

UNCLASSIFIED

AD NUMBER
AD894188
NEW LIMITATION CHANGE
TO Approved for public release, distribution unlimited
FROM Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; AUG 1971. Other requests shall be referred to Air Force Flight Dynamics Laboratory, ATTN: FY, Wright-Patterson AFB, OH 45433.
AUTHORITY
AFFDL ltr dtd 2 May 1979

THIS PAGE IS UNCLASSIFIED

✓

AFFDL-TR-71-108, PART II

2

**PREDICTION OF UNSTEADY AERODYNAMIC
LOADINGS OF NON-PLANAR WINGS
AND WING-TAIL CONFIGURATIONS
IN SUPERSONIC FLOW**

Part II Computer Program Description

GORDON D. KRAMER

GEORGE E. KEYLON

THE BOEING COMPANY

COMMERCIAL AIRPLANE GROUP

TECHNICAL REPORT AFFDL-TR-71-108, PART II

MARCH 1972

D D C
RECEIVED
MAY 17 1972
C

Distribution limited to U.S. Government agencies only; test and evaluation; statement applied August 1971. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FY), Wright-Patterson AFB, Ohio 45433.

**AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO**

401

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

ACCESSION FOR		
CPSTI	WHITE SECTION	☒
DDC	BUFF SECTION	☒
UNANNOUNCED		☐
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY CODES		
DIST.	AVAIL.	SPECIAL
B		

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

PART 1-AD 89383-1

AFFDL-TR-71-108, PART II

**PREDICTION OF UNSTEADY AERODYNAMIC
LOADINGS OF NON-PLANAR WINGS
AND WING-TAIL CONFIGURATIONS
IN SUPERSONIC FLOW**

Part II Computer Program Description

GORDON D. KRAMER

GEORGE E. KEYLON

Distribution limited to U.S. Government agencies only; test and evaluation; statement applied August 1971. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FY), Wright-Patterson AFB, Ohio 45433.

FOREWORD

Part II of this report was prepared by Boeing Computer Services, Inc., in conjunction with The Boeing Company, Commercial Airplane Group, Renton, Washington for the Aerospace Dynamics Branch, Vehicle Dynamics Division, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, under Air Force Contract AF 33615-70-C-1126, "Unsteady Aerodynamics and Flutter of Wing Horizontal Tail Configurations in Supersonic Flow". This work was conducted under Project No. 1370, "Dynamic Problems in Military Flight Vehicles" and Task No. 137003, "Prevention of Dynamic Aeroelastic Instabilities in Advanced Military Aircraft". Part II of this report is intended to describe in detail the computer program developed under the above contract, and is subsidiary to Part I, Theoretical Development, Program Usage, and Application.

Mr. William S. Rowe of The Boeing Company served as supervisor of the work covered under this contract. The co-authors of this part of the report worked from engineering analyses prepared and documented in Part I by Dr. J. M. Li, C. J. Borland and J. R. Hogley. Mr. Lawrence J. Huttshell of the Aerospace Dynamics Branch (FYS) was Project Engineer. Appreciation is expressed to Mr. H. Huffman and Mr. R. Hirst for graphical work done in this report.

Manuscript was released by the authors in August, 1971 for publication as an AFFDL report.

This report has been reviewed and approved.

Walter J. Mykytow
WALTER J. MYKYTOW
Assistant for Research & Technology
Vehicle Dynamics Division
Air Force Flight Dynamics Laboratory

ABSTRACT

The Mach box technique has been extended to include wing and tail with dihedral angles and vertical separation. A digital computer program, written in FORTRAN, is presented. The program provides for up to nine sweep angles of the leading and trailing edges of each surface. First order piston theory thickness correction is available as an option, and two refinement procedures are provided, subdivision with averaging and velocity potential smoothing. For a maximum of twenty oscillatory mode shapes the program calculates normal washes, velocity potentials, lifts, pressures and generalized forces matrices. If only one surface is being analyzed, sampling of wake up-wash, side-wash and longitudinal wash is available.

TABLE OF CONTENTS

Section	Page
I INTRODUCTION	1
II COMPUTER PROGRAM DESCRIPTIONS	2
1. General Description	2
2. General Purpose Subroutines	4
3. Main Control Program	9
4. Data Input Processor	10
5. Geometry Processor	11
6. Modal Data Processor	27
7. Aerodynamic Influence Coefficients Section	34
8. Normal-washes and Velocity Potentials	44
9. Velocity Potential Smoothing Sections	58
10. Generalized Air Forces Section	61
11. Common Block Organization	64
12. Array Storage	71
13. Internal Scratch Files	75
14. Output Files	81
15. Implementation and Debugging	84
III PROGRAM FLOWCHARTS	86
CONTROL	86
DATAPP	87
GEOMEX	88
MODES	89
VICMAIN	90
NWVLPT	91
SMTH	92
CHORDF	93
FORCES	94
REFERENCES	95
APPENDIX A: Sample Input and Output Data	A1
APPENDIX B: Program Listings	B1

LIST OF ILLUSTRATIONS

Figure		Page
1.	Coordinate Systems for a Right Wing	xxii
2.	Coordinate Systems for a Right Tail	xxiv
3.	Program Overlay Structure	3
4.	Output from BXCDPF, Called for a Wing	15
5.	Output from BXCDI, Called for a Coplanar Wing and Tail	18
6.	Possible Arrangement of Sending Boxes, Left Surface to Right	21
7.	MUAIC Array Generated by PWWAIC for Figure 4	21
8.	AIC Integration, Full Box	35
9.	AIC Integration, Edge Boxes	35
10.	AIC Integration, Apex Boxes	36
11.	Tip Chord Trailing Edge Velocity Potential Calculation	50
12.	Subdivided "Effective Area"	53
13.	Unsubdivided Boxes outside the "Effective Area"	54
14.	Box Code Arrays	71
15.	Leading and Trailing Edge Arrays	71
16.	Planform Edge Definitions	72
17.	AIC Array Pointers	72
18.	Row Pointers	73
19.	Planar AIC	73
20.	Spatial AICs	74
21.	Tape Storage of AIC Arrays	82
22.	Tape Map of Forces Output Tape	83
23.	Sample Problem Configuration	A1

NOMENCLATURE

Mathematical Symbol	FORTTRAN Symbol	Dimension	Definition
a		length/time	Speed of Sound = U/M
$A(n,m)$		area	Integration area of box n,m
b_1	B1	length	Chordwise dimension of Mach box.
b_{1S}	B1S	length	b_1/N_S = chordwise dimension of a subdivided box.
$C_{\bar{\rho}\bar{\mu}\bar{\lambda}}$	C	non-dimensional	Velocity potential spatial aerodynamic influence coefficient (AIC).
$C_{\bar{\rho}\bar{\mu}\bar{\lambda}}^{(xy)}$	C	non-dimensional	Spatial AIC giving velocity potential at a point on surface "xy" due to constant outward normal wash over a box on surface "ab"; possible values for superscripts are $x \text{ or } a \begin{cases} \text{R-right} \\ \text{L-left} \end{cases} \quad y \text{ or } b \begin{cases} \text{W-wing} \\ \text{T-tail} \end{cases}$
$C_{\bar{\rho}\bar{\mu}\bar{o}}$ $C_{\bar{\rho}\bar{\mu}}$	PKERNL	non-dimensional	Velocity potential planar AIC
$C_{\bar{\rho}\bar{\mu}\bar{o}}^s$	SKERNL	non-dimensional	Planar AIC defined for subdivided sending boxes.
$\Delta \bar{C}_{r_j}^{n,m}$	DELCP	1/length	Pressure coefficient difference at box n,m for the j^{th} mode (program output)
l_c, c	--	length	Local reference chord

NOMENCLATURE

Mathematical Symbol	FORTRAN Symbol	Dimension	Definition
$C_{l_j}^m$	--	1/length	Local lift coefficient per unit span for the j^{th} mode
$C_{m_j}^m$	SECMOM	$1/(\text{length})^2$	Local moment coefficient per unit span for the j^{th} mode
$\left. \begin{matrix} c_{rW} \\ c_{rT} \end{matrix} \right\}$	---	length	$\left\{ \begin{matrix} \text{Wing} \\ \text{Tail} \end{matrix} \right\}$ root section chord length.
$\frac{D}{Dt}$	--	1/time	Substantial derivative; $\frac{D}{Dt} = \frac{\partial}{\partial t} + U \frac{\partial}{\partial x}$
$f_j(x,y)$	DEFSL(1,L)	non-dimensional	j^{th} mode shape deflection at location (X,Y).
f_{ij}	---	non-dimensional	Deflection of i^{th} lumped mass in mode j
$\frac{\partial f_j(x,y)}{\partial x}$	DEFSL(2,L)	1/length	Slope of j^{th} mode shape function.
$\bar{f}_j^{n,m}$	---	time	Scaled modal displacement at box n,m $\bar{f}_j^{n,m} = \frac{b_1}{U} f_j^{n,m}$
J	---	1/length	j^{th} mode shape deflection / s
i	---	mass x length ²	Moment of inertia about the elastic axis of the i^{th} lumped mass
1_j	---	force/length	Generalized stiffness
β	XKS	Non-dimensional	Reduced frequency based on leading planform semi-span, $k_R = \frac{\omega b}{U}$

