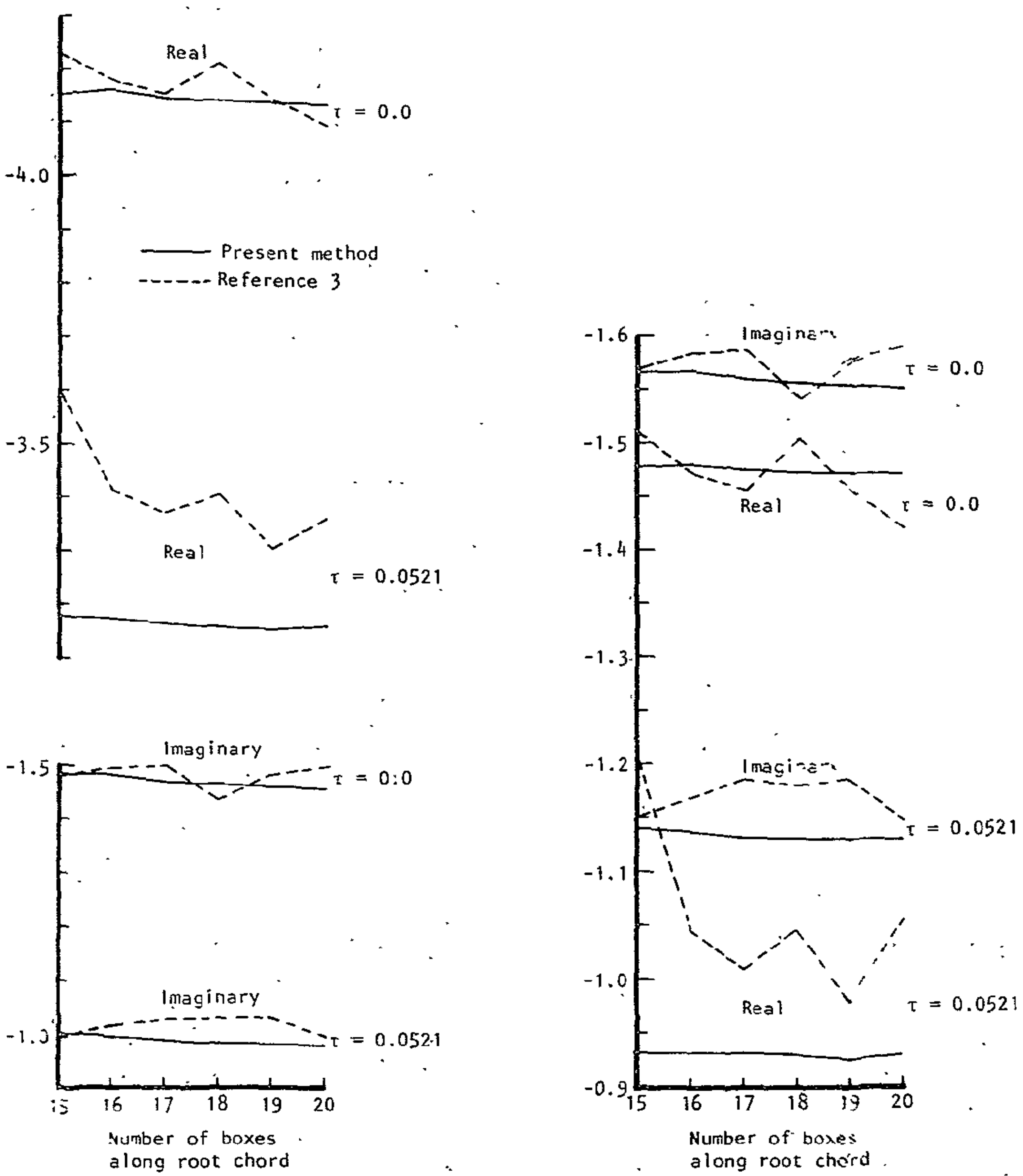
Figure 18. - Convergence of force coefficients due to plunge (mode 1) and pitch (mode 2) for rectangular wing of aspect ratio 2.0 at $k = 0.6$. Zero thickness.



(a) L_{11}.

(b) L_{12}.

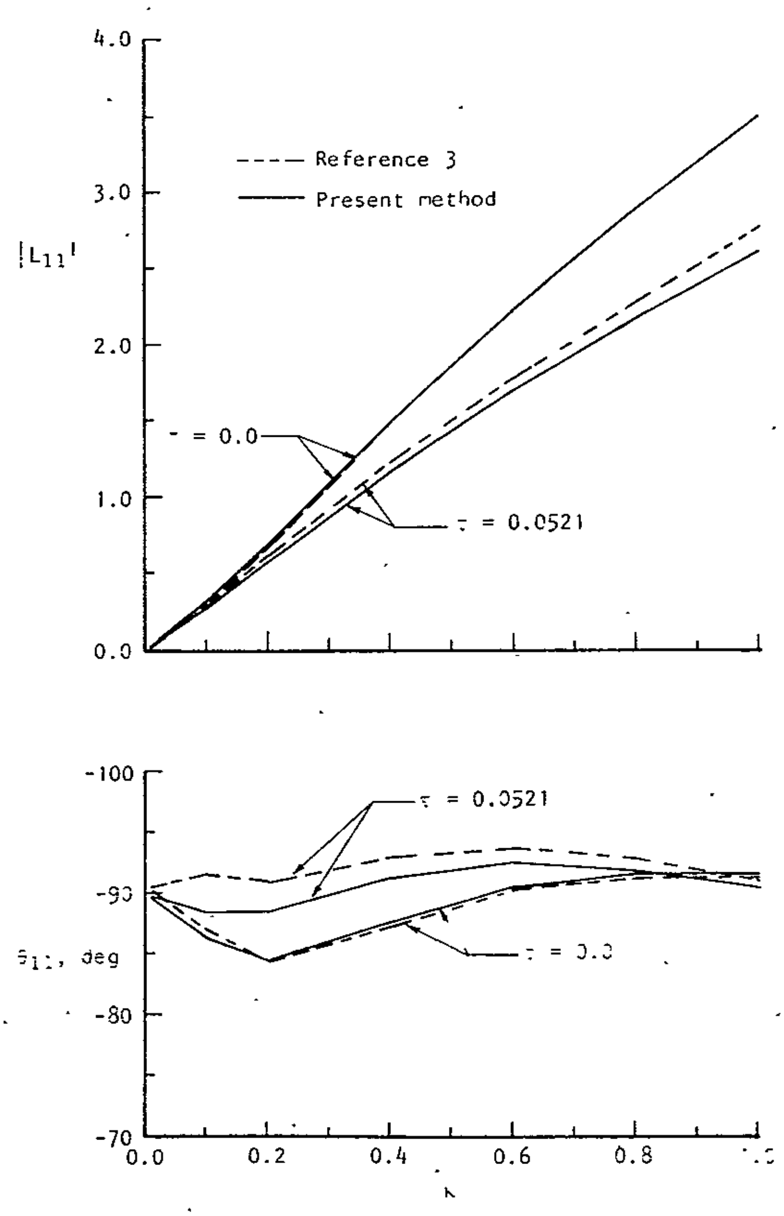
Figure 19. - Convergence of force coefficients due to plunge (mode 1) and pitch (mode 2) for rectangular wing of aspect ratio 2.0 at $k = 0.6$.



(c) L_{21}.

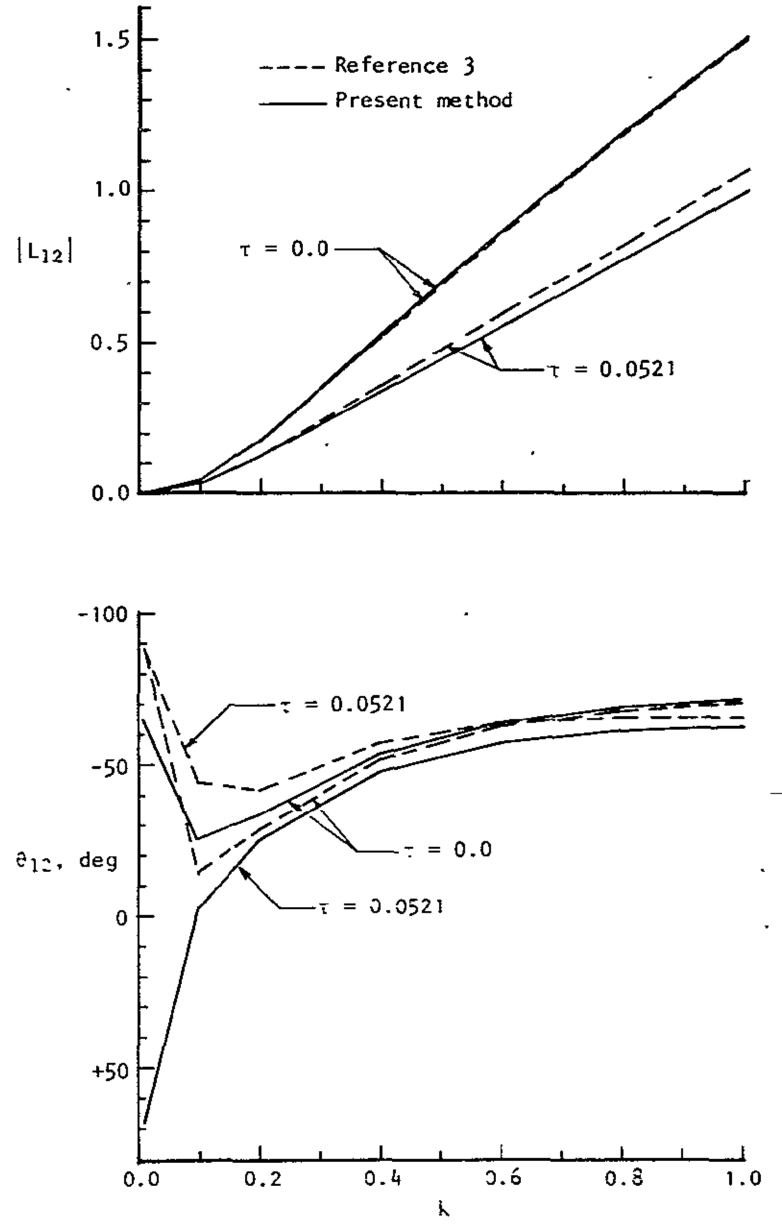
(d) L_{22}.

Figure 19. - Concluded.



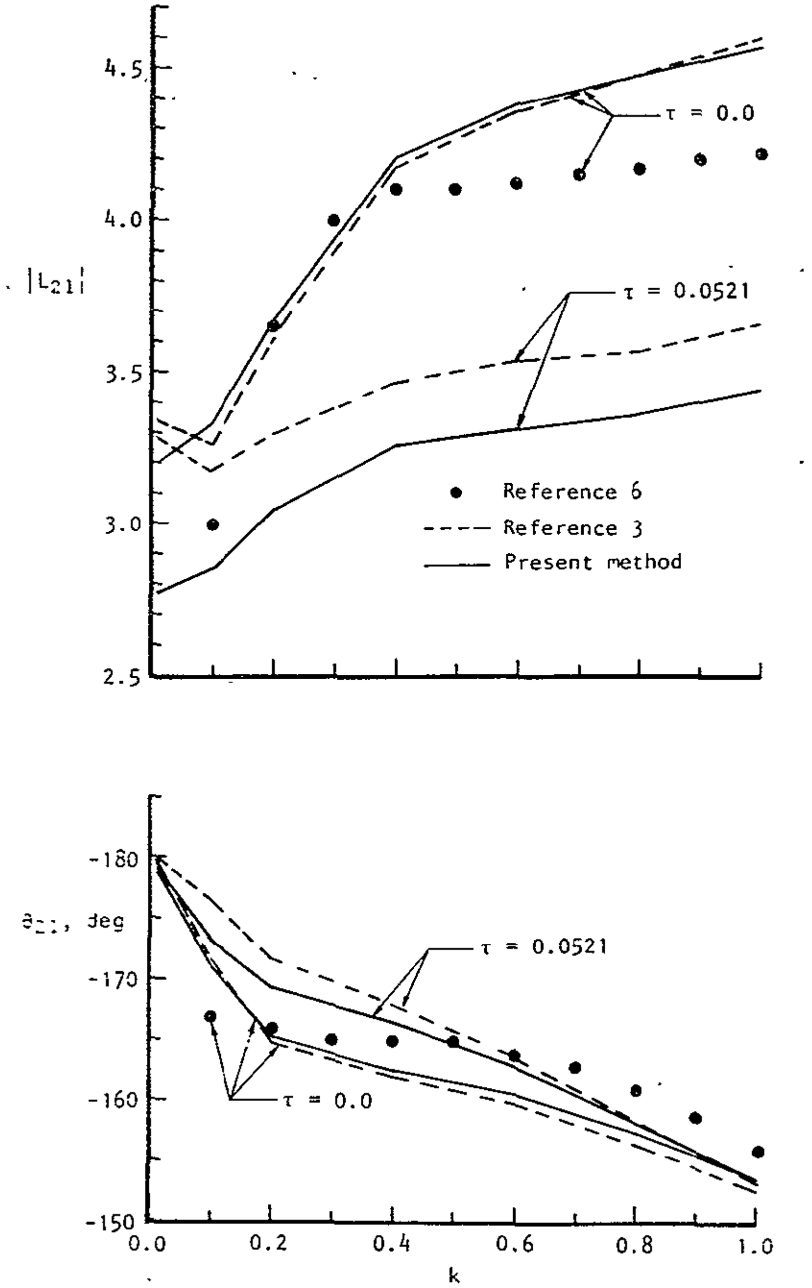
(a) Lift due to plunge.

Figure 20. - Force coefficients due to plunge (mode 1) and pitch (mode 2) for rectangular wing of aspect ratio 2.0 with 20 boxes along root chord.



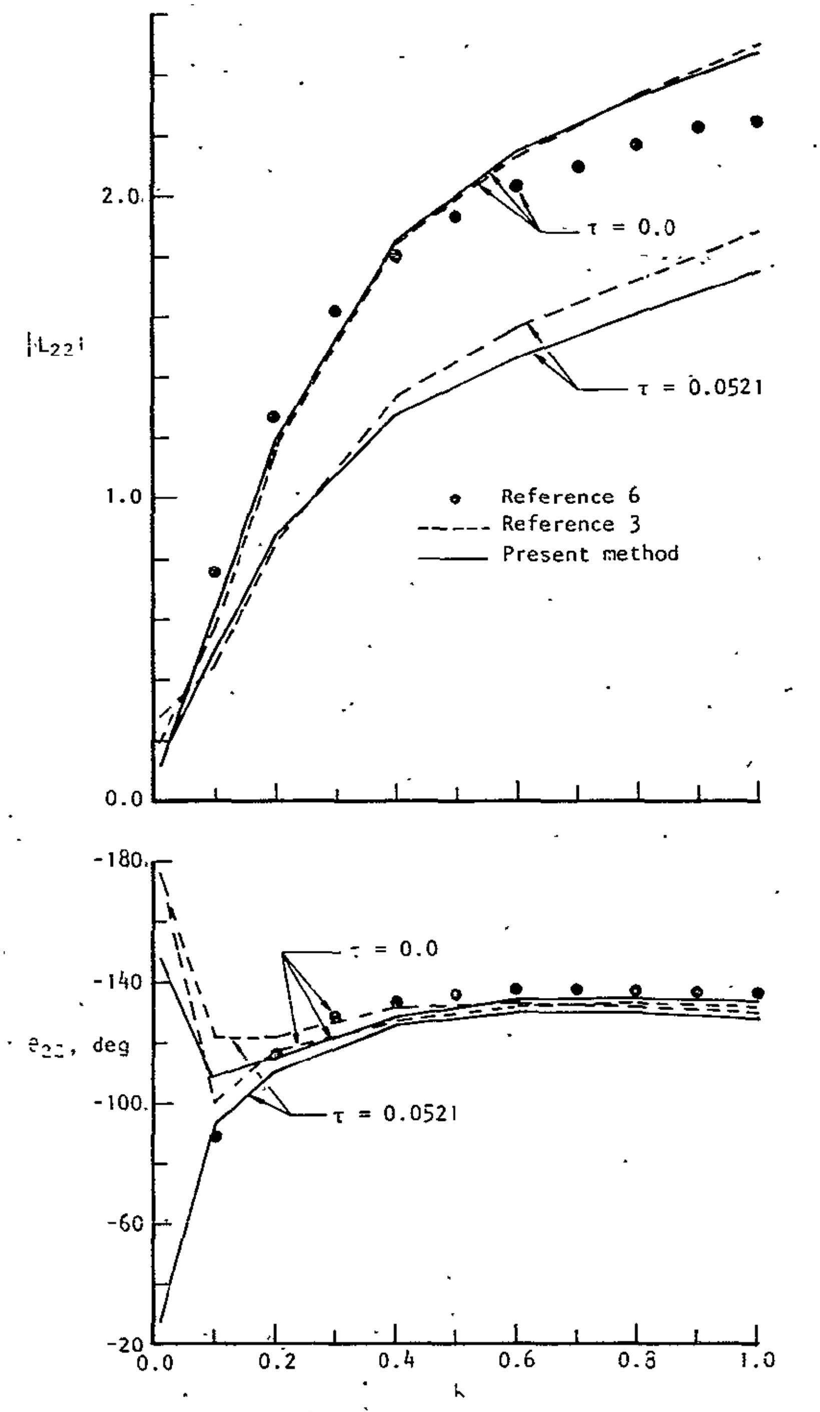
(b) Moment due to plunge.