NOMENCLATURE

Mathe- matical Symbol	FORTRAN Symbol	Dimension	Definition
k_1	XKI,K1	non- dimensional	Reduced frequency based on the chordwise dimension of the Mach box $k_1 = \frac{\omega b_1}{U}$
$\bar{k}_1$	K1BAR	non- dimensional	$k_1 M^2 / \beta^2$
L	CAPL	non- dimensional	Vertical separation of the center lines of the 2 surfaces, positive upward measuring from the wing to the tail.
$L_j^{n,m}$	--	force/length	lift on box n,m for the j^{th} mode
L_j^m	--	force/length	lift on the m^{th} chordwise strip of boxes for the j^{th} mode.
L_j	--	force/length	lift on a complete half-surface or half-airplane for the j^{th} mode.
$\bar{L}_j^{n,m}$	--	force/length	amplitude of box lift $L_j^{n,m}$
$\bar{L}_j^m$	--	force/length	amplitude of section lift L_j^m
$\bar{L}_j$	--	force/length	amplitude of total lift L_j
$\bar{L}_j^{n,m}$	BXLIFT	non- dimensional	Non-dimensional amplitude of box lift (program output) $\bar{L}_j^{n,m} = L_j^{n,m} e^{i\omega t} = \frac{b_1}{\beta} \bar{L}_j^{n,m} e^{i\omega t}$

NOMENCLATURE

Mathematical Symbol	FORTTRAN Symbol	Dimension	Definition
$\bar{L}_j^m$	SLIFT	non-dimensional	nondimensional amplitude of section lift (program output) $L_j^n = \bar{L}_j^n e^{i\omega t} = q \left(\frac{b_1}{\beta} \right) \bar{L}_j^n e^{i\omega t}$
$\bar{L}_j$	TLIFT	non-dimensional	Nondimensional amplitude of total lift (program output) $L_j = \bar{L}_j e^{i\omega t} = q \left(\frac{b_1}{\beta} \right) \bar{L}_j e^{i\omega t}$
l	EL	non-dimensional	The l_c coordinate location of a pulse receiving point, i.e., the perpendicular distance from the sending plane to receiving point.
M	XMACH	non-dimensional	Mach number
M_{jj}	--	mass	Generalized mass for the j^{th} mode
m	--	non-dimensional	The m_c coordinate location of a pulse receiving point.
m_i	--	mass	i^{th} lumped mass
$N_{n,m}^{xyz}$	ENRUS ENRLS, EN, ENSUBD	non-dimensional	Normal wash at box n,m on surface "xyz" due to local source strength, where possible subscript values are; $x = \begin{cases} R\text{-right} \\ L\text{-left} \end{cases}$ $y = \begin{cases} U\text{-upper} \\ L\text{-lower} \end{cases}$ $z = \begin{cases} W\text{-wing} \\ T\text{-tail} \end{cases}$ e.g. $N_{RUW}^{n,m}$ means normal wash on the right upper wing at box n,m

NOMENCLATURE

Mathe- matical Symbol	FORTRAN Symbol	Dimension	Definition
$\hat{N}^{n,m}$ xyz abc	ENRULU, ENRLLL, ENRURW, ENRULW	non- dimensional	Normal wash at box n,m on surface "xyz" due to remote source strengths on surface "abc", where possible subscript values for a,b, and c are the same as for x,y, and z, respectively, defined above; e.g. $\hat{N}^{n,m}$ means normal RUT LLW wash at box n,m on the right upper tail due to source strengths on the left lower wing.
N_s	NSUBDV	non- dimensional, odd integer	No. of "sub-boxes" (chordwise and spanwise) to be used in the subdivision improvement technique.
n	--	non- dimensional	the n_c coordinate location of a pulse receiving point.
n_c, m_c, l_c	---	---	Sending Surface Coordinate System
$\bar{n}_c, \bar{m}_c, \bar{l}_c$	---	---	Receiving Point Coordinate System
$p, p(x,y,t)$	---	force/area	local static pressure
p_∞	---	force/area	Free stream static pressure
$\Delta p(x,y,t)$	---	force/area	pressure difference between upper and lower surfaces at point (x,y) at time t $\Delta p(x,y,t) = p(x,y,t)_{upper} - p(x,y,t)_{lower}$

NOMENCLATURE

Mathe- matical Symbol	FORTTRAN Symbol	Dimension	Definition
$\Delta \bar{p}(x,y)$	--	force/area	Amplitude of pressure difference: $\Delta p(x,y,t) = \Delta \bar{p}(x,y) e^{i\omega t}$
q_{ij}		force/length	Generalized force due to the deformation in the i^{th} elastic mode and loading for the j^{th} modal deflections
$\bar{q}_{ij}$	--	force/length	Amplitude of generalized force
$\bar{\bar{q}}_{ij}$	GENAF	non- dimensional	Non-dimensional generalized force (program output); $q_{ij} = \bar{q}_{ij} e^{i\omega t} = q \frac{b_i}{\beta} \bar{\bar{q}}_{ij} e^{i\omega t}$
$\hat{q}_{ij}$	QAGARD	$1/(\text{length})^2$	Generalized force in the AGARD notation
q'_{ij}, q''_{ij}	GENAFC	$1/(\text{length})^2$	Real and imaginary parts of $\hat{q}_{ij}$ in the AGARD definition (program output)
q	--	force/area	dynamic pressure
$q_j(t)$	--	length	Generalized coordinate relating physical deflection to j^{th} modal deflections: $z(x,y,t) = \sum_j f_j(x,y) q_j(t)$
$\bar{q}_j$	--	length	Amplitude of j^{th} generalized coordinate
s	S	length	Wing semi-span.

NOMENCLATURE

Mathe- matical Symbol	FORTTRAN Symbol	Dimension	Definition
S	--	length ²	Area of integration. Bounded by edge of planform plus diaphragm and lying inside the forward Mach cone of the receiving point.
t	--	time	Time
U	--	length/ time	Free stream velocity.
$U_{\overline{\mu}\lambda}$	---	non- dimensional	Velocity spatial AIC for perturbation velocity parallel to the free stream.
u	---	length/ time	Perturbation velocity in the stream-wise direction, positive downstream.
$V_{\overline{\mu}\lambda}$	V	non- dimensional	Velocity spatial aerodynamic influence coefficient (AIC) for velocity normal to the free stream and parallel to the sending surface.
$\begin{matrix} (xy) \\ v_{ab} \\ \overline{\mu}\lambda \end{matrix}$	V	non- dimensional	Spatial AIC giving velocity normal to the free stream and parallel to surface "ab" at a point on surface "xy" or in the flowfield (FF), due to constant normal wash over a box on surface "ab"
v		length/ time	Perturbation velocity in the span-wise direction, positive right (looking upstream).
$W_{\overline{\mu}\lambda}$	W	non- dimensional	Velocity spatial aerodynamic influence coefficient (AIC) for velocity normal to the sending plane.

NOMENCLATURE

Mathematical Symbol	FORTTRAN Symbol	Dimension	Definition
$\frac{(xy)_{ab}}{\nu/\mu\lambda}$	W	non-dimensional	Spatial AIC giving velocity normal to surface "ab" at a point on surface "xy" or in the flow field (FF) due to a constant normal wash over a box on surface "ab".
w		length/time	Perturbation velocity in the vertical direction, positive upward.
X,Y,Z		length	Reference (global) coordinate system, X positive aft, Y positive right, Z positive upward.

NOMENCLATURE

Mathematical Symbol	FORTTRAN Symbol	Dimension	Definition
X_w, Y_w, Z_w			Wing local coordinate system.
X_T, Y_T, Z_T			Tail local coordinate system
X_{WLE}	XWLE	length	The location of a leading edge definition point of the wing planform, measured along the X_w co-ordinate.
X_{TLE}	XTLE	length	Same as above for the tail planform, measured along the X_T coordinate.
X_{WTE}	XWTE	length	The location of a trailing edge definition point of the wing planform.
X_{TTE}	XTTE	length	Same as above for the tail, measured along the X_T co-ordinate.
$\begin{Bmatrix} X_F \\ X_C \end{Bmatrix}$	$\begin{Bmatrix} XEDGE \\ XCENR \end{Bmatrix}$	length	Location of the $\begin{Bmatrix} \text{edge} \\ \text{center} \end{Bmatrix}$ of a Mach box used for the placement of the box pattern, measured along the X_w coordinate.
$\begin{Bmatrix} X_{CW} \\ X_{CT} \end{Bmatrix}$			The location of the most upstream row of boxes on the $\begin{Bmatrix} \text{wing} \\ \text{tail} \end{Bmatrix}$ measured along the $\begin{Bmatrix} X_w \\ X_T \end{Bmatrix}$ co-ordinate.

NOMENCLATURE

Mathematical Symbol	FORTRAN Symbol	Dimension	Definition
$\begin{Bmatrix} Y_{WLE} \\ Y_{TLE} \end{Bmatrix}$	$\begin{Bmatrix} YWLE \\ YTLE \end{Bmatrix}$	length	The location of a leading edge definition point of the $\begin{Bmatrix} \text{wing} \\ \text{tail} \end{Bmatrix}$ planform measured along coordinate $\begin{Bmatrix} Y_W \\ Y_T \end{Bmatrix}$.
$\bar{y}$	YBAR	non-dimensional	Offset of receiving chord from the nearest sending chord.
$\begin{Bmatrix} Z_U \\ Z_L \end{Bmatrix}(x,y,t) --$		length	$\begin{Bmatrix} \text{Upper} \\ \text{Lower} \end{Bmatrix}$ surface deflection at (x,y) as a function of time
$Z_m(x,y,t) --$		length	mean surface deflection: $Z_M(x,y,t) = f_j(x,y)e^{i\omega t}$
$Z_\tau(x,y) --$		length	Local thickness at (x,y)
$\frac{\partial Z_\tau}{\partial x}(x,y)$	TSLFN	non-dimensional	Local thickness slope at (x,y)
$\bar{Z}_\tau$	--	non-dimensional	Thickness slope piston theory correction; $\bar{Z}_\tau = 1 + \frac{\gamma+1}{2} M \frac{\partial Z_\tau}{\partial x}$
$\alpha_{n,m}$	ALPHA	non-dimensional	Edge box area ratio for box(n,m).
β	BETA	non-dimensional	$\sqrt{M^2 - 1}$

NOMENCLATURE

Mathe- matical Symbol	FORTTRAN Symbol	Dimension	Definition
b_1/β	BIBETA	length	Spanwise dimension of the Mach box.
b_{1s}/β	BIBTAS	length	Spanwise dimension of a sub-divided Mach box.
γ	GAMMA	non-dimensional	Ratio of specific heats, = 1.4
ζ	ZETA	non-dimensional	Dummy variable in the Z_w or Z_T coordinate.
$\bar{\eta}$	ETABAR	non-dimensional	Dummy variable of integration in the $\bar{m}_c$ coordinate.
θ	THETA	radians	$\sin^{-1} \frac{\bar{\eta}}{\tau} = \sin^{-1} \frac{\bar{\eta}}{\sqrt{\bar{\eta}^2 - \bar{\lambda}^2}}$
θ_{ij}	--	radians/ length	Torsion of i^{th} lumped mass in mode j .
$\bar{\lambda}$	--	non-dimensional	$\bar{l}_c$ coordinate location of a pulse sending box.
μ	MU, YMU	non-dimensional	m_c coordinate location of a pulse sending box.
$\bar{\mu}$	YMUBAR	non-dimensional	$\bar{m}_c$ coordinate location of a pulse sending box.

NOMENCLATURE

Mathematical Symbol	FORTTRAN Symbol	Dimension	Definition
ν	NU, XNU	non-dimensional	n_c coordinate location of a pulse sending box.
$\bar{\nu}$	XNUBAR	non-dimensional	$\bar{n}_c$ coordinate location of a pulse sending box.
ξ	XI	non-dimensional	Dummy variable of integration in the X_w or X_T coordinate.
$\bar{\xi}$	XIBAR	non-dimensional	Dummy variable of integration in the $\bar{n}_c$ coordinate.
ρ, ρ_∞	--	Mass/volume	Free stream density.
τ	TAU	non-dimensional	$\sqrt{\bar{\xi}^2 - \bar{\lambda}^2}$
$\phi(x, y, t)$	--	length ² /time	Disturbance velocity potential at point (x,y) and time t, defined so that $\frac{\partial \phi}{\partial x_1}$ is velocity, positive in positive x_1 direction, where $x_1 = X, Y, \text{ or } Z$
$\Delta \phi(x, y, t)$	--	length ² /time	Disturbance velocity potential difference between the top and bottom side of the surface at point (x,y) and time t: $\Delta \phi(x, y, t) = \phi_{\text{upper}} - \phi_{\text{lower}}$

NOMENCLATURE

Mathematical Symbol	FORTRAN Symbol	Dimension	Definition
$\Delta \bar{\phi}_j(x,y)$	--	length / time	Amplitude of velocity potential difference at point (x,y) or for box n,m for the j th mode
$\Delta \bar{\phi}_j(x,y)$ $\Delta \bar{\phi}_{j,n,m}$	DELPHI	non-dimensional	Non-dimensional velocity potential difference due to the unit j th generalized coordinate (program output); $\Delta \phi_j^{n,m} = \bar{\Delta \phi}_j^{n,m} e^{i\omega t} = \frac{U}{b_1} \left(\frac{b_1}{\beta} \right) \bar{\Delta \phi}_j^{n,m} e^{i\omega t}$
$\Delta \bar{\phi}_j(x_{TE},y)$ $\Delta \bar{\phi}_{j,TE}^m$	TVP	non-dimensional	Trailing edge velocity potential difference.
$\left\{ \begin{matrix} \psi_w \\ \psi_T \end{matrix} \right\}$	$\left\{ \begin{matrix} \text{PSIW} \\ \text{PSIT} \end{matrix} \right\}$	degrees	Dihedral angle of $\left\{ \begin{matrix} \text{wing} \\ \text{tail} \end{matrix} \right\}$, radians, positive upwards from horizontal.
ω	---	radians/time	Circular frequency
ω_j	---	radians/time	Circular frequency of mode j

Superscripts

$\left. \begin{matrix} (n,m) \\ (\nu,\mu) \end{matrix} \right\}$ Box location

Subscripts

L Lower limit of Integration; Left-hand surface; Lower surface
R Right-hand Surface
S Subdivided
T Tail
U Upper limit of Integration; Upper Surface
W Wing
FF Flowfield

GLOSSARY OF TERMINOLOGY

Aftmost Box - Each chord on each planform and diaphragm combination has one such box. It is the aftmost box on that chord for which AIC arrays must be calculated and may be on the planform or diaphragm.

AIC - Aerodynamic Influence Coefficient

Area Ratio - On-planform fraction of a box which is cut by the planform boundary.

Apex Box - The box on the sending surface which encloses the apex of the Mach hyperbola associated with the receiving box.

Box Grid - Non-dimensionalized geometric array of boxes whose extent is determined by the geometric properties of the planforms. The term "grid" embraces the arrays on both surfaces.

Control Point - The location at which a receiving box is deemed to be influenced by other boxes. In general, the center of the receiving box.

Effective Area - A concept which relates entirely to the sub-division technique. It is composed of those boxes sufficiently close to the receiving box that their influence on it is large enough for the subdivision refinement to affect results significantly. The size has been arbitrarily set to include the N_{BOX}/N_S rows immediately ahead of the receiving box.

Global Co-ordinate System - An overall reference system of co-ordinates. For example, the airplane co-ordinate system $X \sim$ aft, $Y \sim$ right, $Z \sim$ up. $Y = 0$ at centerline of airplane.

Leading Edge Diaphragm - All diaphragms on which $\Delta\phi = 0$.

Local Co-ordinate System - A co-ordinate system lying in the plane of the surface. $x \sim$ aft, $y \sim$ root to right tip. $y = 0$ at center line of airplane.

Longitudinal Separation - Streamwise distance between the trailing edge of the wing and the leading edge of the tail, measured along the centerline.

GLOSSARY OF TERMINOLOGY

Mach Asymptote - The asymptote of the Mach hyperbola.

Mach Hyperbola - The intersection of the sending plane and the forward Mach cone of the receiving point. Since this is always non-dimensionalized, it is a rectangular hyperbola.

Map - A condensed description of a large amount of data which can be used to locate any desired data element. A map of a banded sparse matrix might consist of two numbers per row, the first being the first non-zero column of that row and the second being the band width for that row. The matrix itself could then be stored as band elements only.

Normal Offset - The l_c distance between the sending box and the receiving point.

Parallel Offset - The m_c distance between the sending box center and the receiving point.

Partial Box - A sending box which is cut by the Mach hyperbola but which is neither an apex box nor an edge box.

Planar A.I.C. - An A.I.C defined by the geometric relation between a sending box and receiving box which lies in the same plane. $C_{\bar{\nu}, \bar{\mu}, 0}$ only.

Receiving Box - In defining the relationship between two boxes the receiving box is the box which can be influenced by the other box.

Receiving Chord - Those receiving boxes which lie on the same chord. The receiving chord is significant in that all the boxes lying on it use AIC arrays which are a subset of those for the aftmost box lying on that chord.

Sending Box - In defining the relationship between two boxes, the sending box is the box which influences the other box (c.f. Receiving Box).

Spatial A.I.C. - An AIC defined by the geometrical relationship between two boxes which do not lie in the same plane. $C_{\bar{\nu}, \bar{\mu}, \bar{\lambda}}, V_{\bar{\nu}, \bar{\mu}, \bar{\lambda}}, W_{\bar{\nu}, \bar{\mu}, \bar{\lambda}}$.

GLOSSARY OF TERMINOLOGY

Sub-box - A member of the array of boxes formed when the grid of sensing boxes is subdivided. Note it refers to the small box which is a fraction of the large box, and not to a large box which has been subdivided.

Tail - The downstream surface.