Figure 20. - Continued.



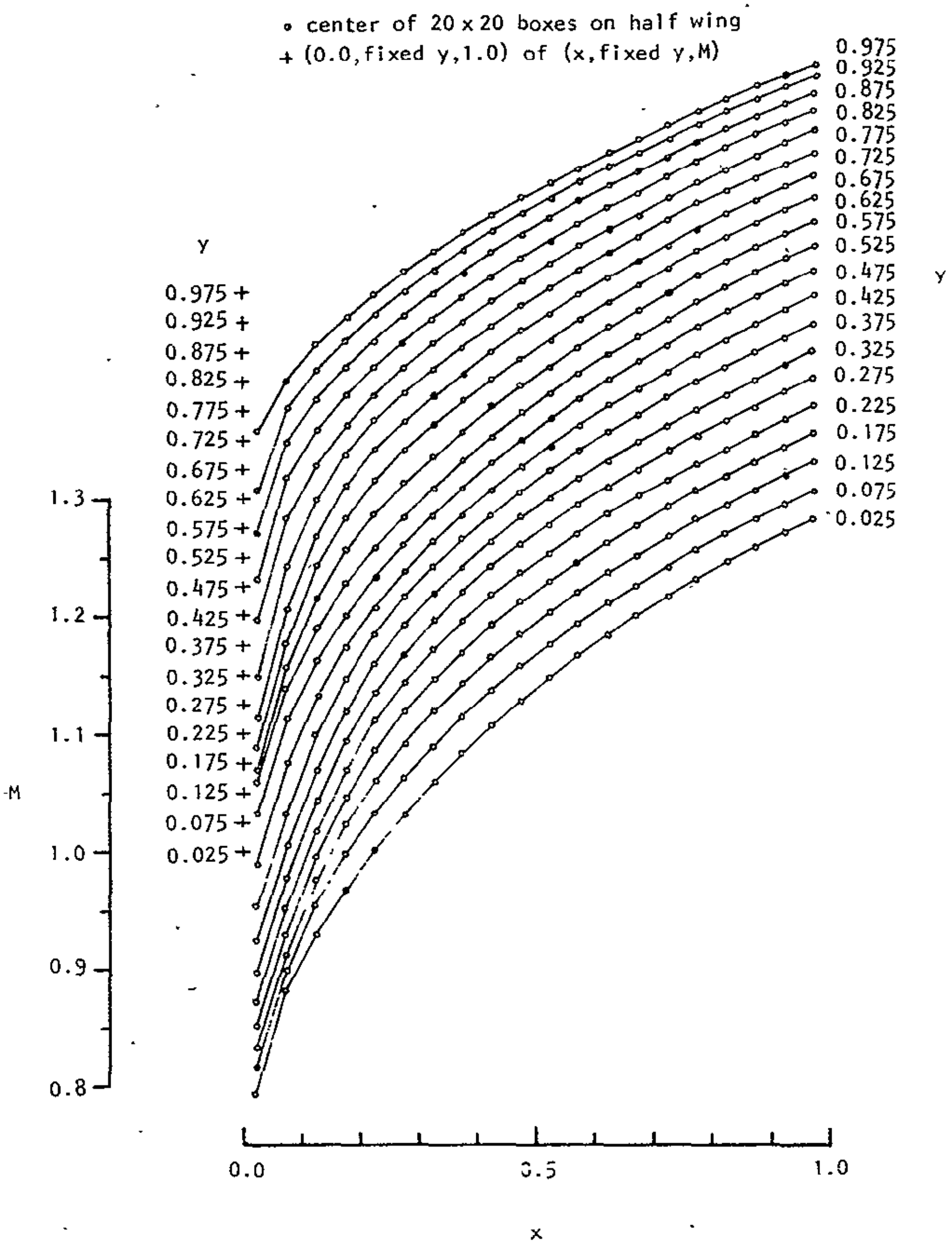
(c) Lift due to pitch.

Figure 20. - Continued.



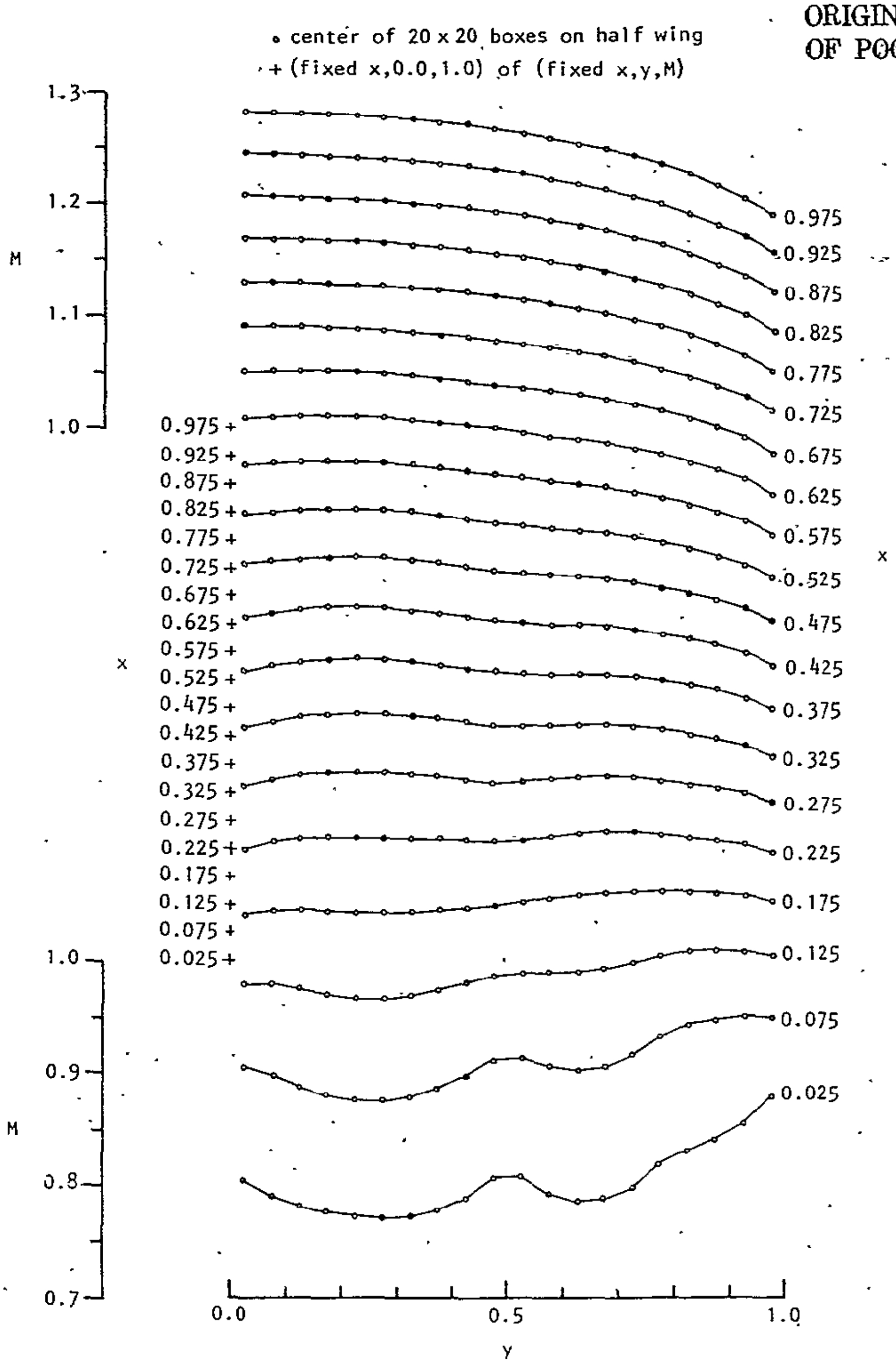
(d) Moment due to pitch.

Figure 20. - Concluded



(a) Chordwise distribution.

Figure 21. - Computed steady state (mean) Mach number on rectangular wing of aspect ratio 2.0 and thickness ratio 0.0521.



(b) Spanwise distribution.

Figure 21. - Concluded.

APPENDIX

COMPUTER PROGRAM

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 dimension 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 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 (n) 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_n, y_n), n_{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 conincides 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 one particular subroutine is needed.

1001: CBA: for wing deflection (DRED)
A=1, for NEW=1 }
B=1, for NEW=2 } spline-surface fitted results
C=1 - coordinates of the selected points used in spline-surface fitting
1002: BA: for wing upwash (WVAL)
A=1, for NEW=1 }
B=1, for NEW=2 } upwash
1003: FEDCBA for velocity potential (BOXP and BOXPO)
A=1, for NEW=1 }
B=1, for NEW=2 } velocity potential
C=1, for NEW=1 }
D=1, for NEW=2 } influence coefficient and solution matrices
E=1, for NEW=1 }
F=1, for NEW=2 } pressure coefficient
1004: A: for Mach number (MRED)
A=1 - spline-surface fitted results
1005: DCBA: for wing shape (SHAPE and PLNFM)
A=1, for NEW=1 } distributions of box, box area, leading and
B=1, for NEW=2 } trailing edges
C=1, - Mach number at box centers
D=1, for NEW=2 - redistributed leading and trailing edge segments

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 improvement in the method used for the functional form of data input. Those marked with ** are reserved for the case where a large number of data input points for deflection or Mach number is required.

Sample Case

A typical data deck set-up and output 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.

Input

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 3).
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         9.3         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
```

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        2.15112E-06        -2.42783E-02          2.42783E-02       -89.9949

   1     2       -1.59136E-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        -2.44696E-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       -1.31530E-05        -2.42779E-02          2.42779E-02       -90.0310

   1     2       -1.11814E-05        -1.61155E-02          1.61156E-02       -90.0398

         GENERALIZED FORCES (WITH THICKNESS EFFECT)

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

   2     1       -2.42785E+00        -2.29378E-02          2.42796E+00      -179.4587

   2     2       -1.61160E+00        -1.70333E-02          1.61169E+00      -179.3945
```

```
        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          3.63985E-04       -2.42141E-01         2.42141E-01         -89.9139

   1     2          1.20057E-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       -2.45150E-01         2.43792E+00        -174.2287

   2     2         -1.60948E+00       -1.81828E-01         1.61972E+00        -173.5544


             MODE NO.  1

             MODE NO.  2

        GENERALIZED FORCES (WITH THICKNESS EFFECT)

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

   1     1         -1.05402E-03       -2.41737E-01         2.41739E-01         -90.2498

   1     2         -8.92541E-04       -1.60300E-01         1.60302E-01         -90.3190

        GENERALIZED FORCES (WITH THICKNESS EFFECT)

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

   2     1         -2.42283E+00       -2.30551E-01         2.43377E+00        -174.5642

   2     2         -1.60745E+00       -1.71370E-01         1.61656E+00        -173.9147
```

```
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        2.60220E-02       -9.51380E-01        9.51736E-01         -88.4332

  1      2        1.93712E-02       -6.28442E-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       -9.98972E-01        2.61373E+00        -157.5298

  2      2       -1.60030E+00       -7.44075E-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        1.54347E-02       -9.39766E-01        9.39893E-01         -89.0591

  1      2        1.22288E-02       -6.20295E-01        6.20416E-01         -88.8706

GENERALIZED FORCES (WITH THICKNESS EFFECT)

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

  2      1       -2.39456E+00       -9.61025E-01        2.58021E+00        -158.1326

  2      2       -1.58606E+00       -7.17542E-01        1.74082E+00        -155.6578
```