Vertical Separation - The vertical distance between the center lines of the two surfaces. Positive if the second surface is above the first.

Wake Diaphragm - That part of the diaphragm where $\Delta P = 0$ due to the influence of a surface.

Wing - Upstream Surface - (E.g. a Canard could be referred to as a wing);

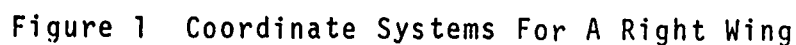
xxii

Figure 1 (Cont'd)

Symbol	Transformation	Definition	Dimension
n_c m_c l_c	$-(n_c - n)$ $-(m_c - m)$ $-(l_c - l)$	Receiving Point Coordinate System parallel to the n_c, m_c, l_c coordinates but opposite in sign and having their origin at the pulse receiving point (n, m, l in the n_c, m_c, l_c coordinates)	non-dimen sional

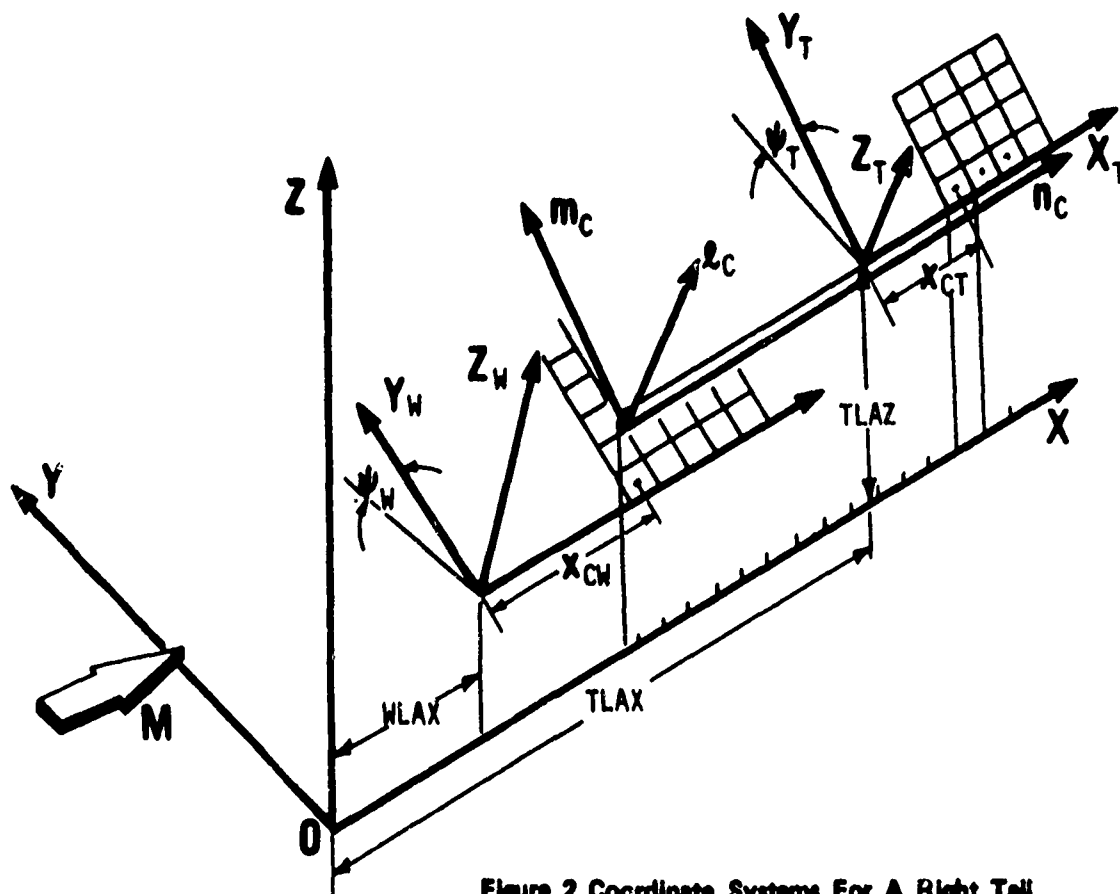


Figure 2 Coordinate Systems For A Right Tail

SYMBOL	TRANSFORMATION	DEFINITION	DIMENSION
X_T Y_T Z_T	$X - TLAX$ $Y \cos \psi_T + (Z - TLAZ) \sin \psi_T$ $(Z - TLAZ) \cos \psi_T - Y \sin \psi_T$	Tail Local Coordinate System used to define tail leading and trailing edges.	length
n_c m_c ρ_c	$\{X_T + TLAX - (WLAX + X_C)\} / b_1 + 1$ $Y_T / (b_1 / \beta) + 1/2$ $Z_T / (b_1 / \beta)$	Sending Surface Coordinate System. In this case the right tail is shown as the sending surface.	non-dimensional

SECTION I

INTRODUCTION

Part II of this report describes the computer program written according to the analysis of Part I. Part II refers implicitly to Part I, Section III, Computer Program Usage, and material covered there is not repeated here. The program computes generalized unsteady air forces on a wing or wing and tail in supersonic flow, given geometric details of the surfaces and the oscillatory mode shapes of the surfaces. The surfaces may be coplanar, may have dihedral angles, and may be separated vertically. The Mach box technique may be used "straight", or three refinements may be applied: 1) Sub-division of the Mach boxes to improve velocity potentials, 2) Least-squares smoothing of calculated velocity potentials to eliminate roughness due to box representation of surface edges, 3) Piston theory correction for airfoil thickness. The refinements may be applied in any combination. As intermediate results, normal-washes, velocity potentials and wake sampling of upwash, sidewash and longitudinal washes may be printed, all at box center locations. The box lifts, pressure distribution, section lifts and total lift are also available for each mode.

SECTION II

COMPUTER PROGRAM DESCRIPTIONS

1. GENERAL DESCRIPTION

The computer program consists of a main (0,0) overlay, one primary level overlay, and eight secondary level overlays (see fig. (3)). The main (0,0) overlay is a general purpose driver, and easily can be used to incorporate other compatible programs with this one as a complete flutter system. Its sole function is to set up buffers and any other system oriented parameters, then call the primary level overlay, CONTROL.

Overlay CONTROL contains the basic logic of the program. It first calls secondary overlay DATAPP, which reads and processes the basic card data necessary for execution of the program. The resulting parameters are stored in labelled common blocks, accessible to all other overlays.

If PRVGEOM = .FALSE., overlay GEOMBX is next called. The planform geometry is read and processed to yield a disk file IGEOSC containing all internally necessary geometric parameters.

If PRVMODE = .FALSE., overlay CONTROL next calls overlay MODES. This area processes the three forms of modal data and places the results, evaluated at box centers, on scratch file MODESC.

Overlay CONTROL next enters a loop on reduced frequency. Each pass through the loop first executes overlay VICMAIN, which computes (or reads from previously saved tapes) all AIC arrays needed at the current reduced frequency. Next overlay NWVPMBX is called, to compute normal-washes, velocity potentials, and optional sample washes. If SMOOTH = .TRUE. overlay SMTH is called to do a least-squares surface fit of the resulting $\Delta\phi$ arrays. If CRDFIT = .TRUE. overlay CHORDF is called to smooth the $\Delta\phi$ values a chord at a time. The final overlay, FORCES, then computes box lifts, section lifts and generalized forces for any smoothed $\Delta\phi$ values first, then for the unsmoothed values. The desired results are printed as they are computed. The loop on reduced frequencies terminates at this point.

Overlay CONTROL reads the termination card which causes a transfer back to the execution of DATAPP (Recycle), the call of another overlay (if available), RETURN to the main (0,0) overlay, or EXIT to control cards.

The following sections give a more detailed description of all of the overlay main programs, and the major subroutines called by each.

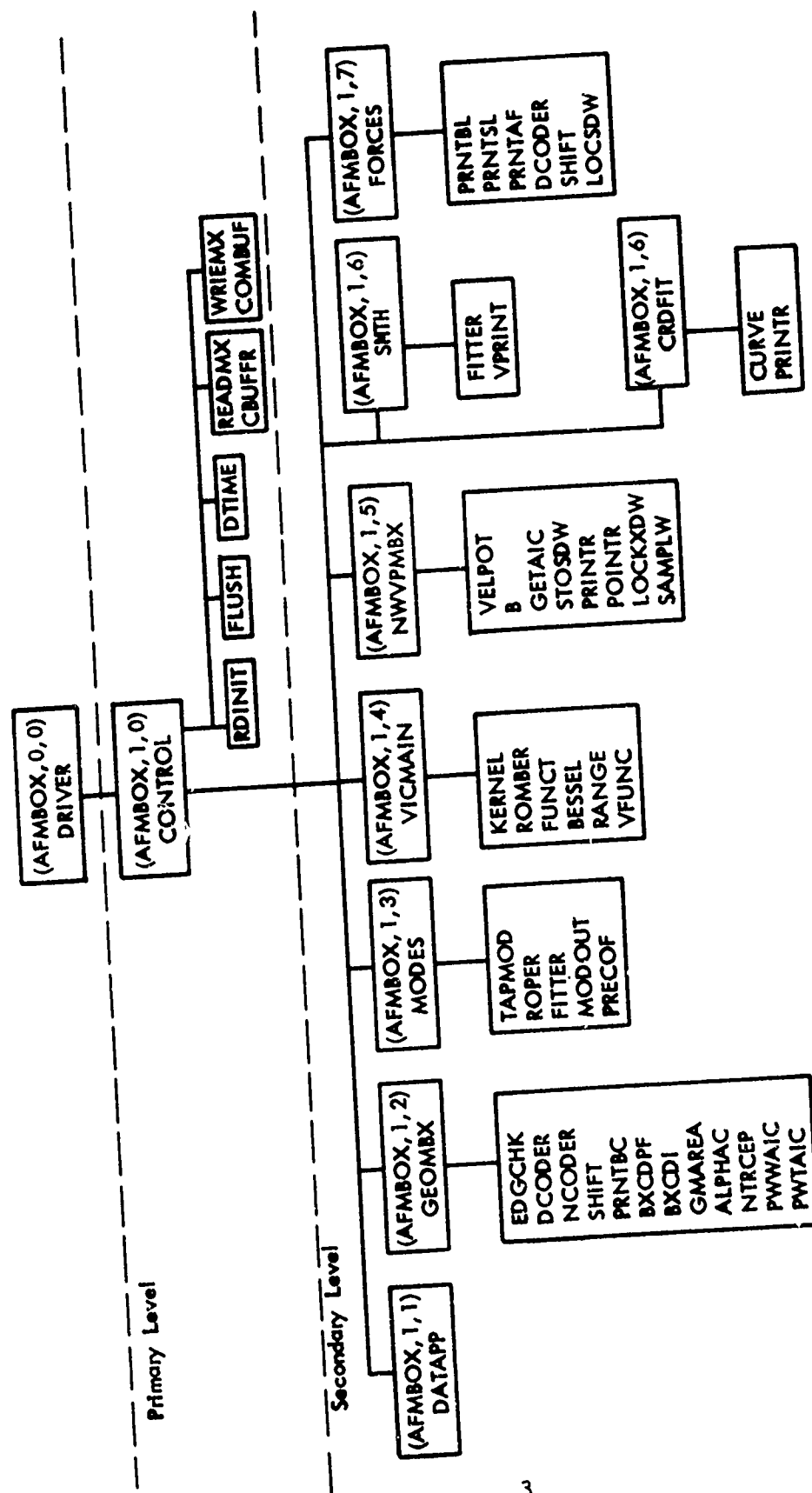


Figure 3 Program Overlay Structure

1. GENERAL PURPOSE SUBROUTINES

Fortran Callable COMPASS Function SHIFT

Author: G. E. Keylon

Purpose: To shift the contents of a word left or right a specified number of bits, identical to the Fortran Extended capability.

Method: The word and the number of bits to be shifted are stored in machine registers. The word is then left circular shifted the number specified. This causes the word to be shifted left circular if the number is positive and right with sign extension if the number is negative. The result is left in register X6 so that this routine must be used as a function subprogram.

Usage: INTEGER SHIFT

.
.
.
IWORD = SHIFT (NWORD,N)

Input

NWORD - The word to be shifted
N - The number to shift the word
If N is positive shift left circular.
If N is negative shift right with sign extension.

Output

IWORD has the results of the shift on NWORD.

Fortran Subprogram WRTEMX

Author: G. E. Keylon

Purpose: To write a matrix on a tape or disk file.

Method: The matrix is placed row-wise into a buffer in labelled common RWBUFF with all of the unused areas of its array omitted. The buffer is then written onto the specified tape or disk file with the Fortran BUFFER OUT statement. A 16 word header record is written in the same manner before each matrix. The header record contains matrix size, name and optional parameters.

Usage: CALL WRTEMX (IOUTFL, MXWRIT, RANDOU, NFS, NMS, LS, NMR, LWS, K, ID, A, ITYPE, M, N, PARM, IRR)

Input

IOUTFL - Tape number or left-justified file name.
MXWRIT - Logical variable, not used.
RANDOU - .T. Random File (not used)
 .F. Sequential File
NFS - Number of files to space before writing
NMS - Number of matrices to space before writing
LS - Level number to space (not used)
NMR - Name or number in random index (not used)
LWS - Level number of this matrix (not used)
K - Row dimension of array A.
 if K ≤ matrix is already in /RWBUFF/
ID - Array containing matrix nome.
A - Array containing matrix
ITYPE - TYPE of matrix (i.e., real, complex, integer, null, mixed)
M - Row dimension of matrix
N - Column dimension of matrix
PARM - 10-word parameter array

Output

IRR - Error return
 0, no error
 1, matrix spacing is negative
 2, File spacing is negative
 4, M*N dimensions greater than buffer size
 1500+I, encountered EOF after matrix I while skipping matrices.

Fortran Subprogram COMBUF

Author: G. E. Keylon

Purpose: To put a complex matrix into a buffer prior to writing on file.

Method: The matrix is placed row-wise into a buffer with all of the unused areas of the matrix omitted. All of the real parts are stored in the first part of the buffer then all of the imaginary parts are placed immediately following the real.

Usage: CALL COMBUF (A, K2, M, N, BUFF)

Input:

A - Array that contains matrix, typed complex
K2 - 2x (row dimension of A)
M - Number of rows in matrix (not array size)
N - Number of columns in matrix (not array size)

Output:

BUFF - Buffer that will contain matrix

Fortran Subprogram READMX

Author: G. E. Keylon

Purpose: To read a matrix from tape or disk file.

Method: A 16-word header record and a matrix record are read from the specified file with BUFFERIN statements. The 16-word header record contains matrix size, name and optional parameters. The matrix is then placed in a given array in correct Fortran storage.

Usage: CALL READMX(INFILE, MXREAD, RANDIN, NFS, NMS, LS, NMR, K, NID, ID, ITYPE, LRS, A, M, N, PARM, IRR)

Input:

INFILE - Tape number or left justified alphanumeric file name

MXREAD - Logical variable (not used)

RANDIN - .T. Random File (not used)
.F. Sequential File

NFS - Number of files to space before reading

NMS - Number of matrices to space before reading

LS - Level number to space (not used)

NMR - Random name or number (not used)

K - Row dimension of array A

If $K \leq 0$ matrix will be left in /RWBUFF/

NID - Number of words available in ID array

In/Out:

ID - Identification array

ITYPE - Real, diagonal, null, mixed, complex

Output:

LRS - Level number of matrix read (not used)

A - Array containing matrix

M - Row dimension of matrix

N - Column dimension of matrix

PARM - Array of numerical parameters stored with the matrix in the 16 word header record

IRR - Error return

0, no error

1, matrix spacing is negative

2, file spacing is negative

4, matrix dimensions illegal

5, $M > K$

1500 + I, encountered EOF after matrix I while skipping matrices.

Fortran Subprogram CBUFFER

Author: G. E. Keylon

Purpose: To move a complex matrix from a buffer to a Fortran array.

Method: The matrix assumed stored row-wise in the buffer with all of the real parts followed by all of the imaginary parts. The conversion leaves the matrix in the array in typical Fortran storage.

Usage: CALL CBUFFER(A,K2,M,N,BUFF)

Input:

K2 - 2x (row dimension of array A)
M - Number of rows in matrix (not array size)
N - Number of columns in matrix (not array size)
BUFF - Buffer that contains matrix

Output:

A - Array that will contain matrix in complex storage

3. MAIN CONTROL PROGRAM

Fortran Program CONTROL

Author: G. E. Keylon, G. D. Kramer

Purpose: To control the flow of the program to the various lower level overlay section.

Method: The program has all of the labeled common blocks so that information can be passed from lower levels to this program which will determine the program flow.

Usage: The CONTROL Program is a main routine. It is the only primary overlay section in the program. It calls all of the lower level or secondary overlay sections. It is called from the initial or main overlay section as follows:

```
CALL OVERLAY(6HAFMBOX, 1, 0, 0)
```

Common Input and Output:

This program does not input or create common values. It is the means by which common values are passed between the secondary overlays of the program.

4. DATA INPUT PROCESSOR

Fortran Program DATAPP

Author: G. E. Keylon

Purpose: To read most of the input data and set flags and options for use throughout the program. It prints the title and options for each run.

Method: The title and all the input options are read in. The heading is printed. The options are read under a NAMELIST format and flags set to default options unless read in.

Usage: The DATAPP program is the main program of a secondary overlay of the Mach Box program. It is called as an overlay section as follows:

```
CALL OVERLAY(6HAFMBOX, 1, 1, 0)
```

All input and output is through labeled common blocks.