Computer Program Listing

```
      PROGRAM SBOXR(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)                 SBOXR
C     MAIN PROGRAM                                                         SBOXR
C     MB=MAXIMUM NUMBER OF BOXES ALONG THE CENTERLINE CHORD ALLOWED        SBOXR
C     .   IN DIMENSION STATEMENT                                           SBOXR
C     MD=MAXIMUM NUMBER OF MODES ALLOWED IN DIMENSION STATEMENT            SBOXR
C     NB=MAXIMUM NUMBER OF BOXES ALLOWED FOR SPLINE FITTING                SBOXR
C     NM=NB+3 ALLOWED IN DIMENSION STATEMENT                               SBOXR
      COMMON    XX(30),YY(30),XEDGI(8),YEDGI(8),XEDG(32),YEDG(32)          SBOXR
     1          ,NS(2),NST(2) ,XTDGI(3),YTDGI(3),XTDG(12),YTDG(12)         SBOXR
     2          ,XLE(2,30),XTE(2,30),YMAX(2),EDG(2,30),AR(30,30,2)         SBOXR
     3          ,SFMX(100),SFMY(100),SFMH(103),KSFM,T(103,104),DA(1005)    SBOXR
     4          ,ML(2,30),MLH(2,30),MLT(2,2,30),MLC(2,2,30)                SBOXR
     5          ,JMAX(2),JMAK(2),CTE(30),AMA(30,30,2)                      SBOXR
     6          ,AREA,CK,D,DH,DI,IEDG,IW,L,MB,NEW,NM,NSMAX,NEDGI,NTDGI     SBOXR
      DIMENSION A(2, 60,30),B(2,30,30)                                     SBOXR
     1          ,SFDX(100,3),SFDY(100,3,2),SFDH(103,3,2),KSFD(3),G(3)      SBOXR
     2          ,XEDGT(8),YEDGT(8),XTDGT(3),YTDGT(3)                       SBOXR
      DATA Z/1H /                                                          SBOXR
      IR=5                                                                 SBOXR
      IW=6                                                                 SBOXR
      NEDGI=8                                                              SBOXR
      NTDGI=3                                                              SBOXR
      MB=30                                                                SBOXR
      MD=3                                                                 SBOXR
      NSMAX=32                                                             SBOXR
      NB=100                                                               SBOXR
C     MATCH ABOVE NUMBERS TO THE DEFINED DIMENSION IN STORAGE ARRAYS       SBOXR
      ME=MB+1                                                              SBOXR
      MC=2*MB                                                              SBOXR
      NM=NB+3                                                              SBOXR
      WRITE(IW,55)                                                         SBOXR
C     READ DATA FOR THE ACTUAL WING                                        SBOXR
  100 CALL [illegible](DA)                                                 SBOXR
      NEW=1                                                                SBOXR
      TEST1=DA(1001)+DA(1002)+DA(1003)+DA(1004)+DA(1005)                   SBOXR
      CK=DA(23)*DA(24)/DA(25)*6.28318531                                   SBOXR
      DI=DA(27)                                                            SBOXR
      L=DI                                                                 SBOXR
      D=1.0/DI                                                             SBOXR
      DH=0.5*D                                                             SBOXR
      WRITE(IW,60) (DA(I),I=1,7)                                           SBOXR
  110 IF (L) 600,600,120                                                   SBOXR
  120 IF (MB-L) 600,130,130                                                SBOXR
  130 WRITE(IW,65) L,DA(24),CK,DA(25),DA(23)                               SBOXR
      IF (DA(26)) 160,150,160                                              SBOXR
  150 CALL SHAPE                                                           SBOXR
      IF(NEW.EQ.2) GO TO 180                                               SBOXR
      AC=6.0/AREA                                                          SBOXR
  160 LIM=ML(NEW,L)                                                        SBOXR
      IF (LIM-MB) 170,170,650                                              SBOXR
  170 LIM2=2*MB                                                            SBOXR
      NFLNS = DA(49)                                                       SBOXR
      IF(NFLNS.EQ.0) NFLNS = L                                             SBOXR
      LPOT = MINO(L,NFLNS)                                                 SBOXR
      LIM1=2*MB                                                            SBOXR
      CALL        [illegible]POT2(LIM1,LIM2,LPOT,CK,D,A)                   SBOXR
  180 CONTINUE                                                             SBOXR
      M=0                                                                  SBOXR
      K=DA(28)                                                             SBOXR
      GO TO 230                                                            SBOXR
```

```
C                                                                       SBOXR     62
C     PRELIMINARY CALCULATIONS ARE FINISHED.                            SBOXR     63
C     THE NEXT SECTION IS GONE THROUGH FOR EACH MODE.                   SBOXR     64
C                                                                       SBOXR     65
  200 IF (DA(26)) 230,210,230                                           SBOXR     66
  210 IF(NEW.EQ.2) GO TO 230                                            SBOXR     67
      CALL DATRD(DA)                                                    SBOXR     68
  230 K=K-1                                                             SBOXR     69
      M=M+1                                                             SBOXR     70
      IF(TEST1.LT.1.0) GO TO 250                                        SBOXR     71
      WRITE(IW,55)                                                      SBOXR     72
  250 WRITE(IW,15) M                                                    SBOXR     73
      WRITE(IW,16) (DA(I),I=13,19)                                      SBOXR     74
   15 FORMAT(1H0,15X,8HMODE NO.,I3)                                     SBOXR     75
   16 FORMAT(1H+,30X,7A10)                                              SBOXR     76
      IF (DA(26)) 290,280,290                                           SBOXR     77
C                                                                       SBOXR     78
  280 IPRINT=DA(1001)                                                   SBOXR     79
      CALL        DRED(SFDX,SFDY,SFDH,KSFD,SFMX,SFMY,SFMH,KSFM,DA,T     SBOXR     80
     $                 ,XX,YY,IW,L,MLC,M,MD,NB,NEW,NM,KSFS,IPRINT)      SBOXR     81
C                                                                       SBOXR     82
      G(M)=DA(51)                                                       SBOXR     83
  290 IF(K) 310,310,300                                                 SBOXR     84
  300 IF (M-MD) 200,310,310                                             SBOXR     85
  310 CONTINUE                                                          SBOXR     86
C                                                                       SBOXR     87
      IF(IFIX(DA(50)).EQ.2.AND.NEW.EQ.1) GO TO 490                      SBOXR     88
      DO 480 M1=1,M                                                     SBOXR     89
      IF (G(M1))  480,320,480                                           SBOXR     90
  320 CONTINUE                                                          SBOXR     91
C                                                                       SBOXR     92
      IPRINT=DA(1002)                                                   SBOXR     93
      CALL        WVAL(XX,YY,XTDG,YTDG,XTE,SFDX,SFDY,SFDH,KSFD,S,D,CK   SBOXR     94
     $                 ,IW,L,ML,M1,NEW,NST,MB,MD,NB,NM,KSFS,IPRINT)     SBOXR     95
C                                                                       SBOXR     96
      IF(YEDGI(1)) 590,360,330                                          SBOXR     97
  330 LIM2=2*ML(NEW,1)                                                  SBOXR     98
C     LEADING EDGE CORRECTION                                           SBOXR     99
      DO 350 J=1,LIM2                                                   SBOXR    100
  350 S(J,1,1)=S(J,1,1)*2.0/3.14159265                                  SBOXR    101
  360 CONTINUE                                                          SBOXR    102
C                                                                       SBOXR    103
      YE=YEDGI(1)                                                       SBOXR    104
      DAN    =DA(49)                                                    SBOXR    105
      IPRINT=DA(1003)                                                   SBOXR    106
      CALL        BOXP(XX,YY,XTDG,YTDG,XTE,XLE,A,AR,DAN,T,EDG,S,CK,D,YE SBOXR    107
     1                 ,IPRINT,MLT,MLC,AMA                              SBOXR    108
     2                 ,JMAX,L,ML,MLM,M1,NEW,NST,IW,MB,MC,ME,MD)        SBOXR    109
C                                                                       SBOXR    110
  380 CONTINUE                                                          SBOXR    111
C                                                                       SBOXR    112
C     COMPUTATION OF GENERALIZED FORCES FOR EACH MODE                   SBOXR    113
C                                                                       SBOXR    114
      IF(TEST1.LT.1.0) GO TO 410                                        SBOXR    115
      WRITE(IW,55)                                                      SBOXR    116
      WRITE(IW,16) (DA(I),I=1,7)                                        SBOXR    117
      WRITE(IW,65) L,DA(24),CK,DA(25),DA(23)                            SBOXR    118
  410 WRITE (IW,20)                                                     SBOXR    119
      IF(NEW.EQ.1) WRITE(IW,22)                                         SBOXR    120
      IF(NEW.EQ.2) WRITE(IW,24)                                         SBOXR    121
      WRITE(IW,25)                                                      SBOXR    122
   20 FORMAT(/1H010X,18HGENERALIZED FORCES)                             SBOXR    123
   22 FORMAT(1H+,28X,22H (NO THICKNESS EFFECT))                         SBOXR    124
   24 FORMAT(1H+,28X,24H (WITH THICKNESS EFFECT))                       SBOXR    125
   25 FORMAT(1H0,5X,5HMODES/4X,11HPRES. DEFL.,8X,9HREAL PART,10X,9HIMAG SBOXR    126
     $PART,10X,10HABS. VALUE,6X,11HPHASE ANGLE)                         SBOXR    127
C                                                                       SBOXR    128
      DO 470 M2=1,M                                                     SBOXR    129
C                                                                       SBOXR    130
      CALL        FORCI(XX,YY,S,SFDX,SFDY,SFDH,KSFD,KSFS,XLE,XTE,CTE,YMAX SBOXR  131
     1                 ,AMA,JMAX(NEW),MLT,L,NEW,M2,MB,MD,NB,NM,CK,T)    SBOXR    132
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C                                                                       SBOXR   133
      S1=AC*T(1,1)                                                      SBOXR   134
      S2=AC*T(2,1)                                                      SBOXR   135
      S3= SQRT(S1**2+S2**2)                                             SBOXR   136
      S4=57.29578*ATAN2(S2,S1)                                          SBOXR   137
  450 WRITE(IW,30) M1,M2,S1,S2,S3,S4                                    SBOXR   138
   30 FORMAT (1H0216,1P3E19.5,0P1F16.4)                                 SBOXR   139
  470 CONTINUE                                                          SBOXR   140
  480 WRITE  (IW,55)                                                    SBOXR   141
  490 CONTINUE                                                          SBOXR   142
      IF(DA(26).GT.C.) GO TO 510                                        SBOXR   143
      DO 500  I=13,22                                                   SBOXR   144
  500 DA(I)=Z                                                           SBOXR   145
C                                                                       SBOXR   146
C     CHECK IF TRANSFORMED WING HAS BEEN CALCULATED                     SBOXR   147
C     IF NOT, GO TO  SHAPE  TO TRANSFORM THE WING                       SBOXR   148
C                                                                       SBOXR   149
  510 IF(FIX(DA(50))-2) 515,[illegible]                                 SBOXR   150
  515 IF(NEW-1) 520,[illegible]                                         SBOXR   151
  520 IF(DA(26).LE.0.0) GO TO 530                                       SBOXR   152
      DO 525 I=1,NEDGE                                                  SBOXR   153
      XEDG(I)=XEDGT(I)                                                  SBOXR   154
      YEDG(I)=YEDGT(I)                                                  SBOXR   155
      IF(I.GT.NTDG) GO TO 525                                           SBOXR   156
      XTDG(I)=XTDGT(I)                                                  SBOXR   157
      YTDG(I)=YTDGT(I)                                                  SBOXR   158
  525 CONTINUE                                                          SBOXR   159
      GO TO 570                                                         SBOXR   160
  530 IF(DA(26)) 535,535,545                                            SBOXR   161
  535 DO 540 I=1,NEDGE                                                  SBOXR   162
      [illegible]                                                       SBOXR   163
      [illegible]                                                       SBOXR   164
      IF(I.GT.NTDG) GO TO 540                                           SBOXR   165
      [illegible]                                                       SBOXR   166
      [illegible]                                                       SBOXR   167
  540 CONTINUE                                                          SBOXR   168
  545 DO 550 I=1,NEDGE                                                  SBOXR   169
      XEDG(I)=XEDG2(I)                                                  SBOXR   170
      YEDG(I)=YEDG2(I)                                                  SBOXR   171
      IF(I.GT.NTDG) GO TO 550                                           SBOXR   172
      XTDG(I)=XTDG!(I)                                                  SBOXR   173
      YTDG(I)=YTDGI(I)                                                  SBOXR   174
  550 CONTINUE                                                          SBOXR   175
  570 CONTINUE                                                          SBOXR   176
      IF(NEW.EQ.2) GO TO 100                                            SBOXR   177
      NEW=2                                                             SBOXR   178
      IF(DA(26)) 180,580,190                                            SBOXR   179
  580 IPRINT=DA(1004)                                                   SBOXR   180
      CALL        MKED(DA,T,NM,NS,KSFM,SFMX,SFMY,SFMH,IW,IPRINT)        SBOXR   181
      GO TO 150                                                         SBOXR   182
C                                                                       SBOXR   183
C     ERROR EXITS                                                       SBOXR   184
C                                                                       SBOXR   185
  590 IPR=[illegible]                                                   SBOXR   186
      GO TO 610                                                         SBOXR   187
  600 IPR=27                                                            SBOXR   188
  610 WRITE (IW,35) IPR                                                 SBOXR   189
   35 FORMAT(1H0,10X,8HBAD DATA,I4)                                     SBOXR   190
      GO TO 700                                                         SBOXR   191
  650 WRITE    (IW,40)                                                  SBOXR   192
   40 FORMAT(1H0,10X,[illegible] LIMIT ON NUMBER OF BOXES EXCEEDED.)    SBOXR   193
      [illegible]                                                       SBOXR   194
   55 FORMAT(1H1)                                                       SBOXR   195
   50 FORMAT(1H0,10X,7A10)                                              SBOXR   196
    5 FORMAT([illegible] BOXES ALONG ROOT CHORD,                        SBOXR   197
     [illegible]                LENGTH     =,F6.2,3H FT/                SBOXR   198
     [illegible]             FREQUENCY =,F6.3,                          SBOXR   199
     [illegible]            STREAM VELOCITY =,F6.2,7H FT/SEC/           SBOXR   200
     [illegible]       FREQUENCY =,[illegible]11.3,10H CYCLE/SEC)       SBOXR   201
                                                                        SBOXR   202
```

```
      SUBROUTINE DATRD(DATA)                                            DATRD   2
C                    CARD-READ SUBROUTINE "DATRD(DATA(I))"              DATRD   3
      DIMENSION DRBU(12),DATA(1),DDRBU(10)                              DATRD   4
      DATA ATEST/5HALPHA/,DTEST/1H /,ETEST/1H-/,STEST/1H+/              DATRD   5
      DATA Z/1H /                                                       DATRD   6
      IR=5                                                              DATRD   7
      IW=6                                                              DATRD   8
    1 READ   (IR,2) EMIN,ALP,IND,(DRBU(I),I=1,7)                        DATRD   9
    2 FORMAT(A1,A5,I6,6A10,A8)                                          DATRD  10
      IF(IND.EQ.0) GO TO 20                                             DATRD  11
      IF (EMIN.NE.STEST) GO TO 105                                      DATRD  12
C     NEW WING IF COLUMN 1 CONTAINS A PLUS SIGN                         DATRD  13
C     INITIALIZATION OF DATA ARRAY                                      DATRD  14
      DO 101 I=23,1005                                                  DATRD  15
  101 DATA(I)=0.0                                                       DATRD  16
      DO 102 I=1,22                                                     DATRD  17
  102 DATA(I)=Z                                                         DATRD  18
      DO 103 I=104,700,4                                                DATRD  19
  103 DATA(I)=1.0                                                       DATRD  20
  105 CONTINUE                                                          DATRD  21
      IF (ALP.EQ.ATEST) GO TO 9                                         DATRD  22
      IF (ALP.NE.DTEST) GO TO 8                                         DATRD  23
C     NUMERIC CARD                                                      DATRD  24
      DO 3 I=1,6                                                        DATRD  25
      DDRBU(I)=DRBU(I)                                                  DATRD  26
    3 CONTINUE                                                          DATRD  27
      DECODE(60,990,DDRBU)(DRBU(I),I=1,5)                               DATRD  28
      DO 5 I=1,5                                                        DATRD  29
      IF(DRBU(I))4,6,4                                                  DATRD  30
C     TEST FOR BLANK FIELD                                              DATRD  31
    6 IF(SIGN (1.0,DRBU(I)))5,5,4                                       DATRD  32
    4 DATA(IND)=DRBU(I)                                                 DATRD  33
    5 IND=IND+1                                                         DATRD  34
      GOTO 11                                                           DATRD  35
C     ALPHA CARD                                                        DATRD  36
    9 DO 10 I=1,7                                                       DATRD  37
      DATA(IND)=DRBU(I)                                                 DATRD  38
   10 IND=IND + 1                                                       DATRD  39
   11 IF (EMIN.NE.ETEST) GO TO 1                                        DATRD  40
C     RETURN IF COLUMN 1 CONTAINS A MINUS SIGN                          DATRD  41
   13 RETURN                                                            DATRD  42
C     END OF DATA CARDS                                                 DATRD  43
   20 WRIT (IW,995)                                                     DATRD  44
      CALL EXIT                                                         DATRD  45
      STOP                                                              DATRD  46
C     BAD CARD                                                          DATRD  47
    8 CONTINUE                                                          DATRD  48
      WRITE                                                             DATRD  49
     1 (IW,993) EMIN,ALP,IND,(DRBU(I),I=1,7)                            DATRD  50
      WRITE    (IW,991)                                                 DATRD  51
      STOP                                                              DATRD  52
  990 FORMAT(5E12.0)                                                    DATRD  53
  991 FORMAT(38H BAD DATA ON THIS CARD. JOB TERMINATED  )               DATRD  54
  993 FORMAT(12H0CARD IMAGE=A1,A5,I6,7A10)                              DATRD  55
  995 FORMAT(///1H ,10X,17HNO MORE DATA CARD/)                          DATRD  56
      END                                                               DATRD  57
```

```
      [illegible]                                                              SHAPE
      SUBROUTINE SHAPE                                                         SHAPE