Common Input:

```
PREVEX  
OMACH  
DEFAULT
```

Common Output:

TITLE	ERR	SYM
PRVGEOM	XKUAL	MTYPEW
PRVMODE	OPLAIC	MTYPET
DIHW	OSPAIC	COPLAN
DIHT	WTGEOM	NSUBDV
XMACH	WTGNAF	NSURF
NKVALS		
XKI		
XKS		
NT5	WTBL	SMOOTH
NT6	PRBOX	NDEG
INTAPE	PRPAIC	DPPCPR
INFSP	PRSAIC	
ISMPLW		GEOCPR
NPLAIC	PRMODS	MODCPR
NSPAIC	PRCOEF	
	PRNW	AICCPR
NOUTP	PRUW	
IOUFSP	PRSW	NWSCPR
	PRVP	
OSAIC	PRBL	GAFCPR
	PRSL	
	PRGNAF	
	PRDCP	
	PRGNAC	

5. GEOMETRY PROCESSOR

Fortran Program GEOMBX

Author: G. D. Kramer

Purpose: To read geometric data from cards and compute all necessary geometric parameters.

Method: Cards F through L are read in this section. As they are read they are printed, then checked for inconsistent or missing data, with suitable diagnostics. The leading and trailing edge data is checked in EDGCHK, then transformed to non-dimensional coordinates. Planform and diaphragm box code patterns are determined in BXCDPF and BXCDI, and optionally printed by PRNTBC. The fractional on-planform portion of all boxes cut by a planform edge is determined by GMAREA, which in turn calls ALPHAC and NTRCEP. If spatial AIC's are necessitated by non-zero dihedral angles or vertical separation of wing and tail, integer arrays MUAIC are determined for each AIC set (C,W,V). These serve as a map, so that only those AIC values needed will be calculated. The MUAIC arrays are computed in PWWAIC and PWTaic. All resulting arrays are written on scratch file IGEOsc.

Usage: The GEOMBX program is the main program of a secondary overlay. It is called by:

CALL OVERLAY (6HAFMBØX, 1, 2, 0)

Common Input:

OMACH	NSUBDV
TITLE	NSURF
PRVGEOm	MYBW
DIHW	PRBOX
DIHT	GEOCPR
XMACH	

Common Output:

COPLAN	MXBW	MXBT	FSMPLW
XSUBDV	MXBBW	MYBT	ICHORD
NSUBDV	MYBBW	MYBBT	IBOXF
NSUBD2	MXBSW	MXBST	IBOXL
NSUBCN	MYBSW	MYBST	ZLOC
B1	MYBBSW	MYBBST	
BLBETA	IXBW	IXBT	
B1S	XCENTR	IXBST	
B1BTAS	TLAX	CAPL	

WLAX	TLAZ	NSPATK
WLAZ	PSIT	
PSIW		

Arrays output on scratch file IGEOSC:

- IBOXW - Wing box codes (Wing and tail if
COPLAN = .TRUE.)
- IBOXT - Tail Box codes
- FEXLOC - Leading edge locations at chord centers
- TEXLOC - Trailing edge locations at chord centers
- ALPHA - Fractional areas of boxes cut by a
planform edge
- IJALPH - Locations of cut boxes, of the form
(1000*J+I)₈
- KPT - Table of contents for the MUAIC arrays
(and AIC's)
- MUAIC - Pointer array indicating where contri-
buting boxes will be found for one
spatial AIC set.

Fortran Subroutine EDGCHK

Author: G. D. Kramer

Purpose: Given the leading or trailing edge values, to check for illegal combinations.

Method: Either a leading or trailing edge is checked for monotonic increasing y-values, starting at zero. The last trailing edge value is compared with the previous last value. A leading edge is checked for monotonically increasing x-values.

Usage: DIMENSION XEDGE(10), YEDGE(10)
CALL EDGCHK (XEDGE, YEDGE, NEDGE, IEDGE, IRR)

Input Parameters:

XEDGE - Array of X-values for edge location points
YEDGE - Array of Y-values for edge location points
NEDGE - Number of points to check
IEDGE - =1, leading edge
 =2, trailing edge

Output Parameter:

IRR =0, Successful
 = 1, Non-monotonic y-values
 = 2, Non-monotonic x-values, leading edge only
 = 4, Y-values not starting at zero
 = 8, Tip y-values not agreeing
 Other, additive combination of above conditions

Fortran Subroutine BXCDPF

Author: G. D. Kramer

Purpose: To generate on-planform box codes for one surface, and store them in a compressed format.

Method: For each (subdivided) chord, the location of the leading edge and trailing edge (FEXLOC, TEXLOC) is determined. Codes for all boxes between those values are then set to 1 in subroutine NCODER. The remainder of the box code array is not changed.

Usage: The subroutine is called by:
CALL BXCDPF(XLE, YLE, NLE, XTE, YTE, NTE, LSROWS, IBOX)

Input Parameters:

$\begin{Bmatrix} XLE \\ YLE \end{Bmatrix} = \begin{Bmatrix} x \\ y \end{Bmatrix}$ locations of leading edge, measured in n_c, m_c, l_c system

$\begin{Bmatrix} XTE \\ YTE \end{Bmatrix} = \begin{Bmatrix} x \\ y \end{Bmatrix}$ locations of trailing edge, measured in n_c, m_c, l_c system

$\begin{Bmatrix} NLE \\ NTE \end{Bmatrix} =$ Number of $\begin{Bmatrix} \text{leading} \\ \text{trailing} \end{Bmatrix}$ edge definition points.

LSROWS = Maximum number of subdivided rows allowed.

In/Out Parameters:

$\begin{Bmatrix} IXBW \\ IXBT \end{Bmatrix} =$ Input: 0 indicates the $\begin{Bmatrix} \text{wing} \\ \text{tail} \end{Bmatrix}$ is to be done.
Output: Subdivided row of first unsubdivided box center on the surface.

Output Parameters:

IBOX Compressed box codes, 1 for on-planform boxes found, unchanged elsewhere. See Figure 4.

Output Common Parameters:

$\begin{Bmatrix} MXBS \\ MYBS \end{Bmatrix} \begin{Bmatrix} W \\ T \end{Bmatrix}$ Maximum $\begin{Bmatrix} X \\ Y \end{Bmatrix}$ (aft) extension of the subdivided $\begin{Bmatrix} \text{wing} \\ \text{tail} \end{Bmatrix}$ pattern

$\begin{Bmatrix} MYBS \\ MXB \end{Bmatrix} \begin{Bmatrix} W \\ T \end{Bmatrix}$ Maximum $\begin{Bmatrix} Y \\ X \end{Bmatrix}$ (outward) extension of the subdivided on-planform $\begin{Bmatrix} \text{wing} \\ \text{tail} \end{Bmatrix}$ pattern

$\begin{Bmatrix} MXB \\ MYB \end{Bmatrix} \begin{Bmatrix} W \\ T \end{Bmatrix}$ Maximum $\begin{Bmatrix} X \\ Y \end{Bmatrix}$ extension of the unsubdivided $\begin{Bmatrix} \text{wing} \\ \text{tail} \end{Bmatrix}$ pattern

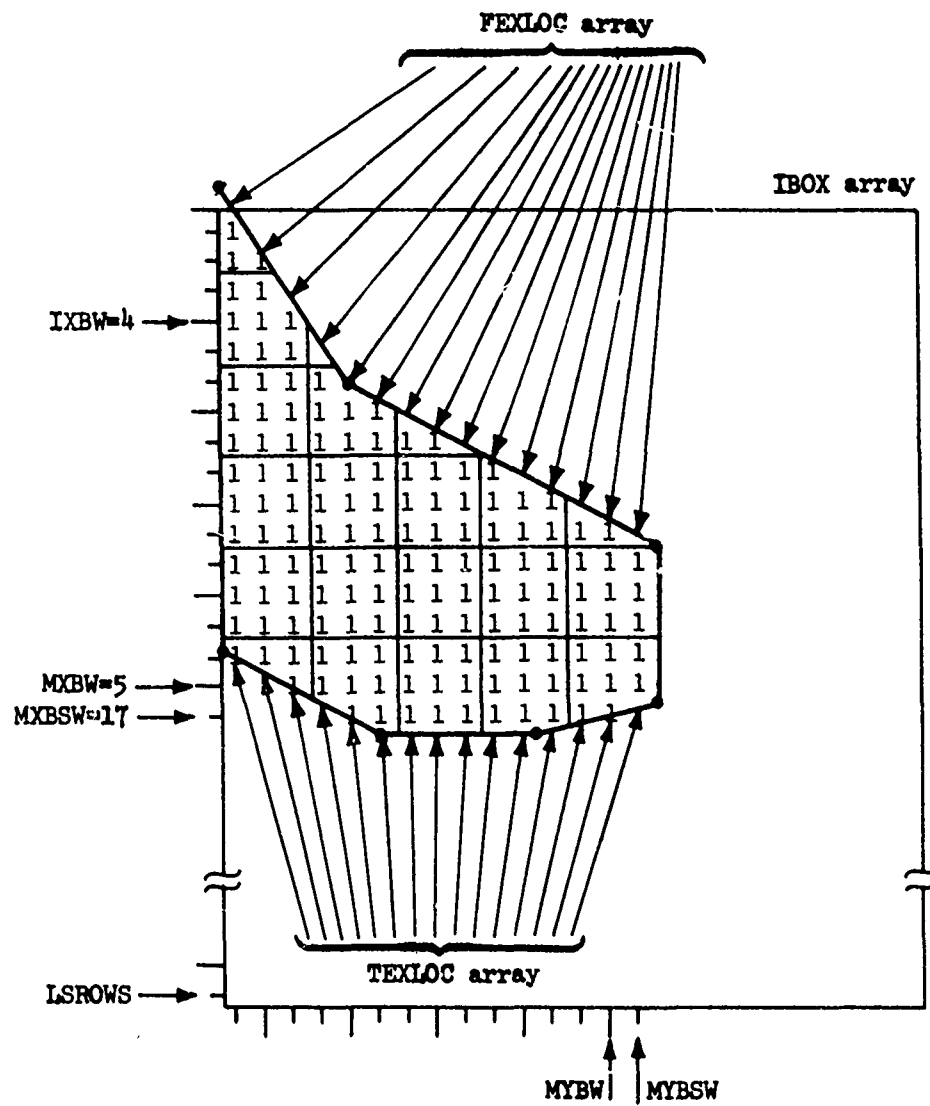


FIGURE 4 Output from BXCDPF, Called for a Wing (NSUBDV=3)