      COMMON [illegible]                                                       SHAPE
     1 [illegible] XTDG(12),YTDG(12)                                           SHAPE
     2 [illegible] AK(30,30)                                                   SHAPE
     3 [illegible] DA(1005)                                                    SHAPE
     4 [illegible] HLC(2,2,30)                                                 SHAPE
     5 [illegible] CT(30),AHA(30,30,2)                                         SHAPE
     6 [illegible] NB,NSMAX,NEDGI,NTDGI                                        SHAPE
C                                                                              SHAPE
      ERR=1.E-05                                                               SHAPE
      IPRIN=DA(1005)                                                           SHAPE
      NRITE=IPRIN/1000                                                         SHAPE
      IPRIN=IPRIN-1000*NRITE                                                   SHAPE
      JRITE=IPRIN/100                                                          SHAPE
      IPRIN=IPRIN-100*JRITE                                                    SHAPE
      IRITE=IPRIN/10                                                           SHAPE
      IF(NEW.EQ.1) IRITE=IPRIN-10*IRITE                                        SHAPE
      IF(NEW.EQ.2) GO TO 200                                                   SHAPE
      IEDG=0                                                                   SHAPE
      NS(1)=DA(29)                                                             SHAPE
      IF (NS(1)) 850,850,100                                                   SHAPE
  100 IF(NS(1)-NEDGI+1) 105,105,850                                            SHAPE
  105 NSP=NS(1)+1                                                              SHAPE
      NS(1)=NSP                                                                SHAPE
      IF (DA(24)) 855,855,110                                                  SHAPE
  110 DO 115 I=1,NSP                                                           SHAPE
      XEDG(I)=DA(2*I+27)/DA(24)                                                SHAPE
  115 YEDG(I)=DA(2*I+28)/DA(24)                                                SHAPE
      XEDG(1)=0.0                                                              SHAPE
      NST(1)=DA(45)                                                            SHAPE
      IF(NST(1)) 871,125,130                                                   SHAPE
  125 XTDG(1)=1.0                                                              SHAPE
      NST(1)=1                                                                 SHAPE
      GO TO 141                                                                SHAPE
  130 IF(NST(1)-NTDGI+1) 135,135,871                                           SHAPE
  135 NSTP=NST(1)                                                              SHAPE
      DO 140 I=1,NSTP                                                          SHAPE
      YTDG(I)=DA(2*I+43)/DA(24)                                                SHAPE
  140 XTDG(I)=DA(2*I+44)/DA(24)                                                SHAPE
  141 NSTP=NST(1)+1                                                            SHAPE
      NST(1)=NSTP                                                              SHAPE
      XTDG(NSTP)=XEDG(NSP)                                                     SHAPE
      YTDG(NSTP)=YEDG(NSP)                                                     SHAPE
      YTDG(1)=0.0                                                              SHAPE
      IF(NST(1).EQ.NTDGI) GO TO 150                                            SHAPE
      JL=NST(1)+1                                                              SHAPE
      DO 142 I=JL,NTDGI                                                        SHAPE
      XTDG(I)=XTDG(I-1)                                                        SHAPE
  142 YTDG(I)=YTDG(I-1)                                                        SHAPE
  150 IF(ABS(XTDG(NSTP)-XEDG(NSP))+YTDG(NSTP)-YEDG(NSP)).GT.ERR)               SHAPE
     $   GO TO 871                                                             SHAPE
      DO 160 I=1,NEDGI                                                         SHAPE
      XEDGI(I)=XEDG(I)                                                         SHAPE
      YEDGI(I)=YEDG(I)                                                         SHAPE
      IF(I.GT.NTDGI) GO TO 160                                                 SHAPE
      XTDGI(I)=XTDG(I)                                                         SHAPE
      YTDGI(I)=YTDG(I)                                                         SHAPE
  160 CONTINUE                                                                 SHAPE
      GO TO 550                                                                SHAPE
C                                                                              SHAPE
C     FOR MODIFIED WING                                                        SHAPE
C     COMPUTE MACH NUMBER DISTRIBUTION ON PHYSICAL WING                        SHAPE
  200 DM=0.                                                                    SHAPE
      DO 210 K=1,2                                                             SHAPE
      DO 210 I=1,MB                                                            SHAPE
      DO 210 J=1,MB                                                            SHAPE
  210 AHA(J,I,K)=1.0                                                           SHAPE
      DO 215 I=1,L                                                             SHAPE
```

```
      IF(MLC(1,1,I).EQ.0) GO TO 235
      J=MLC(1,1,I)
      K=MLC(1,2,I)
      CALL SURF2(XX(I),YY,J,K,1,AMA(1,I,1),DM,DM,SFMX,SFMY,SFMM,KSFM,1)
  215 CONTINUE
      IF(KRITE.EQ.0) GO TO 235
C     PRINT OUT MACH DISTRIBUTION
      WRITE(IW,60)
      DO 230 I=1,L
      IF(MLC(1,1,I).EQ.0) GO TO 230
      JL=MLC(1,2,I)-MLC(1,1,I)+1
      IF(JL.EQ.0) GO TO 230
      WRITE(IW,70)  I
      JLP=JL/6
      IF(JL-6*JLP.NE.0) JLP=JLP+1
      K1=MLC(1,1,I)-1
      JL=MLC(1,2,I)
      DO 225 J=1,JLP
      K=K1+J
  225 WRITE(IW,80) ((J1,AMA(J1,I,1)),J1=K,JL,JLP)
  230 CONTINUE
C
  235 CONTINUE
      K=1
      NSP=1.+DA(27)/2. + ERRR
      D2=2.*D
C     DEFINE EDGES OF MODIFIED WING
      DO 240 I=2,NSP
      XEDG(I)=FLOAT(I-1)*D2
      IF(XEDG(I).GT.XEDGI(K)) K=K+1
      YEDG(I)=YEDGI(K-1)+(YEDGI(K)-YEDGI(K-1))*(XEDG(I)-XEDGI(K-1))/
     $               (XEDGI(K)-XEDGI(K-1))
  240 CONTINUE
      IF(IFIX(DA(27))-2*(NSP-1)) 810,255,250
  250 NSP=NSP+1
      NS1=NS(1)
      XEDG(NSP)=XEDGI(NS1)
      YEDG(NSP)=YEDGI(NS1)
  255 CONTINUE
C
      K1=2
      K2=NSP
      NS1=NS(1)
      DO 300 I=2,NS1
      DO 260 K=K1,K2
      K3=K
      IF(ABS(XEDGI(I)-XEDG(K)).LT.ERRR) GO TO 290
      IF(XEDG(K).GT.XEDGI(I)) GO TO 270
  260 CONTINUE
      GO TO 300
  270 K4=K2+K3
      DO 280 J=K3,K2
      K=K4-J
      XEDG(K+1)=XEDG(K)
  280 YEDG(K+1)=YEDG(K)
      NSP=NSP+1
      XEDG(K3)=XEDGI(I)
      YEDG(K3)=YEDGI(I)
      K2=K2+1
  290 K1=K1+1
  300 CONTINUE
      IF(KRITE) 310,320,310
C     PRINT OUT LEADING EDGES TO BE TRANSFORMED
  310 KI=1
      WRITE(IW,65)
      NS(NEW)=NSP
      WRITE(IW,40) K1,NEW,NS(NEW),IEDG
      WRITE(IW,50) (K,XEDG(K),YEDG(K),K=1,NSP)
  320 CONTINUE
      IF(NSP.GT.NSMAX) GO TO 820
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C     CHECK IF MODIFIED WING HAS A FOLD-OVER EDGE
C     TRANSFORM Y-COORDINATE
      DO 350 I=1,NSP
      YMP=0.
      YI=0.
      J=1
      Y=YY(1)
  330 CALL SURF2(XEDG(I),Y,1,1,1,EM,DM,DMDY,SFMX,SFMY,SFMH,KSFN,3)
      YM=Y*EM
      YMD=YM-YMP
      IF(YI.GT.Y) YMD=-YMD
      IF(YMD.GE.0.) GO TO 335
      T(1,1)=XEDG(I)
      T(2,1)=Y
      T(3,1)=EM
      T(4,1)=DMDY
      T(5,1)=YMD
      IER=5
C     ACCEPT A FOLD-OVER OF NOT MORE THAN 1/5 OF A BOX
      IF(ABS(YMD).GT.0.4*DH) GO TO 815
      IERROR=350
      WRITE(IW,20) IERROR,(T(J,1),J=1,IER)
  335 IF(ABS(YM-YEDG(I)).LT.ERRR) GO TO 340
      [illegible]
      J=J+1
      Y=YY(J)
      YMP=YM
      IF(Y.GT.YEDG(I)) Y=YEDG(I)
      GO TO 330
  340 YEDG(I)=YEDG(I)*EM
  350 CONTINUE
C
      IF(KRITE) 360,370,360
C     PRINT OUT LEADING EDGES OF TRANSFORMED WING
  360 K1=2
      WRITE(IW,40) K1,NEW,NS(NEW),IEDG
      WRITE(IW,50) (K,XEDG(K),YEDG(K),K=1,NSP)
  370 CONTINUE
C
C     READJUST THE POINT DISTRIBUTION TO DEFINE NS(2)
      K1=0
      ICHECK=0
      YMP=0.
      DO 450 I=2,NSP
      YMP=AMAX1(YMP,YEDG(I))
      IF(YEDG(I)-YEDG(I-1)) 400,420,440
C     ACCEPT SMALL WAKE BEHIND TRANSFORMED WING LEADING EDGE  AS PART
C         OF WING, IF NOT MORE THAN 1/5 OF A BOX
  400 T(1,1)=XEDG(I)
      T(2,1)=YEDG(I)
      T(3,1)=XEDG(I-1)
      T(4,1)=YEDG(I-1)
      T(5,1)=YMP
      IER=5
      IF(YMP-YEDG(I).GT.0.4*DH) GO TO 805
      IERROR=400
      WRITE(IW,20) IERROR,(T(J,1),J=1,IER)
      YEDG(I) = YMP
  420 IF(I.EQ.ICHECK+1) GO TO 430
      ICHECK=I
      GO TO 440
  430 ICHECK=I
      K1=K1+1
  440 K=I-K1
      XEDG(K)=XEDG(I)
      YEDG(K)=YEDG(I)
  450 CONTINUE
      IF(ABS(1.-XEDG(K)).GT.ERRR) GO TO 800
      IF(YEDG(K)-YEDG(K-1)) 500,460,470
```

```
  460 NS(2)=K-1
      GO TO 480
  470 NS(2)=K
  480 NSP=NS(2)
C
      IF(KRITE) 490,495,490
C     PRINT OUT LEADING EDGES OF TRANSFORMED WING
  490 K1=3
      WRITE(IW,40) K1,NEW,NS(NEW),IEDG
      WRITE(IW,50) (K,XEDG(K),YEDG(K),K=1,NSP)
  495 CONTINUE
C
C     TRAILING EDGE TRANSFORMATION
C
      NST(NEW)=NST(1)
      NSTP    =NST(NEW)
      CALL SURF2(XTDG,YTDG,1,NSTP,0,T(1,1),DM,DM,SFMX,SFMY,SFMH,KSFM,1)
      DO 510 I=1,NSTP
  510 YTDG(I)=YTDG(I)+T(I,1)
      IF(KRITE) 520,530,520
C     PRINT OUT TRAILING EDGES OF TRANSFORMED WING
  520 WRITE(IW,45) NEW,NST(NEW),IEDG
      WRITE(IW,50) (K,XTDG(K),YTDG(K),K=1,NSTP)
  530 CONTINUE
  550 Y1=YEDG(1)
      YMAX(NEW)=YEDG(NSP)
      IF(Y1.LT.0.0) GO TO 860
      IF(NEW.EQ.2) GO TO 565
      AREA=0.0
      XX(1)=DH
      YY(1)=DH
      DO 560 I=2,MB
      YY(I)=YY(I-1)+D
  560 XX(I)=XX(I-1)+D
C
C     CALCULATE PHYSICAL FULL WING AREA  AND  ML(NEW,I)
C     COMPUTE AND IDENTIFY BOX DISTRIBUTION ON THE WING PLANFORM
  565 CALL        PLNFM(XEDG,YEDG,XTDG,YTDG,XX,YY,XLE,XTE,AR,AREA,D
     $                  ,IW,L,MB,ML,MLC,MLW,NEW,NS,NST,IRITE)
C
      IF(ML(NEW,L).GT.MB) GO TO 825
C
      JMAX(NEW)=0
      JMAK(NEW)=0
      DO 570 I=1,L
      JMAK(NEW)=MAX0(JMAK(NEW),MLC(NEW,2,I))
  570 JMAX(NEW)=MAXC(JMAX(NEW),ML(NEW,I))
      JMX=JMAK(NEW)
C
C     FIND ORDER OF LEADING EDGE BOX AT J-TH CHORDWISE ROW
C
      JMA=JMAX(NEW)
      DO 620 J=1,JMA
      EDG(NEW,J)=0.C
      DO 620 K=1,2
  620 MLT(NEW,K,J)=0
      I1=1
      I2=1
      K1=0
      DO 655 J=1,JMX
  630 IF(MLC(NEW,2,I1).GE.J) GO TO 632
      I1=I1+1
      GO TO 630
  632 MZ1=I1
C     FIND ORDER OF TRAILING EDGE BOX AT J-TH CHORDWISE ROW
  634 IF(I2.GT.L) GC TO 638
      IF(MLC(NEW,2,I2).GT.0) K1=1
      IF(K1.EQ.0) GC TO 635
      IF(MLC(NEW,2,I2).EQ.0.AND.MZ1.NE.0) GO TO 646
  635 MLS=IABS(MLW(NEW,I2))
      IF(MLW(NEW,I2).NE.0) GO TO 636
```

```
      I2=I2+1
      K2=1
      GO TO 634
  636 IF(MLS-3) 644,646,638
  638 IF(K2.LT.0) GO TO 648
      IF(MLW(NEW,I2)) 642,642,646
  642 I2=I2+K2
      IF(I2.LE.L) GO TO 634
      NZ2=L
      I2 =L
      GO TO 652
  644 K2=MLW(NEW,I2)/MLS
      GO TO 642
  646 NZ2=I2-1
      GO TO 652
  648 NZ2=I2
  652 CONTINUE
      MLT(NEW,1,J)=NZ1
      MLT(NEW,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(NEW,I)) 664,662,662
  662 MLC(NEW,1,I)=MLW(NEW,I)+1
      IF(MLC(NEW,2,I).EQ.0) MLC(NEW,1,I)=0
      GO TO 670
  664 MLC(NEW,2,I)=IABS(MLW(NEW,I))-1
      MLC(NEW,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(NEW,1,J),MLT(NEW,2,J)
      WRITE(IW,83)
      DO 674 J=1,L
C 674 WRITE(IW,85) J,MLC(NEW,1,J),MLC(NEW,2,J)
  674 WRITE(IW,85) J,MLC(NEW,1,J),MLC(NEW,2,J),MLW(NEW,J),ML(NEW,J)
  678 CONTINUE
C
      IF(XEDG(NSP).GT.1.) GO TO 865
      IF(NEW.EQ.2) GC TO 680
      NSM=NSP-1
      DY=YEDG(NSP)-YEDG(NSM)
  680 CCNTINUE
      IF(ABS(DY).LE.ERRR) IEDG=1
      IF(DY.LT.0.) GO TO 827
C      COMPUTE VALUES FOR LEADING EDGE CORRECTION
      YMAX2=YMAX(NEW)*YMAX(NEW)
      DO 720 J=1,JMX
      IF (IEDG) 710,715,710
C     D IN FOLLOWING EXPRESSION IS ARBITRARY
  710 EDG(NEW,J)=SQRT((YMAX2-YY(J)*YY(J))/(D*(2.*YMAX(NEW)-D)))
      IF(EDG(NEW,J).GT.1.0) GO TO 715
      GO TO 720
  715 EDG(NEW,J)=1.0
  720 CONTINUE
C
      IF(NEW.EQ.1) GO TO 750
C     COMPUTE Y-COORDINATE AND MACH NUMBER OF POINTS ON PHYSICAL WING
C        AT TRAILING EDGE OF TRANSFORMED WING
C        AT BOX CENTERS OF TRANSFORMED WING
      DO 745 J=1,JMX
      K1=MLT(NEW,1,J)
      K2=MLT(NEW,2,J)-K1+2
      I =K1
```

```
      DO 740 K=1,K2                                                     SHAPE
      K3=0                                                              SHAPE
      DEL=0.                                                            SHAPE
      X=XX(I)                                                           SHAPE
      IF(K.EQ.K2) X=XTE(NEW,J)                                          SHAPE
      Y=YY(J)                                                           SHAPE
  730 Y=Y+DEL                                                           SHAPE
      K3=K3+1                                                           SHAPE
C     FIND MACH NUMBER AT X,Y ON PHYSICAL WING                          SHAPE
      CALL SURF2(X,Y,1,1,1,EM,DM,EMY,SFMX,SFMY,SFMM,KSFM,3)             SHAPE
      S1=YY(J)-Y*EM                                                     SHAPE
      IF(ABS(S1).LE.ERRR) GO TO 735                                     SHAPE
      IF(K3.GT.100) GO TO 830                                           SHAPE
      DEL= S1/(EM+Y*EMY+1.E-20)                                         SHAPE
      GO TO 730                                                         SHAPE
  735 IF(K.NE.K2) AMA(J,I,2)=EM                                         SHAPE
      IF(K.EQ.K2) CTE(J    )=EM                                         SHAPE
  740 I=I+1                                                             SHAPE
  745 CONTINUE                                                          SHAPE
  750 CONTINUE                                                          SHAPE
C                                                                       SHAPE
  790 CONTINUE                                                          SHAPE
C                                                                       SHAPE
      RETURN                                                            SHAPE
C                                                                       SHAPE
  800 IERROR=800                                                        SHAPE
      T(1,1)=XEDG(K)                                                    SHAPE
      T(2,1)=YEDG(K)                                                    SHAPE
      IER=2                                                             SHAPE
      GO TO 840                                                         SHAPE
  805 IERROR=805                                                        SHAPE
      GO TO 840                                                         SHAPE
  810 IERROR=810                                                        SHAPE
      T(1,1)=DA(27)                                                     SHAPE
      T(2,1)=NSP                                                        SHAPE
      IER=2                                                             SHAPE
      GO TO 840                                                         SHAPE
  815 IERROR=815                                                        SHAPE
      GO TO 840                                                         SHAPE
  820 IERROR=820                                                        SHAPE
      T(1,1)=XEDG(NSP)                                                  SHAPE
      T(2,1)=YEDG(NSP)                                                  SHAPE
      T(3,1)=NSP                                                        SHAPE
      IER=3                                                             SHAPE
      GO TO 840                                                         SHAPE
  825 IERROR=825                                                        SHAPE
      T(1,1)=NEW                                                        SHAPE
      T(2,1)=K                                                          SHAPE
      T(3,1)=ML(NEW,K)                                                  SHAPE
      T(4,1)=ML(NEW,K-1)                                                SHAPE
      IER=4                                                             SHAPE
      GO TO 840                                                         SHAPE
  827 IERROR=827                                                        SHAPE
      T(1,1)=YEDG(NSP)                                                  SHAPE
      T(2,1)=YEDG(NSP)                                                  SHAPE
      T(3,1)=NSP                                                        SHAPE
      T(4,1)=NSM                                                        SHAPE
      IER=4                                                             SHAPE
      GO TO 840                                                         SHAPE
  830 IERROR=730                                                        SHAPE
      T(1,1)=I                                                          SHAPE
      T(2,1)=J                                                          SHAPE
      T(3,1)=X                                                          SHAPE
      T(4,1)=Y                                                          SHAPE
      T(5,1)=EM                                                         SHAPE
      T(6,1)=EMY                                                        SHAPE
      T(7,1)=DEL                                                        SHAPE
      IER=7                                                             SHAPE
  840 WRITE(IW,20) IERROR, (T(I,1),I=1,IER)                             SHAPE
      STOP                                                              SHAPE
```

```
 850 IPR=19                                                        SHAPE     421
     GO TO 890                                                     SHAPE     422
 855 IPR=24                                                        SHAPE     423
     GO TO 890                                                     SHAPE     424
 860 IPR=30                                                        SHAPE     425
     GO TO 890                                                     SHAPE     426
 865 K=  MINO(K,NS(I))                                             SHAPE     427
     IPR=2*K+29                                                    SHAPE     428
     GO TO 890                                                     SHAPE     429
 871 IPR=45                                                        SHAPE     430
 890 WRITE                                                         SHAPE     431
    C  (IW,10)IPR                                                  SHAPE     432
  10 FORMAT(1H0,10X,17HSHAPE -- BAD DATA,I5)                       SHAPE     433
  20 FORMAT(1H0,10X,38HBAD NUMBER IN SHAPE NEAR STATEMENT NO.,I5,  SHAPE     434
    1   71H ,15X,1P8E14.6)                                         SHAPE     435
  40 FORMAT(/1H0,5X,3HNO.,I2,43H REDISTRIBUTION OF WING LEADING EDGES, SHAPE 436
    1 NS(,I1,4H) = ,I2,7H,  IEDG,3H = ,I1)                         SHAPE     437
  45 FORMAT(/1H0,5X,25HWING TRAILING EDGES, NST(,I1,4H) = ,I2,     SHAPE     438
    $        10H,  IEDG = ,I1)                                     SHAPE     439
  50 FORMAT(4(6X,I3,1P2E11.4))                                     SHAPE     440
  60 FORMAT(1H1,5X,47HLOCAL MACH NUMBER DISTRIBUTION ON PHYSICAL WING/) SHAPE 441
  65 FORMAT(1H1)                                                   SHAPE     442
  70 FORMAT(1H0,5X,I2,19H-TH SPANWISE COLUMN)                      SHAPE     443
  75 FORMAT(1H0,5X,I2,17H-TH CHORDWISE ROW)                        SHAPE     444
  77 FORMAT(40X,13HPHYSICAL WING///)                               SHAPE     445
  78 FORMAT(40X,16HTRANSFORMED WING///)                            SHAPE     446
  80 FORMAT(1H ,5X,6(2X,I3,1PE13.5))                               SHAPE     447
  82 FORMAT(15X,68HORDER OF FIRST(LEADING) AND LAST(TRAILING) WING BOX SHAPE  448
    1IN CHORDWISE ROW//20X,1HJ,3X,12HMLT(NEW,1,J),3X,12HMLT(NEW,2,J)/) SHAPE  449
  83 FORMAT(///15X,62HORDER OF FIRST(ROOT) AND LAST(TIP) WING BOX IN SP SHAPE 450
    1ANWISE COLUMN//20X,1HI,3X,12HMLC(NEW,1,I),3X,12HMLC(NEW,2,I)  SHAPE     451
    2    ,4X,11HMLW(NEW,I),5X,10HML(NEW,I))/)                      SHAPE     452
  85 FORMAT(16X,I5,5X,I5,3(10X,I5))                                SHAPE     453
  86 FORMAT(20X,58HGAUSSIAN INTEGRATION POINTS IN CHORDWISE ROW - GX(NE SHAPE 454
    1W,K,J)//)                                                     SHAPE     455
  87 FORMAT(///20X,52HGAUSSIAN INTEGRATION POINTS IN SEMI-SPAN - GY(NEW SHAPE 456
    1,K)//)                                                        SHAPE     457
  88 FORMAT(///20X,60HSPANWISE COORDINATE (GYP) AND MACH NUMBER (GMP) A SHAPE 458
    ST GX(2,K,J)//)                                                SHAPE     459
  89 FORMAT(1H ,5X,3(2X,I3,1P2E16.6))                              SHAPE     460
     STOP                                                          SHAPE     461
     END                                                           SHAPE     462
```

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

C-2

ORIGINAL PAGE IS
OF POOR QUALITY

```
      GO TO 290
  280 IF(Y-YL+ERRR.GE.D) GO TO 290
      AR(J,I,NEW)=AR(J,I,NEW)+ASNG*(YL-Y+D)/D
      X1=XL
      Y1=YL
      IF(ICHG.NE.1) GO TO 310
      IF(JK) 284,285,285
  284 IF(YE2.LE.Y-D) GO TO 320
      GO TO 286
  285 IF(YE2.GE.Y) GO TO 320
  286 CONTINUE
      IF(XE1.GT.XL) GO TO 400
      IEDZ=2
      GO TO 400
  290 X1=X2
      Y1=Y2
  295 IEDZ=0
      IF(ICHG.NE.1) GO TO 310
      YR=YE2
      GO TO 315
  310 YR=YE1+TNG*(XR-XE1)
  315 IF(JK) 317,316,316
  316 IF(YR.GE.Y) GO TO 320
      GO TO 318
  317 IF(YR.LE.Y-D) GO TO 320
  318 CONTINUE
      IF(ICHG.EQ.1) GO TO 325
      IFIN=1
      X2=XR
      Y2=YR
      GO TO 330
  320 JFIN=1
      IF(JK) 322,321,321
  321 Y2=Y
      GO TO 323
  322 Y2=Y-D
  323 X2=XE1+(Y2-YE1)/TNG
      IF(ABS(X2-XR).GT.ERRR) GO TO 330
      IFIN=1
      JFIN=0
      GO TO 330
  325 IEDZ=1
      X2=XE2
      Y2=YE2
  330 AR(J,I,NEW)=AR(J,I,NEW)+0.5*ASNG*(2.*XR-X2-X1)*(Y2-Y1)/D2
C
C
      IF(ABS(TNG).LT.ERRR) TNG=1.E-20
 1331 CONTINUE
      IF(JK) 332,331,331
  331 IF(J.LT.JLT) GO TO 334
      XLT=XE1+(YY(JLT)-YE1)/TNG
      IF(XLT.GT.XE2.AND.ASNG.GT.0.0) GO TO 334
      IF(YEDG(1).GT.0.0.AND.YY(JLT).LE.YL.AND.ASNG.GT.0.0) XLT=0.0
      IF(YY(JLT).GE.YTDG(NSTP)) XLT=XTDG(NSTP)
      IF(ASNG.GT.0.0) XLE(NEW,JLT)=XLT
      IF(ASNG.LT.0.0) XTE(NEW,JLT)=XLT
      GO TO 333
  332 IF(J.GT.JLT) GO TO 334
      XLT=XE1+(YY(JLT)-YE1)/TNG
      IF(YY(JLT).GE.YTDG(NSTP)) XLT=XTDG(NSTP)
      XTE(NEW,JLT)=XLT
  333 JLT=JLT+JK
      GO TO 1331
  334 CONTINUE
C
      IF(IEDZ.EQ.1) GO TO 400
      SAM=SAM+AR(J,I,NEW)
      IF(JK) 344,342,342
  342 CONTINUE
```

```
      IF(JFIN.EQ.1) GO TO 250
      GO TO 350
  344 IF(JFIN.EQ.1) GO TO 495
  350 IF(ASNG) 355,360,360
  355 IF(IFIN.EQ.1) GO TO 478
      GO TO 530
  360 IF(IFIN.EQ.1) GO TO 465
      GO TO 500
C
C     LEADING EDGE AND TRAILING EDGE COMPUTATION
C
  400 IF(ASNG) 440,405,405
  405 IF(JK) 430,410,410
  410 K1=K1+JK
      IF(K1.GE.NSP) GO TO 420
      XE1=XEDG(K1)
      YE1=YEDG(K1)
      K=K1+JK
      XE2=XEDG(K)
      YE2=YEDG(K)
      GO TO 450
  420 IF(JK.LT.0) GO TO 240
      JK=-1
      JLT=J
      MLC(NEW,2,I)=0
      K2=K2-JK
  430 K2=K2+JK
      XE1=XTDG(K2)
      YE1=YTDG(K2)
      K=K2+JK
      XE2=XTDG(K)
      YE2=YTDG(K)
      GO TO 450
  440 K2=K2+1
      XE1=XTDG(K2)
      YE1=YTDG(K2)
      XE2=XTDG(K2+1)
      YE2=YTDG(K2+1)
  450 G=XE2-XE1
      F=YE2-YE1
      TNG=F/(G+1.E-20)
      IF(G.LT.0.) GO TO 650
      IF(NEW.EQ.2) GO TO 455
      AREA=AREA+ASNG*G*(YE2+YE1)
C
  455 CONTINUE
      ICHQ=0
      IF(XR.GT.XE2) ICHQ=1
      IF(IEDZ.EQ.1) GO TO 290
      IF(IEDZ.EQ.2) GO TO 295
      GO TO 250
C
C     NUMBER OF BOXES IN SPANWISE COLUMN
C     WAKE TYPE AND ORDER OF WAKE BOX
C
  465 IF(ASNG) 478,467,467
  467 IF(JK) 476,469,469
  469 JJ=0
      IF(AR(J,I,NEW).LT.ERR2) JJ=1
      ML(NEW,I)=J-JJ
C
      JJ=ML(NEW,I)
      DO 471 K=1,JJ
      KK=K
      IF(AR(K,I,NEW).LT.AWAK) GO TO 472
  471 CONTINUE
      MLC(NEW,2,I)=KK
      GO TO 473
  472 MLC(NEW,2,I)=KK-1
  473 CONTINUE
```

```
C      ADJUSTMENT OF FIRST BOX AT LEADING EDGE ALONG KK-TH CHORDWISE ROW  PLNFM   211
       KK=MLC(NEW,2,I)                                                    PLNFM   212
       IF(XX(I).GE.XLE(NEW,KK)) GO TO 474                                 PLNFM   213
       MLC(NEW,2,I)=MLC(NEW,2,I)-1                                        PLNFM   214
       GO TO 473                                                          PLNFM   215
   474 CONTINUE                                                           PLNFM   216
C                                                                         PLNFM   217
       GO TO 500                                                          PLNFM   218
   476 MLS=-1                                                             PLNFM   219
       ML(NEW,I)=ML(NEW,I-1)                                              PLNFM   220
       K=ML(NEW,I)                                                        PLNFM   221
       GO TO 480                                                          PLNFM   222
   478 MLS=1                                                              PLNFM   223
       K=J                                                                PLNFM   224
   480 DO 490 JJ=1,K                                                      PLNFM   225
       IF(AR(JJ,I,NEW)) 481,484,481                                       PLNFM   226
   481 IF(ABS(AR(JJ,I,NEW))-ERR2) 483,483,482                             PLNFM   227
C                                                                         PLNFM   228
   482 WRITE(IW,96) ERR2,JJ,I,AR(JJ,I,NEW)                                PLNFM   229
C                                                                         PLNFM   230
   483 AR(JJ,I,NEW)=C.0                                                   PLNFM   231
   484 CONTINUE                                                           PLNFM   232
       IF(MLW(NEW,I).NE.0) GO TO 490                                      PLNFM   233
       IF(ASNG) 487,486,486                                               PLNFM   234
C      WIG TIP WAKE                                                       PLNFM   235
   486 IF(AR(JJ,I,NEW).GE.AWAK) GO TO 490                                 PLNFM   236
       MLW(NEW,I)=JJ                                                      PLNFM   237
       GO TO 488                                                          PLNFM   238
C      WING ROOT WAKE                                                     PLNFM   239
   487 IF(AR(JJ,I,NEW).LT.AWAK) GO TO 490                                 PLNFM   240
       MLW(NEW,I)=JJ-1                                                    PLNFM   241
   488 MLW(NEW,I)=MLW(NEW,I)+MLS                                          PLNFM   242
   490 CONTINUE                                                           PLNFM   243
       GO TO 500                                                          PLNFM   244
C                                                                         PLNFM   245
C      TRAILING EDGE BOX AREA ADJUSTMENT FOR WING TIP WAKE                PLNFM   246
C                                                                         PLNFM   247
   495 CONTINUE                                                           PLNFM   248
       J=J-1                                                              PLNFM   249
       Y=Y-J                                                              PLNFM   250
       JFIN=0                                                             PLNFM   251
       SAM=SAM-AR(J,I,NEW)                                                PLNFM   252
       GO TO 290                                                          PLNFM   253
C                                                                         PLNFM   254
C      END OF A SPANWISE COLUMN                                           PLNFM   255
C                                                                         PLNFM   256
   500 CONTINUE                                                           PLNFM   257
       IF(ASNG.GT.0.) GO TO 516                                           PLNFM   258
C      ADJUST WING TIP BOX FOR WING ROOT WAKE                             PLNFM   259
       JJ=ML(NEW,I)                                                       PLNFM   260
   510 IF(AR(JJ,I,NEW).LT.AWAK) GO TO 512                                 PLNFM   261
       MLC(NEW,2,I)=JJ                                                    PLNFM   262
       GO TO 516                                                          PLNFM   263
   512 JJ=JJ-1                                                            PLNFM   264
       IF(JJ) 514,514,510                                                 PLNFM   265
   514 MLC(NEW,2,I)=C                                                     PLNFM   266
       IF(MLW(NEW,I).EQ.0) GO TO 516                                      PLNFM   267
       WRITE(IW,155) I,I,MLW(NEW,I),I,MLC(NEW,2,I)                        PLNFM   268
       STOP                                                               PLNFM   269
   516 CONTINUE                                                           PLNFM   270
       SUM=SUM+SAM                                                        PLNFM   271
C                                                                         PLNFM   272
   530 CONTINUE                                                           PLNFM   273
C                                                                         PLNFM   274
       IF(ASNG.LT.0.C) GO TO 700                                          PLNFM   275
       IF(XTOG(1).GE.XTOG(NSTP)) GO TO 700                                PLNFM   276
C                                                                         PLNFM   277
C      INITIALIZATION FOR WING ROOT WAKE                                  PLNFM   278
C                                                                         PLNFM   279
       K2=0                                                               PLNFM   280
```

```
      XR=0.0                                                            PLNFM    281
      YR=0.0                                                            PLNFM    282
      XE2=XTDG(1)                                                       PLNFM    283
      YE2=YTDG(1)                                                       PLNFM    284
      X2=XE2                                                            PLNFM    285
      Y2=YE2                                                            PLNFM    286
      ASNG=-1.0                                                         PLNFM    287
      JLT=1                                                             PLNFM    288
      GO TO 220                                                         PLNFM    289
C     ERROR MESSAGE                                                     PLNFM    290
  650 K=450                                                             PLNFM    291
      A=G                                                               PLNFM    292
      Z=Z2                                                              PLNFM    293
      WRITE(IW,150) K,Z,A                                               PLNFM    294
      STOP                                                              PLNFM    295
C                                                                       PLNFM    296
  700 CONTINUE                                                          PLNFM    297
      IF(JRITE.EQ.0) GO TO 750                                          PLNFM    298
      WRITE(IW,65)                                                      PLNFM    299
      IF(NEW.EQ.1) WRITE(IW,66)                                         PLNFM    300
      IF(NEW.EQ.2) WRITE(IW,67)                                         PLNFM    301
      DO 750 I=1,L                                                      PLNFM    302
      JL=ML(NEW,I)                                                      PLNFM    303
      K=JL/5                                                            PLNFM    304
      IF(JL.LE.5*K) GO TO 710                                           PLNFM    305
      K=K+1                                                             PLNFM    306
  710 WRITE(IW,70) I                                                    PLNFM    307
      WRITE(IW,72) MLC(NEW,2,I)                                         PLNFM    308
      IF(MLW(NEW,I)) 715,725,720                                        PLNFM    309
  715 WRITE(IW,75) MLW(NEW,I)                                           PLNFM    310
      GO TO 725                                                         PLNFM    311
  720 WRITE(IW,80) MLW(NEW,I)                                           PLNFM    312
  725 DO 740 JJ=1,K                                                     PLNFM    313
  740 WRITE(IW,85) (J,AR(J,I,NEW), J=JJ,JL,K)                           PLNFM    314
  750 CONTINUE                                                          PLNFM    315
C                                                                       PLNFM    316
      JL=ML(NEW,L)                                                      PLNFM    317
      WRITE(6,140)                                                      PLNFM    318
      WRITE(6,145) (J,XLE(NEW,J),XTE(NEW,J), J=1,JL)                    PLNFM    319
C                                                                       PLNFM    320
      SUM=2.*SUM*DZ                                                     PLNFM    321
      WRITE(IW,90) AREA,SUM                                             PLNFM    322
  760 RETURN                                                            PLNFM    323
   65 FORMAT(1H1,20X,37HNON-DIMENSIONAL BOX-AREA DISTRIBUTION)          PLNFM    324
   66 FORMAT(1H+,57X,21H - ( PHYSICAL   WING)//)                        PLNFM    325
   67 FORMAT(1H+,57X,21H - (TRANSFORMED WING)//)                        PLNFM    326
   70 FORMAT(//2X,I2,19H-TH SPANWISE COLUMN/)                           PLNFM    327
   72 FORMAT(12X,16HFIRST L.E. BOX =,I3,22H-TH BOX FROM WING ROOT)      PLNFM    328
   75 FORMAT(12X,16HFIRST WAKE BOX =,I3,22H-TH BOX FROM WING ROOT/)     PLNFM    329
   80 FORMAT(12X,16HLAST  WAKE BOX =,I3,22H-TH BOX FROM WING ROOT/)     PLNFM    330
   85 FORMAT(5X,5(1X,I3,1PE11.4))                                       PLNFM    331
   90 FORMAT(//10X,25HWING AREA CALCULATED FROM/                        PLNFM    332
     1           15X,28HLEADING AND TRAILING EDGES =,1PE11.4/           PLNFM    333
     2           15X,28HSUMMATION OF AREA OF BOXES =,1PE11.4)           PLNFM    334
   96 FORMAT(//10X,43HNEGATIVE BOX AREA EXCEEDS ALLOWABLE LIMIT (,      PLNFM    335
     1            1PE11.4,1H)/10X,3HAR(,I3,1H,,I3,1H,,6HNEW) =,1PE11.4, PLNFM    336
     2            5X,21HCOMPUTATION CONTINUES//)                        PLNFM    337
  140 FORMAT(1H1,9X,54HLEADING AND TRAILING EDGE POINTS AT EACH CHORDWISPLNFM    338
     *SE ROW/13X,1HJ,6X,7HLEADING,8X,8HTRAILING/)                       PLNFM    339
  145 FORMAT(10X,I3,1P2E15.5)                                           PLNFM    340
  150 FORMAT(1H0,4X,29HPLNFM--NEGATIVE VALUE NEAR SN,I4,2X,A2,1PE11.4)  PLNFM    341
  155 FORMAT(1H0,4X,25HORDER OF WING TIP BOX IN ,I2,43H-TH SPANWISE COLUPLNFM    342
     1MN IS NOT PROPERLY DEFINED//                                      PLNFM    343
     2             15X,8HMLW(NEW,,I2,3H) =,I3/15X,8HMLL(NEW,,I2,3H) =,I3)PLNFM    344
      END                                                               PLNFM    345
```

```
      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 RANGE OF THE SECOND SUBSCRIPT IN THE ARRAY,
C     DIMENSIONED A(2,M2,N2), BUT TREATED HERE AS AN ARRAY
C     WITH TWO SUBSCRIPTS
C
      DIMENSION A(2,1)
      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 I1=1,2
      K1=1
```

```
      DO 5 J=1,N
      DO 4 I=1,M1
      K=K1+M-I
    4 A(IL,K)=A(IL,K)-A(IL,K-1)
      A(IL,K1)=2.0*A(IL,K1)
    5 K1=K1+M2
      CM=0.0
      DM=0.0
      DDM=DK
      DO 12 I=1,M
      C7=0.0
      C8=0.0
      C9=0.0
      C10=0.0
      P1=0.0
      P2=0.0
      CN=1.0
      B6=0.5*DK12
      K=I
      DO 10 J=1,N
      A1=CM/CN
      A2=DM/CN
      IF (A1-0.2) 7,7,8
    7 B1=2.0-A1**2/3.0
      B2=-DK/(6.0*CN)
      GO TO 9
    8 B3= SIN(A1)/A1
      B1=2.0*B3
      B2=(B3- COS(A1))/A2-DM/CN*B3
    9 B3= COS(A2)/CN
      B4= SIN(A2)/CN
      C3=B1*B3+B2*C4