$MYB \begin{Bmatrix} W \\ T \end{Bmatrix}$ Maximum Y extension of the unsubdivided
 $\begin{Bmatrix} \text{wing} \\ \text{tail} \end{Bmatrix}$ pattern
 $\begin{Bmatrix} FEXLOC \\ TEXLOC \end{Bmatrix} = \begin{Bmatrix} \text{Leading} \\ \text{Trailing} \end{Bmatrix}$ edge X location at (subdivided)
 chord centers.

Fortran Subroutine BXCDI

Author: G. D. Kramer

Purpose: Given an array indicating a pattern of on-planform Mach boxes, to determine the associated off-planform diaphragm boxes

Method: Leading edge diaphragm boxes are first determined, followed by wake diaphragm boxes. The tip diaphragm is then determined as a function of the tip chord. For the wing, an integer array is interrogated to determine whether additional wake areas (and tip diaphragm) are needed for wing-tail interference.

Usage: The subroutine is called by:
CALL BXCDI (IWAKE, LSROWS, LSCHDS, IBOX)

Input Parameters:

IWAKE: Array of locations on the wing for aft-most unsubdivided box in each chord affecting a tail surface. Not used for the tail surface, first element = 0

LSROWS: Maximum number of subdivided rows allowed

LSCHDS: Maximum number of subdivided chords allowed

Input/Output

IBOX: Array of subdivided box codes, previously set 1 at planform locations by subroutine BXCDPF. See figure 5.

Common Input

MXBBSW } Maximum X extension of the subdivided {wing}
MXBBST } box pattern, including diaphragm {tail}

IXBST X-location of the first subdivided tail row

MYBSW } Maximum Y extension of the subdivided plan-
MYBST } form {wing} pattern {tail}

MYBBSW } Maximum Y extension of the subdivided {wing}
MYBBST } pattern, including diaphragm {tail}

NSUBDV Number of subdivisions

Common Output

MYBBSW } Modified, if necessary
MYBBST }

MYBSW } Maximum Y extension of the unsubdivided
MYBST } {wing} pattern, including diaphragm {tail}

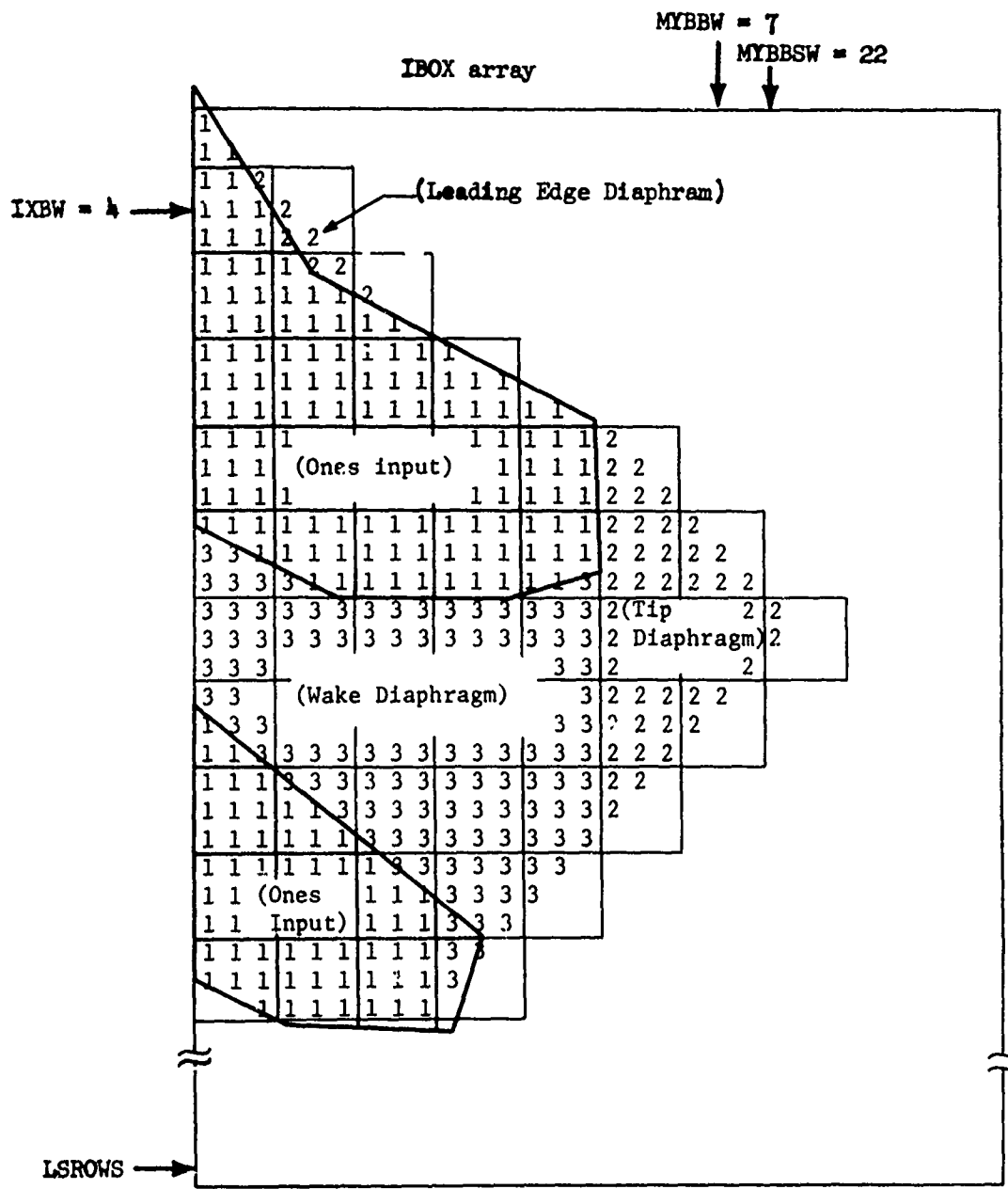


Figure 5 Output from BXCDI, called for a Coplanar Wing and Tail (NSUBDV = 3)

Fortran Subroutine PRNTBC

Author: G. D. Kramer, G. E. Keylon

Purpose: Print the array of box codes, either all values or only unsubdivided box-center values

Method: The compressed box code array is decompressed using subroutine DCODER, one row at a time, and printed. If unsubdivided codes have been requested, only the control point values are printed.

Usage: CALL PRNTBC (IBOX, LBXCD, IROW, MXB, MYB, SUBD)

- IBOX - Box code array
- LBXCD - Row size of box code array
- IROW - First row to print
- MXB - Last row to print
- MYB - Number of chords to print
- SUBD - .T., subdivided box codes desired
.F., unsubdivided (control point) box codes desired

Fortran Subroutine PWWAIC

AUTHOR: G. D. Kramer

PURPOSE: Given the box pattern and dihedral angle of the surface, to determine a pointer array (MUAIC) for one chord on the right surface which indicates contributing regions (if any) of the left surface on the given chord.

METHOD: The geometric relationship of the sending surface to the receiving chord is first determined. Then for all rows, from the last receiving box forward to the forward edge of the box pattern, any sending boxes on the left surface are indicated in the MUAIC array.

USAGE: CALL PWWAIC(WING,IBOX,LBXCD,IWAKE,JCOL)

Input Parameters:

WING .T., wing is being considered.
 .F., tail is being considered.

IBOX Array of box codes (IBOXW or IBOXT).

LBXCD Length of array IBOX.

IWAKE Array of locations of aft-most box to be considered on the wing. Ignored if WING = .F.

JCOL Chord being considered (receiving).

Common Input:

PSI $\begin{Bmatrix} W \\ T \end{Bmatrix}$ Dihedral angle

NSUBDV Number of subdivisions

XSUBDV Number of subdivisions, real

NSUBD2 NSUBDV/2

IXBW Location of first unsubdivided box center.

Output Parameters:

The computed results are returned via common block MUAICS. They are:

SURF Logical indicator - true means a sending surface was encountered.

MUAIC(2,50) Unsubdivided row "map" of sending box locations, see Figure 7.

EL Normal offset of receiving chord from sending surface.

YBAR Parallel offset of receiving chord.

NROWS Number of rows considered.