      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,1),SFDH(NM,MD,1),KSFD(1)
     1          ,SFMX(1),SFMY(1),SFMH(1),DA(1),T(NM,1),MLC(2,2,1)
     2          ,XX(1),YY(1)
C
      ERRR=1.E-06
      IRITE=IPRINT/10
      JRITE=IPRINT/100
      IF(NEW.EQ.1) IRITE=IPRINT-10*IRITE
      IF(NEW.EQ.2) GO TO 400
      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,1)=DA(KP+2)/DA(24)
      SFDH(IP,M,2)=DA(KP+3)
      SFDH(IP,M,1)=SFDH(IP,M,2)
  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 SURF1(NM,KSFD(M),T,SFDX(1,M),SFDY(1,M,1),SFDH(1,M,1),IRITE)
      GO TO 200
C     PRESENTLY FOR PITCH AND PLUNGE OF FORM  Z=A0+A1*X
  170 SFDH(1,M,1)=DA(52)
      SFDH(2,M,1)=DA(53)*DA(24)
      SFDH(3,M,1)=0.0
      GO TO 145
  200 ICHECK=0
      RETURN
C
C     FOR MODIFIED WING
C     1-CALCULATE DEFLECTION ON THE ORIGINAL WING
C     2-TRANSFORM X, Y, DEF TO THE TRANSFORMED PLANE
C     3-SURFACE FIT DATA
C
  400 CONTINUE
      IF(KSFD(M).EQ.0) GO TO 500
      KP=KSFD(M)
C     CALCULATE LOCAL MACH NUMBER ON PHYSICAL WING (INPUT POINTS)
      CALL SURF2(SFDX(1,M),SFDY(1,M,1),1,KP,0,T(1,1),DUM,DUM,SFMX,SFMY
     $          ,SFMH,KSFM,1)
C     STORE TRANSFORMED DATA
      DO 450 I=1,KP
  450 SFDY(I,M,2)=SFDY(I,M,1)*T(I,1)
C     FIT DEFLECTION DATA IN TRANSFORMED SPACE
      CALL SURF1(NM,KP,T,SFDX(1,M),SFDY(1,M,2),SFDH(1,M,2),IRITE)
      RETURN
  500 ICHECK=ICHECK+1
      IF(ICHECK.GT.1) GO TO 560
C     SELECT POINTS ON TRANSFORMED WING FOR DEFLECTION SURFACE-FITTING
      MS=M
      CALL         SLCT(L,MS,NEW,MLC,KSFS,IW
     $              ,XX,YY,SFDX(1,M),SFDY(1,M,2),JRITE)
      IF(JRITE.GT.0) WRITE(IW,55)
```

```
C     COMPUTE Y-COORDINATE OF POINTS ON PHYSICAL WING CORRESPONDING      DRED      71
C        TO THOSE SELECTED ON TRANSFORMED WING                           DRED      72
      DO 550 K=1,KSFS                                                    DRED      73
      DEL=0.0                                                            DRED      74
      X=SFDX(K,M)                                                        DRED      75
      Y=SFDY(1,M,2)                                                      DRED      76
  520 Y=Y+DEL                                                            DRED      77
C     FIND MACH NUMBER AT (X,Y) ON PHYSICAL WING                         DRED      78
      CALL        SURF2(X,Y,1,1,1,EM,DUM,EMY,SFMX,SFMY,SFMH,KSFM,3)      DRED      79
      S1=SFDY(K,M,2)-Y*EM                                                DRED      80
      IF(ABS(S1).LE.ERRR) GO TO 530                                      DRED      81
      DEL=S1/(EM+Y*EMY+1.E-20)                                           DRED      82
      GO TO 520                                                          DRED      83
C     STORE Y OF PHYSICAL SPACE                                          DRED      84
  530 SFDY(K,M,1)=SFDY(K,M,2)/(EM+1.E-20)                                DRED      85
  550 CONTINUE                                                           DRED      86
      GO TO 580                                                          DRED      87
  560 DO 570 K=1,KSFS                                                    DRED      88
      SFDX(K,M)  =SFDX(K,MK)                                             DRED      89
      SFDY(K,M,1)=SFDY(K,MK,1)                                           DRED      90
  570 SFDY(K,M,2)=SFDY(K,MK,2)                                           DRED      91
  580 CONTINUE                                                           DRED      92
C     COMPUTE DEFLECTION AT SELECTED POINTS ON TRANSFORMED WING          DRED      93
C     BY CALCULATING DEFLECTION ON THE CORRESPONDING POINTS ON THE       DRED      94
C     PHYSICAL WING                                                      DRED      95
      CALL        SURF2(SFDX(1,M),SFDY(1,M,1),1,KSFS,0,T(1,1),DUM,DUM    DRED      96
     $           ,SFDX(1,M),SFDY(1,M,1),SFDH(1,M,1),KSFD(M),1)           DRED      97
      DO 610 I=1,KSFS                                                    DRED      98
  610 SFDH(I,M,NEW)=T(I,1)                                               DRED      99
C                                                                        DRED     100
      IF(IRITE) 620,630,620                                              DRED     101
  620 WRITE(IW,40) M                                                     DRED     102
      WRITE(IW,10)                                                       DRED     103
  630 CONTINUE                                                           DRED     104
C                                                                        DRED     105
C     FIT DEFLECTION OF TRANSFORMED WING BASED ON SELECTED POINTS        DRED     106
      CALL        SURF1(NM,KSFS,T,SFDX(1,M),SFDY(1,M,2),SFDH(1,M,2)      DRED     107
     $           ,IRITE)                                                 DRED     108
      RETURN                                                             DRED     109
  730 IPR=98                                                             DRED     110
      GO TO 750                                                          DRED     111
  750 WRITE                                                              DRED     112
     0  (IW,20)IPR                                                       DRED     113
      STOP                                                               DRED     114
   10 FORMAT(1H0,10X,73HCOMPUTED DEFLECTION = A0+A1*X+A2*Y+ SUM OF H(I)* DRED     115
     1(K(I)**2)*(ALOG(K(I)**2)))                                         DRED     116
   20 FORMAT(1H0,10X,16H[illegible] -- BAD DATA,I5)                      DRED     117
   30 FORMAT(1H0,8X,26HPHYSICAL PLANE -- MODE NO.,I3)                    DRED     118
   40 FORMAT(1H0,8X,29HTRANSFORMED PLANE -- MODE NO.,I3)                 DRED     119
   55 FORMAT(1H1)                                                        DRED     120
      END                                                                DRED     121
```

```
      SUBROUTINE SLCT(L,NB,NEW,MLC,ISLCT,IW,XX,YY,XI,YI,JRITE)
C     L   --- INPUT  (NUMBER OF SPANWISE ROWS)
C     NB  --- INPUT  (MAXIMUM NUMBER OF POINTS ALLOWED IN DIMENSION)
C     MLC --- INPUT  (ORDER OF FIRST AND LAST WING BOX OF I-TH COLUMN)
C     NEW --- INPUT  (SELECT FROM  1-PHYSICAL,  2-TRANSFORMED)
C     JRITE- INPUT  (INTERMEDIATE RESULTS PRINT-OUT CONTROL)
C     XX,YY- INPUT  (COORDINATES OF BOX GRID)
      DIMENSION MLC(2,2,1),XX(1),YY(1),XI(1),YI(1)
      NTOT=0
      DO 20 I=1,L
      IF(MLC(NEW,1,I).EQ.0) GO TO 20
      NTOT=NTOT+MLC(NEW,2,I)-MLC(NEW,1,I)+1
   20 CONTINUE
      NP=0
      NPT=0
      I=0
      K=1
      DJ=0.0
      DN=FLOAT(NTOT)/FLOAT(NB)
      IF(DN.LT.1.0) DN=1.0
      IF(JRITE.EQ.0) GO TO 25
      WRITE(IW,200)
      IF(NEW.EQ.1) WRITE(IW,210)
      IF(NEW.EQ.2) WRITE(IW,220)
      WRITE(IW,230)
   25 NP=NP+1
   30 IF(K.LE.NPT) GO TO 50
      I=I+1
      NPTP=NPT
      IF(MLC(NEW,1,I).EQ.0) GO TO 30
      NPT=NPT+MLC(NEW,2,I)-MLC(NEW,1,I)+1
      GO TO 30
   50 XI(NP)=XX(I)
      J=K-NPTP
      YI(NP)=YY(J)
      IF(JRITE.GT.0) WRITE(IW,100) I,J,K,XI(NP),YI(NP)
      IF(NP.GT.NB) GO TO 60
      DJ=DJ+DN
      NJ=DJ+0.6
      DJ=DJ-FLOAT(NJ)
      K=K+NJ
      IF(K.GT.NTOT) GO TO 80
      GO TO 25
   60 WRITE(IW,250)
      DO 70 I=1,NP
   70 WRITE(IW,110) I,XI(I),YI(I)
      STOP
   80 CONTINUE
      ISLCT=NP
      RETURN
  100 FORMAT(5X,3I5,1P4E14.7)
  110 FORMAT(1H ,10X,I4,1P4E14.7)
  200 FORMAT(1H1,4X,19HSELECTED POINTS ON )
  210 FORMAT(1H+,23X,12H PHYSICAL    )
  220 FORMAT(1H+,23X,12HTRANSFORMED )
  230 FORMAT(1H+,35X,31HWING FOR SPLINE-SURFACE FITTING//
     $9X,1HI,4X,1HJ,4X,1HK,4X,5HXI(K),9X,5HYI(K)/)
  250 FORMAT(1H0,5X,59HNUMBER OF POINTS SELECTED IN  SLCT  EXCEEDS ALLOW
     $ABLE LIMIT//14X,1HI,3X,5HXI(I),9X,5HYI(I)/)
      END
```

```
      SUBROUTINE WVAL(XX,YY,XTDG,YTDG,XTE,SFDX,SFDY,SFDH,KSFD,S,D,CK
     $                ,IW,L,ML,M1,NEW,NST,MB,MD,NB,NM,KSFS,IPRINT)
C     CALCULATES DOWNWASH VELOCITY DISTRIBUTION (REAL AND IMAGINARY)
      DIMENSION XX(1),YY(1),XTDG(1),YTDG(1),XTE(2,1),S(2,MB,1),ML(2,1)
     $          ,NST(1),SFDX(NB,1),SFDY(NB,MD,1),SFDH(NM,MD,1),KSFD(1)
      IRITE=IPRINT/1C
      IF(NEW.EQ.1) IRITE=IPRINT-10*IRITE
      JLP=KSFD(M1)
      IF(KSFD(M1).EC.0.AND.NEW.EQ.2) JLP=KSFS
      JL=ML(NEW,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(NEW,I)
      IF(JL) 40,8C,4C
   40 CONTINUE
      DO 70 J=1,JL
      IF(XX(I).GT.(XTE(NEW,J)+D)) GO TO 70
      CALL SURF2(XX(I),YY(J),1,1,1,VALU,VALUD,DM,SFDX(1,M1)
     $           ,SFCY(1,M1,NEW),SFDH(1,M1,NEW),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)
      IF(NEW.EQ.1) WRITE(IW,12) M1
      IF(NEW.EQ.2) WRITE(IW,14) M1
   10 FCRMAT(1H1,10X,47HUPWASH (REAL, IMAGINARY, ABSOLUTE, PHASE ANGLE))
   12 FORMAT(1H+,58X,28H---PHYSICAL PLANE---MODE NO.,I3/)
   14 FORMAT(1H+,58X,31H---TRANSFORMED PLANE---MODE NO.,I3/)
      DO 170 I=1,L
      JL=ML(NEW,I)
      IF (JL) 110,17C,110
  110 WRITE(IW,20) I
      JLP=JL/2
      IF(JL-2*JLP.NE.0) JLP=JLP+1
      DO 160 J=1,JLP
      S1=SQRT(S(1,J,I)*S(1,J,I)+S(2,J,I)*S(2,J,I))
      IF(S1.GT.0.C) GO TO 120
      S2=0.0
      GC TO 130
  120 S2=57.29578*ATAN2(S(2,J,I),S(1,J,I))
  130 J1=J+JLP
      IF(J1.LE.JL) GC 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,I2,17H-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(2,1),XLE(2,1),A(2,MC,MB)
     1          ,AR(MB,MB,2),EDG(2,1),S(2,MB,MB),T(ME,ME,1),NST(1)
     2          ,ML(2,1),MLW(2,1),MLT(2,2,1),MLC(2,2,MB),JMAX(2)
     3          ,AMA(MB,MB,1)
      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
     $               ,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(NEW)
      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 IF(NEW.EQ.1) WRITE(IW,32) M1
      IF(NEW.EQ.2) WRITE(IW,34) M1
   30 FORMAT(1H1,10X,61HPOTENTIAL CALCULATED (REAL, IMAGINARY, ABSOLUTE,
     $ PHASE ANGLE))
   32 FORMAT(1H+,71X,26H---PHYSICAL PLANE---MODE NO.,I3/)
   34 FORMAT(1H+,71X,31H---TRANSFORMED PLANE---MODE NO.,I3/)
C     PRINT-OUT
  216 DO 250 I=1,L
      JL=ML(NEW,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=0.0
      T(J1,I,1)=0.0
      T(J1,I,2)=0.0
      GO TO 240
  230 CONTINUE
      S3=SQRT(T(J1,I,1)*T(J1,I,1)+T(J1,I,2)*T(J1,I,2))
      IF(S3.NE.0.0) GO TO 234
      S4=0.0
      GO TO 236
  234 CONTINUE
      S4=57.29578*ATAN2(T(J1,I,2),T(J1,I,1))
  236 CONTINUE
  240 WRITE(IW,25) J,T(J,I,1),T(J,I,2),S1,S2
     *             ,J1,T(J1,I,1),T(J1,I,2),S3,S4
  250 CONTINUE
C
  270 CONTINUE
      IF(ICHECK.GE.1) GO TO 400
C     LEADING EDGE CORRECTION
      DO 380 J=1,JMAK
      IF (MLT(NEW,1,J).EQ.0) GO TO 285
      I1=MLT(NEW,1,J)
      I2=MLT(NEW,2,J)
      S3=0.0
      DO 280 I=I1,I2
      K=I
      IF(AR(J,I,NEW).GE.1.0) GO TO 290
  280 CONTINUE
  285 S3=D
  290 IF(K.EQ.1) S3=D
      S3=XX(K)-XLE(NEW,J)-DH+S3
      DO 370 I=1,L
      S1=(XX(I)-XLE(NEW,J))/S3
      IF(S1) 300,340,310
  300 S1=0.0
      GO TO 340
  310 IF(YE) 320,320,330
  320 S1=S1*(XX(I)+XLE(NEW,J))/(S3+2.*XLE(NEW,J))
  330 IF(S1.GT.1.0) S1=1.0
      S1=EDG(NEW,J)*SQRT(S1)
  340 DO 350 K=1,2
  350 S(K,J,I)=S1*S(K,J,I)
  370 CONTINUE
  380 CONTINUE
      ICHECK=1
      IF(IRITE) 390,400,390
  390 WRITE(IW,36)
   36 FORMAT(1H1,10X,61HPOTENTIAL CORRECTED  (REAL, IMAGINARY, ABSOLUTE,
     1 PHASE ANGLE))
C     TRANSFER TO THE PRINT-OUT OF CORRECTED VELOCITY POTENTIAL
      GO TO 200
  400 CONTINUE
      IF(KRITE.EQ.0) GO TO 700
      IF(ICHECK.GE.2) GO TO 520
C     CALCULATE AND PRINT PRESSURE COEFFICIENT
      ICHECK=2
      DO 410 I=1,L
      JL=ML(NEW,I)
      DO 410 J=1,JL
      T(J,I,3)=S(1,J,I)
  410 T(J,I,4)=S(2,J,I)
  420 JMAK=JMAX(NEW)
      DO 430 I=1,L
      JL=ML(NEW,I)
      DO 430 J=1,JL
      DO 430 K=1,2
  430 T(J,I,K)=0.0
      DO 500 J=1,JMAK
      I1=MLT(NEW,1,J)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      I2=MLT(MEW,2,J)-MLT(MEW,1,J)+1
      II=I1
      IK=I2+1
      T(1, 6,5)=XLE(MEW,J)
      T(1, 8,5)=XLE(MEW,J)
      T(1,10,5)=0.0
      T(1,12,5)=0.0
      IF(ABS(XLE(MEW,J)-XX(II)).GT.1.E-05) GO TO 445
C     ADJUSTMENT -- LEADING EDGE AND FIRST BOX COINCIDES
      II=II+1
      IK=IK-1
  445 CONTINUE
      DO 450 I=2,IK
      T(I, 6,5)=XX(II)
      T(I, 8,5)=XX(II)
      T(I,10,5)=T(J,II,3)
      T(I,12,5)=T(J,II,4)
      T(I-1,1,5)=XX(II)
  450 II=II+1
      CALL SPLN1(I2,T(1, 1,5),T(1, 2,5),T(1, 4,5)
     $          ,IK,T(1, 6,5),T(1,10,5),T(1,14,5),NSMOS)
      CALL SPLN1(I2,T(1, 1,5),T(1, 3,5),T(1, 5,5)
     $          ,IK,T(1, 8,5),T(1,12,5),T(1,14,5),NSMOS)
      II=I1
      DO 470 I=1,I2
      T(J,II,1)=T(I,4,5)-CK*T(I,3,5)
      T(J,II,2)=T(I,5,5)+CK*T(I,2,5)
  470 II=II+1
  500 CONTINUE
      WRITE(IW,40)
   40 FORMAT(1H1,10X,61HPRESSURE COEFFICIENT (REAL, IMAGINARY, ABSOLUTE,
     $ PHASE ANGLE))
C     TRANSFER TO PRINT-OUT SECTION
      GO TO 215
  520 IF(MEW.EQ.1.OR.ICHECK.EQ.3) GO TO 700
C     CALCULATE PRESSURE COEFFICIENT ON PHYSICAL WING FROM NEW=2 RESULTS
      DO 600 I=1,L
      J1=MLC(1,1,I)
      J2=MLC(1,2,I)
      J3=J2-J1+1
      J4=J1
C     TRANSFORMED Y-COORDINATE OF PHYSICAL WING ALONG I-TH COLUMN
      DO 540 J=1,J3
      T(J,1,5)=YY(J4)*AMA(J4,I,1)
  540 J4=J4+1
      JA=MLC(2,1,I)
      JB=MLC(2,2,I)
      JC=JB-JA+1
      JD=JA
C     KNOWN POINTS OF TRANSFORMED WING ALONG I-TH COLUMN
      DO 560 J=1,JC
      T(J, 6,5)=YY(JD)
      T(J, 8,5)=YY(JD)
      T(J,10,5)=S(1,JD,I)
      T(J,12,5)=S(2,JD,I)
  560 JD=JD+1
C     INTERPOLATE S AT T(J,1,5) OF TRANSFORMED WING
      CALL SPLN1(J3,T(1, 1,5),T(1, 2,5),T(1, 4,5)
     $          ,JC,T(1, 6,5),T(1,10,5),T(1,14,5),NSMOS)
      CALL SPLN1(J3,T(1, 1,5),T(1, 3,5),T(1, 5,5)
     $          ,JC,T(1, 8,5),T(1,12,5),T(1,14,5),NSMOS)
C     TRANSFORM S INTO PHYSICAL QUANTITY
      J4=J1
      DO 580 J=1,J3
      T(J4,I,3)=T(J,2,5)/AMA(J4,I,1)
      T(J4,I,4)=T(J,3,5)/AMA(J4,I,1)
  580 J4=J4+1
  600 CONTINUE
      MEW=1
      ICHECK=3
```

```
C      TRANSFER TO POTENTIAL INTERPOLATING SECTION TO OBTAIN              BOXP     211
C      VALUES FOR PRESSURE COEFFICIENT COMPUTATION ON PHYSICAL WING       BOXP     212
       GO TO 420                                                          BOXP     213
   700 CONTINUE                                                           BOXP     214
       RETURN                                                             BOXP     215
    20 FORMAT(1H0,5X,I2,17H-TH SPANWISE COLUMN)                           BOXP     216
    25 FORMAT(1H ,5X,4(2X,I3,1P4E13.5))                                   BOXP     217
       END                                                                BOXP     218
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE BOXPO(XX,YY,XTDG,YTDG,XTE,A,AR,DAN,T,S,CK,D
     $                  ,IW,L,MB,MC,ML,MLC,MLW,M1,NEW,NST,JRITE)
C     SOLUTION OF SIMULTANEOUS EQUATIONS FOR THE POTENTIAL
      DIMENSION XX(1),YY(1),XTDG(1),YTDG(1),XTE(2,1),A(2,MC,1)
     1         ,AR(MB,MB,1),T(2,MB,1),S(2,MB,1)
     2         ,ML(2,1),MLC(2,2,1),MLW(2,1),NST(1)
      DH=0.5*D
      IF(JRITE.EQ.0) GO TO 25
      IF(NEW.EQ.2.OR.M1.GT.1) GO TO 25
C     PRINT INFLUENCE COEFFICIENT
      WRITE(IW,100)
      JL=2*ML(NEW,L)
      K=JL/4
      IF(JL-4*K.NE.0) K=K+1
      DO 20 I=1,L
      I1=I-1
      WRITE(IW,110) I1
      DO 15 J1=1,K
   15 WRITE(IW,120) ((J,A(1,J,I),A(2,J,I)),J=J1,JL,K)
   20 CONTINUE
   25 CONTINUE
      I1=NST(NEW)
      AMI=1.0
      DO 26 I=1,I1
   26 AMI=AMIN1(XTDG(I),AMI)
C
      IF(JRITE.EQ.0) GO TO 28
      WRITE(IW,168)
   28 CONTINUE
      I1=0
      NFLNS = DAN
      DO 90 I=1,L
      X=XX(I)
C     ADJUST UPSTREAM INFLUENCE
      KO = 1
      IF(NFLNS.EQ.0) GO TO 30
      KO = MAX0(1,I-NFLNS+1)
   30 CONTINUE
      JL=ML(NEW,I)
      IF (JL.EQ.0) GO TO 90
C     DEFINE WING AND WAKE BOXES
C                                               JW=NO. OF WAKE BOXES IN ROW
C                                               JE=NO. OF WING BOXES
C                                               JS=ORDER OF FIRST WING BOX
C                                               JN=ORDER OF LAST  WING BOX
C                                               JSW=ORDER OF FIRST WAKE BOX
C                                               JNW=ORDER OF LAST  WAKE BOX
      JW=0
      IF(X.LE.AMI) GO TO 34
      J=1
   32 IF(J.GT.JL) GO TO 34
      IF(X.LT.(XTE(NEW,J)+DH)) GO TO 33
      JW=JW+1
   33 J=J+1
      GO TO 32
   34 JE=JL-JW
      IF(MLW(NEW,I)) 36,35,36
   35 JS =JW+1
      JSW=1
      GO TO 37
   36 JS =1
      JSW=JE +1
   37 JN =JS +JE-1
      JNW=JSW+JW-1
      IF (I1.EQ.0) GO TO 50
C     SUBTRACTION OF CONTRIBUTIONS OF PRECEDING ROWS TO UPWASH
      DO 47 J=JS,JN
      DO 45 K=KO,I1
      KL=ML(NEW,K)
```

```
      K1=I+1-K                                                          BOXPO
      IF (K1.EQ.0) GO TO 45                                             BOXPO
      DO 40 N=1,K1                                                      BOXPO
      N1=N+J                                                            BOXPO
      N2=IABS(N-J)+1                                                    BOXPO
      A1=A(1,N1,K1)+A(1,N2,K1)                                          BOXPO
      A2=A(2,N1,K1)+A(2,N2,K1)                                          BOXPO
      WT=1.0                                                            BOXPO
      IF(N.GE.MLC(NEW,2,K).AND.MLC(NEW,2,K).NE.0) WT=AR(N,K,NEW)        BOXPO
      S(1,J,I)=S(1,J,I)-(A1*S(1,N,K)-A2*S(2,N,K))*WT                    BOXPO
   40 S(2,J,I)=S(2,J,I)-(A2*S(1,N,K)+A1*S(2,N,K))*WT                    BOXPO
   45 CONTINUE                                                          BOXPO
   47 CONTINUE                                                          BOXPO
C     SETTING UP MATRIX OF SIMULTANEOUS EQUATIONS                       BOXPO
   50 DO 52 J=1,JE                                                      BOXPO
      DO 52 K=1,JE                                                      BOXPO
      N1=J+K                                                            BOXPO
      N2=IABS(J-K)+1                                                    BOXPO
      WT=1.0                                                            BOXPO
      IF(K.GE.MLC(NEW,2,I).AND.MLC(NEW,2,I).NE.0) WT=AR(K,I,NEW)        BOXPO
C                                                                       BOXPO
      IF(J.EQ.K) WT=1.0                                                 BOXPO
C                                                                       BOXPO
      T(1,J,K)=(A(1,N1,1)+A(1,N2,1))*WT                                 BOXPO
      T(2,J,K)=(A(2,N1,1)+A(2,N2,1))*WT                                 BOXPO
   52 CONTINUE                                                          BOXPO
C                                         SUBTRACTION OF CONTRIBUTION   BOXPO
C                                          FROM WAKE BOXES -- S(WING)=  BOXPO
C                                          S(WING)-T(WAKE)*PHI(WAKE)    BOXPO
      IF(JW.EQ.0) GO TO 60                                              BOXPO
      DO 56 J=JS,JW                                                     BOXPO
      DO 55 N=JSW,JWF                                                   BOXPO
      S(1,J,I)=S(1,J,I)-T(1,J,N)*S(1,N,I)+T(2,J,N)*S(2,N,I)             BOXPO
   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)             BOXPO
   56 CONTINUE                                                          BOXPO
      IF(MLW(NEW,I).LT.0) GO TO 60                                      BOXPO
C                                          RE-POSITION ELEMENTS OF T    BOXPO
      DO 59 N=1,JE                                                      BOXPO
      NW=N+JW                                                           BOXPO
      DO 58 J=1,JE                                                      BOXPO
      MW=J+JW                                                           BOXPO
      DO 58 K=1,2                                                       BOXPO
   58 T(K,J,N)=T(K,MW,NW)                                               BOXPO
   59 CONTINUE                                                          BOXPO
C                                                                       BOXPO
C     SOLUTION OF EQUATIONS                                             BOXPO
   60 CONTINUE                                                          BOXPO
C                                                                       BOXPO
      IF(JWRITE.EQ.0) GO TO 72                                          BOXPO
      WRITE(IW,167) I                                                   BOXPO
      DO 70 K=1,2                                                       BOXPO
      N1=1                                                              BOXPO
      N2=N1+3                                                           BOXPO
   66 DO 67 J=1,JE                                                      BOXPO
   67 WRITE(IW,169) (J,N,T(K,J,N),N=N1,N2)                              BOXPO
      IF(N2.GE.JE) GO TO 68                                             BOXPO
      N1=N2+1                                                           BOXPO
      N2=N1+3                                                           BOXPO
      GO TO 66                                                          BOXPO
   68 CONTINUE                                                          BOXPO
      DO 69 J=1,JE                                                      BOXPO
      WRITE(IW,170) K,J,I,S(K,J,I)                                      BOXPO
   69 CONTINUE                                                          BOXPO
   70 CONTINUE                                                          BOXPO
  167 FORMAT(1H0,5X,I4,17H-TH SPANWISE COLUMN)                          BOXPO
  168 FORMAT(1H1,10X,84HCOEFFICIENT MATRIX (A) OF  (A)*(X)=(8)  FOR VELO BOXPO
     *CITY POTENTIAL (X) ALONG EACH COLUMN//)                           BOXPO
  169 FORMAT(3X,4(2X,1H(,I2,1H,,I2,2H) ,E13.6))                         BOXPO
  170 FORMAT(9X,2HS(,I2,1H,,I2,1H,,I2,2H)=,E13.6)                       BOXPO
   72 CONTINUE                                                          BOXPO
```

```
C
      K = MSIMEC(MC,JE,1,I,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(NEW,JS)-DH)) GO TO 82
      IF(X.GE.(XTE(NEW,JS)+DH)) GO TO 82
      IF(X.GT.XTE(NEW,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(NEW,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(NEW,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 I1=I1+1
      RETURN
   95 WRITE (IW,140)
      STOP
  100 FORMAT(1H1,20X,42HINFLUENCE COEFFICIENT (REAL AND IMAGINARY)///)
  110 FORMAT(1H0,3X,I3,38H- )X SEPARATION IN CHORDWISE DIRECTION)
  120 FORMAT(1H ,5X,4(I4,1P2E13.6))
  140 FORMAT(1H010X,59HSOLUTION OF SIMULTANEOUS EQUATIONS FOR THE POTENT
     1IAL FAILED)
      END
```

```
      SUBROUTINE FORCI(XX,YY,S,SFDX,SFDY,SFDM,KSFD,KSFS,XLE,XTE,[illegible]
     1                 ,AHA,JMAX      ,MLT,L,NEW,M2,NB,NQ,ND,NR,CK,[illegible]
      DIMENSION MLT(2,2,NB),XLE(2,NB),XTE(2,NB),XX(NB),YY(NB),S([illegible]
     1           ,SFDX(NB,ND),SFDY(NB,NQ,1),SFDM(NR,ND,1),KSFD(ND),[illegible]
     2           ,AHA(NB,NB,1),CTE(NB),YMAX(2)
C
      N=0
C     NO SMOOTHING IN SUBROUTINE SMOOTH (SPLN1)
      D=XX(2)-XX(1)
      AM2=1.0
      MZ=KSFD(M2)
      IF(KSFD(M2).EQ.0.AND.NEW.EQ.2) MZ=KSFS
      DO 150 J=1,JMAX
      MLT1=MLT(NEW,1,J)
      MZ=MLT(NEW,2,J)-MLT1         +2
      NQ=MZ+1
      K1=1
      K2=4
      KC=0
      T(1,1)=XTE(NEW,J)
   40 KK=MLT1
      T(1, 2)=XLE(NEW,J)
      T( 1,K2)=0.0
      KA=2
      KB=MZ
      IF(K1.EQ.3.OR.MZ.EQ.2) GO TO 45
      IF(KC.EQ.2) GO TO 45
C     EXCLUDE LEADING EDGE POINT  IF IT IS TOO CLOSE TO  FIRST BOX
      IF(XX(KK)-XLE(NEW,J).GT.0.2*D) GO TO 45
      KA=1
      KB=MZ-1
      KC=1
   45 CONTINUE
      DO 50 K=KA,KB
      T(K, 2)=XX(KK)
      IF(K1.EQ.3) GC TO 50
      T(K,K2)=S(K1,J,KK)
      IF(NEW.EQ.1.OR.K1.EQ.1) GO TO 50
      T(K,12)=AHA(J,KK,NEW)
   50 KK=KK+1
      IF(KC.EQ.2) GC TO 51
      IF(K1.EQ.3) GC TO 60
      CALL SPLN1(1,T(1,1),T(1,8),T(1,9),KB,T(1, 2),T(1,K2),T(1,10),N)
      T(NQ,K2)=T(1,8)
      IF(KC.NE.1) GC TO 51
      KC=2
      GO TO 40
   51 CONTINUE
C
      IF(NZ.GT.2) GC TO 52
C     ADJUSTMENT FOR TRAILING EDGE VELOCITY POTENTIAL
      KK=MLT(NEW,2,J)
      DUM=SQRT((XTE(NEW,J)-XLE(NEW,J))/(XX(KK)-XLE(NEW,J)))
      T(NQ,K2)=DUM*T(NZ,K2)
   52 CONTINUE
C
      KC=0
      IF(K1.EQ.2) GC TO 55
      K1=2
      K2=6
      GO TO 40
   55 K1=3
      GO TO 40
   60 T(NQ, 2)=XTE(NEW,J)
      T(NQ,12)=CTE(J)
      T( 1,12)=1.0
      IF(ABS(XLE(NEW,J)-XX(MLT1)).GT.1.E-05) GO TO 67
C     LEADING EDGE AND FIRST BOX COINCIDE
      NZ=NZ-1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      NQ=NQ-1                                                           FORCI   71
      DO 65 K=2,NQ                                                      FORCI   72
      T(K, 2)=T(K+1, 2)                                                 FORCI   73
      T(K, 4)=T(K+1, 4)                                                 FORCI   74
      T(K, 6)=T(K+1, 6)                                                 FORCI   75
      IF(NEW.EQ.1) GO TO 65                                             FORCI   76
      T(K,12)=T(K+1,12)                                                 FORCI   77
   65 CONTINUE                                                          FORCI   78
   67 CONTINUE                                                          FORCI   79
C                                                                       FORCI   80
C     INTERPOLATE DEFLECTION AT BOX CENTER, LEADING AND TRAILING EDGES  FORCI   81
      CALL SURF2(YY(J),T(1, 2),1,NQ,2,T(1,13),T(1,14),DUM,SFDX(1,M2)    FORCI   82
     $          ,SFDY(1,M2,NEW),SFDH(1,M2,NEW),MZ,2)                    FORCI   83
C     PERFORM CHORDWISE INTEGRATION                                     FORCI   84
      DO 75 K=1,NQ                                                      FORCI   85
      IF(NEW.EQ.2) AM2=T(K,12)*T(K,12)                                  FORCI   86
      T(K,15)=(T(K,14)*T(K,4)+CK*T(K,13)*T(K,6))/AM2                    FORCI   87
   75 T(K,16)=(T(K,14)*T(K,6)-CK*T(K,13)*T(K,4))/AM2                    FORCI   88
      CALL INTGL(T(1,2),T(1,15),T(J,17),T(J,18),T(1,19),2,NQ,NZ,NM)     FORCI   89
      IF(NEW.EQ.2) AM2=CTE(J)*CTE(J)                                    FORCI   90
      DUM=T(NQ,13)/AM2                                                  FORCI   91
      T(J,17)=T(NQ,4)*DUM-T(J,17)                                       FORCI   92
      T(J,18)=T(NQ,6)*DUM-T(J,18)                                       FORCI   93
  150 CONTINUE                                                          FORCI   94
C     PERFORM SPANWISE INTEGRATION                                      FORCI   95
      NQ=JMAK+2                                                         FORCI   96
      K1=17                                                             FORCI   97
      K2=15                                                             FORCI   98
      T(1,1)=0.0                                                        FORCI   99
  165 DO 170 K=1,JMAK                                                   FORCI  100
  170 T(K, 2)=YY(K)                                                     FORCI  101
      NZ=JMAK                                                           FORCI  102
      CALL SPLN1(1,T(1,1),T(1,8),T(1,9),NZ,T(1,2),T(1,K1),T(1,10),N)    FORCI  103
      T(NQ,K2)=0.0                                                      FORCI  104
      T( 1,K2)=T(1,8)                                                   FORCI  105
      DO 175 K=1,JMAK                                                   FORCI  106
  175 T(K+1,K2)=T(K,K1)                                                 FORCI  107
      IF(K1.EQ.18) GO TO 180                                            FORCI  108
      K1=18                                                             FORCI  109
      K2=16                                                             FORCI  110
      GO TO 165                                                         FORCI  111
  180 CONTINUE                                                          FORCI  112
      NZ=JMAK+1                                                         FORCI  113
      T( 1,2)=0.0                                                       FORCI  114
      T(NQ,2)=YMAX(NEW)                                                 FORCI  115
      DO 195 K=1,JMAK                                                   FORCI  116
  195 T(K+1,2)=YY(K)                                                    FORCI  117
C     WING TIP CORRECTION                                               FORCI  118
      K1=0                                                              FORCI  119
      XD=T(NQ,2)-T(NZ-1,2)                                              FORCI  120
      PR=T(NZ-1,15)                                                     FORCI  121
      PI=T(NZ-1,16)                                                     FORCI  122
      IF(T(NZ,2)+.5*D.LT.T(NQ,2)) GO TO 195                             FORCI  123
  190 DUM=SQRT((T(NQ,2)-T(NZ,2))/XD)                                    FORCI  124
      T(NZ,15)=DUM*PR                                                   FORCI  125
      T(NZ,16)=DUM*PI                                                   FORCI  126
      IF(K1.EQ.1) GO TO 196                                             FORCI  127
  195 IF(YY(JMAK)+1.05*D.GE.YMAX(NEW)) GO TO 196                        FORCI  128
      K1=1                                                              FORCI  129
      NZ=NZ+1                                                           FORCI  130
      NQ=NQ+1                                                           FORCI  131
      T(NQ,2)=T(NQ-1,2)                                                 FORCI  132
      T(NZ,2)=T(NZ-1,2)+D                                               FORCI  133
      T(NQ,15)=T(NQ-1,15)                                               FORCI  134
      T(NQ,16)=T(NQ-1,16)                                               FORCI  135
      GO TO 190                                                         FORCI  136
  196 CONTINUE                                                          FORCI  137
      IF(ABS(T(NQ,2)-T(NZ,2)).GE.1.E-05) GO TO 197                      FORCI  138
      NQ=NQ-1                                                           FORCI  139
      NZ=NZ-1                                                           FORCI  140
```

```
      T(NO, 2)=T(NO+1, 2)                                               FORCI     141
      T(NO,15)=T(NO+1,15)                                               FORCI     142
      T(NO,16)=T(NO+1,16)                                               FORCI     143
  192 CONTINUE                                                          FORCI     144
C                                                                       FORCI     145
      CALL INTGL(T(1,2),T(1,15),T(1,1),T(2,1),T(1,19),2,NO,NZ,NM)       FORCI     146
      RETURN                                                            FORCI     147
      END                                                               FORCI     148
```

ORIGINAL PAGE IS
OF POOR QUALITY

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

```
      SUBROUTINE INTGL(X,Y,VK,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
      VK=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)
      DIMENSION X(1),Y(1),D(1)
      DATA JMAX,R6,W/20,.16666667,1.07179677/
      D(1)=0.
      IF(N.EQ.1) RETURN
      D(N)=0.
      AN=N
      DO 1 II=2,N
      I=N+2-II
      X(I)=X(I)-X(I-1)
    1 Y(I)=(Y(I)-Y(I-1))/X(I)
      IF(N.EQ.2) GO TO 5
      DO 2 II=3,N
      I=N+3-II
      D(I-1)=2.*(Y(I)-Y(I-1))/(X(I)+X(I-1))
      Y(I)=1.5*D(I-1)
    2 X(I)=0.5*X(I-1)/(X(I)+X(I-1))
      DO 3 J=1,JMAX
      DO 3 I=3,N
    3 D(I-1)=W*(Y(I)-X(I)*D(I-2)-(0.5-X(I))*D(I))-(W-1.0)*D(I-1)
      DO 4 I=3,N
      X(I)=X(I-1)*(0.5/X(I)-1.)
    4 Y(I)=0.33333333*Y(I)*(X(I)+X(I-1))+Y(I-1)
    5 SAVE=Y(2)-R6*X(2)*(2.*D(1)+D(2))
      DO 6 I=2,N
      C=R6*X(I)
      S1=Y(I)-C*(2.*D(I-1)+D(I))
      S2=Y(I)+C*(D(I-1)+2.*D(I))
      D(I-1)=0.5*(SAVE+S1)
    6 SAVE=S2
      D(N)=SAVE
      IF(IFRMS.NE.1) GO TO 8
      RMS=0.
      IF(N.EQ.2) GO TO 8
      DO 7 I=3,N
      C=2.*D(I-1)
    7 RMS=RMS+((C+D(I-2)-3.*Y(I-1))/X(I-1)+(D(I)+C-3.*Y(I))/X(I))**2
      RMS=2.*SQRT(RMS/AN)
    8 DO 9 I=2,N
      Y(I)=Y(I)*X(I)+Y(I-1)
    9 X(I)=X(I)+X(I-1)
      RETURN
      END
```

```
      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
```

ORIGINAL PAGE IS
OF POOR QUALITY

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

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

ORIGINAL PAGE IS
OF POOR QUALITY

```
      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
```

```
      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(M,N,L,A,B)                                        MSIMEC
      DIMENSION A(M,1),B(M,1)                                           MSIMEC
      COMPLEX A,B,G                                                     MSIMEC
      DO 30 I = 1,N                                                     MSIMEC
      C = 0.0                                                           MSIMEC
      DO 10 J = 1,N                                                     MSIMEC
   10 C=AMAX1(C,ABS(REAL(A(I,J))),ABS(AIMAG(A(I,J))))                   MSIMEC
      IF(C.EQ.0.0) GO TO 1000                                           MSIMEC
      DO 20 J = 1,N                                                     MSIMEC
   20 A(I,J) = A(I,J)/C                                                 MSIMEC
      DO 30 J = 1,L                                                     MSIMEC
   30 B(I,J) = B(I,J)/C                                                 MSIMEC
      IF(N.EQ.1) GO TO 205                                              MSIMEC
      NM = N - 1                                                        MSIMEC
      DO 200 J = 1,NM                                                   MSIMEC
      C = 0.0                                                           MSIMEC
      K = 0                                                             MSIMEC
      DO 40 I = J,N                                                     MSIMEC
      D=ABS(REAL(A(I,J)))+ABS(AIMAG(A(I,J)))                            MSIMEC
      IF(C.GE.D) GO TO 40                                               MSIMEC
      K = I                                                             MSIMEC
      C = D                                                             MSIMEC
   40 CONTINUE                                                          MSIMEC
      IF(K.EQ.0.OR.C.LT.1.E-7) GO TO 1000                               MSIMEC
      IF(K.EQ.J) GO TO 70                                               MSIMEC
      DO 50 JJ = J,N                                                    MSIMEC
      G = A(J,JJ)                                                       MSIMEC
      A(J,JJ) = A(K,JJ)                                                 MSIMEC
   50 A(K,JJ) = G                                                       MSIMEC
      DO 60 JJ = 1,L                                                    MSIMEC
      G = B(J,JJ)                                                       MSIMEC
      B(J,JJ) = B(K,JJ)                                                 MSIMEC
   60 B(K,JJ) = G                                                       MSIMEC
   70 G = 1.0/A(J,J)                                                    MSIMEC
      JP = J + 1                                                        MSIMEC
      DO 80 JJ = JP,N                                                   MSIMEC
   80 A(J,JJ) = A(J,JJ)*G                                               MSIMEC
   90 DO 100 JJ = 1,L                                                   MSIMEC
  100 B(J,JJ) = B(J,JJ)*G                                               MSIMEC
      DO 200 I = 1,N                                                    MSIMEC
      IF(I.EQ.J) GO TO 200                                              MSIMEC
      G = A(I,J)                                                        MSIMEC
      DO 110 JJ = JP,N                                                  MSIMEC
  110 A(I,JJ) = A(I,JJ) - G*A(J,JJ)                                     MSIMEC
      DO 120 JJ = 1,L                                                   MSIMEC
  120 B(I,JJ) = B(I,JJ) - G*B(J,JJ)                                     MSIMEC
  200 CONTINUE                                                          MSIMEC
  205 G = A(N,N)                                                        MSIMEC
      IF (ABS(REAL(G)) + ABS(AIMAG(G)).LT.1.E-7) GO TO 1000             MSIMEC
      DO 210 J = 1,L                                                    MSIMEC
  210 B(N,J) = B(N,J)/G                                                 MSIMEC
      IF(N.EQ.1) GO TO 230                                              MSIMEC
      DO 220 I = 1,NM                                                   MSIMEC
      DO 220 JJ = 1,L                                                   MSIMEC
  220 B(I,JJ) = B(I,JJ) - A(I,N)*B(N,JJ)                                MSIMEC
  230 MSIMEC = 1                                                        MSIMEC
      RETURN                                                            MSIMEC
 1000 MSIMEC = 2                                                        MSIMEC
      RETURN                                                            MSIMEC
      END                                                               MSIMEC
```

1. Report No.
NASA CR-158906

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
IMPROVED SONIC-BOX COMPUTER PROGRAM FOR CALCULATING TRANSONIC AERODYNAMIC LOADS ON OSCILLATING WINGS WITH THICKNESS

5. Report Date
September 1978

6. Performing Organization Code

7. Author(s)
S. Y. Ruo

8. Performing Organization Report No.
LG78ER0225

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 transonic potential flow over an oscillating wing of finite span. The program is based on the original sonic-box computer program for planar wing, which has previously been extended to account for the effect of wing thickness. The latter work is further extended herein to improve computational efficiency and accuracy and to account for swept trailing edges. Account for the non-uniform flow caused by finite thickness is made by application of the local linearization concept with appropriate coordinate transformation. A brief description of each computer routine and the applications of cubic spline and spline-surface data fitting techniques used in the program are given, and the method of input is shown in detail. Sample calculations as well as a complete listing of the computer program listing 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)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	127	$7.25

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