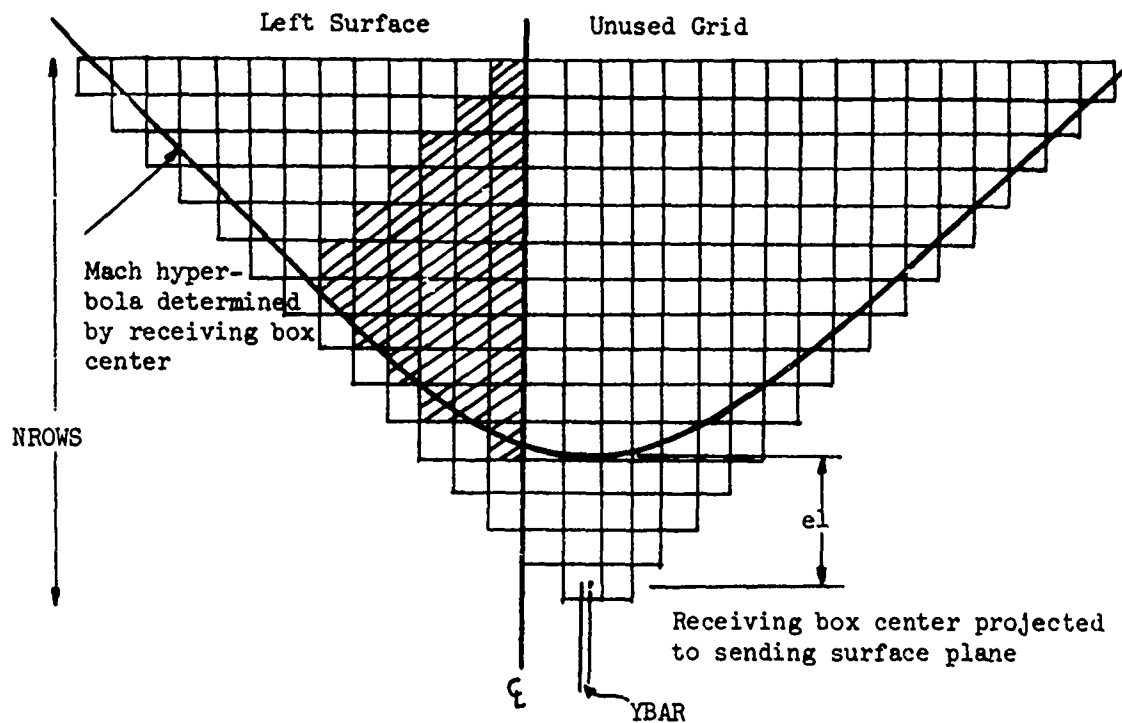


FIGURE 6. Possible Arrangement of Sending Boxes, Left Surface to Right
(Actual sending boxes shaded.)

0	0	0	0	3	2	2	2	2	3	5	7	9	11	13
0	0	0	0	3	4	5	6	7	8	9	10	11	12	13

NROWS = 15

FIGURE 7. MUAIC Array Generated by PWWAIC for Figure 4.

Fortran Subroutine PWTaic

Author: G. D. Kramer

Purpose: Given the box patterns and dihedral angles of the two surfaces, to determine pointer arrays (MJAIC arrays) for the right wing and the left wing contributing regions to a desired tail chord.

Method:] The geometric relationship of the sending surfaces to the receiving chord is first determined. Then for all rows, from the last receiving box forward to the forward edge of the sending box patterns, any sending boxes are indicated in the MJAIC arrays.

Usage: The subroutine is called, after suitable setup, by:
CALL PWTaic (IBOXW, LBXCDW, IROW, JCOL, CAPLL, YMUVSP)

Input Parameters:

IBOXW - Array of wing box codes
LBXCDW - Row dimension of IBOXW
IROW - Unsubdivided receiving row number
JCOL - Unsubdivided receiving chord number
CAPLL - Vertical Separation of sending center line
receiving center line
YMUVSP - $\bar{\mu}$ contribution due to vertical separation
= CAPLL * $\sin(\psi_w)$

Input Common Variables

PSIDIF = $\phi_r - \phi_w$ PSIT
NSUBDV FEXLOC
PSIW TEXLOC
MYBBW SYM
MYBSW
IXBW

Output Parameters

The computed results are returned via common block MJAICS. They are:

{SURF } = .T., Contributing boxes were found on the
{SURFL} {right } wing
{left }
= .F., No contributions were found

- $\begin{Bmatrix} \text{MUAIC} \\ \text{MUAICL} \end{Bmatrix}$. Map of contributing boxes on the $\begin{Bmatrix} \text{right} \\ \text{left} \end{Bmatrix}$ wing, see Figure 7.
- $\begin{Bmatrix} \text{EL} \\ \text{ELL} \end{Bmatrix}$ The normal offset between the wing $\begin{Bmatrix} \text{right} \\ \text{left} \end{Bmatrix}$ plane and the receiving point
- $\begin{Bmatrix} \text{YBAR} \\ \text{YBARL} \end{Bmatrix}$ The paralalled offset between the nearest chord center on the wing $\begin{Bmatrix} \text{right} \\ \text{left} \end{Bmatrix}$ box pattern and the receiving point
- $\begin{Bmatrix} \text{NROWS} \\ \text{NROWSL} \end{Bmatrix}$ Number of rows covered by the MUAIC array for the $\begin{Bmatrix} \text{right} \\ \text{left} \end{Bmatrix}$ wing contributions

Fortran Subroutine GMAREA

AUTHOR: G. E. Keylon, G. D. Kramer

PURPOSE: To compute the fractional on-planform portion of all planform boxes which are cut by a planform edge.

METHOD: For each chord, the X coordinates of the left side intercept, right side intercept, and any kinks within the box width are determined by subroutine NTRCEP for each planform edge cut by the chord (wing and/or tail). Then for each planform box on the chord, the routine determines whether any edge cuts the box or causes a contribution to the box area. For any affected box, subroutine ALPHAC is called to compute the fractional area, which is then stored in array ALPH, and its location is stored in array IJALPH as $(J * 512 + I)$. The fraction may be greater than one, since it includes the planform area of any chordwise adjacent box whose center is off planform.

USAGE: The routine is called by:

CALL GMAREA (IBOX, LBXCD, WING, ALPHA, IJALPH, NALPH)

Input Parameters:

IBOX Box code array
LBVCD Size of box code array
WING .T., Wing or coplanar case
.F., Tail

Input Common Parameters:

COPLAN	MXBT	NSURP	XWLE	XTLE
FEXLOC	MXBW	NWLE	YWLE	YTLE
TEXLOC	MYBT	NWTE	XWTE	XTTE
IXBT	MYBW	NTLE	YWTE	YTTE
IXBW	NSUBDV	NTTE		

Output Parameters:

ALPHA Array of area multipliers
IJALPH Array of corresponding IJ locations, as $(J * 512 + I)$
NALPH Number of fractions calculated

Fortran Subroutine NTRCEP

AUTHOR: G. E. Keylon

PURPOSE: To compute the X coordinates of the intersections of a planform edge with the sides and center of a chord.

METHOD: The routine determines in which interval of the edge the desired point lies. The x-coordinate of the point is then obtained from the standard two point equation of a line. This is done for all three points.

USAGE: The routine is called by:

CALL NTRCEP (J, YEDG, XEDG, L1, C1, R1, NBK1, K1, IDEX)

Input Parameters:

J = Chord number
YEDG } = Arrays of $\begin{Bmatrix} Y \\ X \end{Bmatrix}$ locations of the edge definition points
XEDG }
IDEX = 1, leading edge
 2, trailing edge

Common Input Values (from local common block/LAREA/)

LEFT = Y-location of the left side of the chord
RIGHT = Y-location of the right side of the chord

Output Parameters

L1 = X coordinate of left side intersection
C1 = X coordinate of center line intersection
R1 = X coordinate of right side intersection

NBK1 = Number of edge definition points encountered between the left and right sides of the chord.

K1 = 0 if no edge definition point lies between the left and right sides of the chord.
 = The first (leftmost) edge definition point number lying within the chord.

Subroutine ALPHAC

AUTHOR: G. E. Keylon, G. D. Kramer

PURPOSE: To compute the on-planform area of a box which is partially off the planform or which must include area from neighboring off-planform box(es) cut by a planform edge.

METHOD: If the box is the first box on the chord, or the last box on the chord, the box is divided spanwise into a series of trapezoids (or triangles) determined by planform edge definition points occurring within the chord. The areas of these trapezoids are then added, yielding α .

If the box is an interior box which is cut by one or more planform edge segments, the area is first set to one, then the area of the off-planform corner(s) determined as trapezoids or triangles is subtracted.

USAGE: The routine is called by

CALL ALPHAC (X, XLED, YLED, XTED, YTED, L1, C1, R1, NBK1, K1,
L2, C2, R2, NBK2, K2, AREA)

Input Parameters:

X = X coordinate of box center
XLED } = Planform leading edge definition points
YLED }
XTED } = Planform trailing edge definition points
YTED }
L1 } = { Left }
C1 } = { Center } Chord edge intersections with the planform
R1 } = { Right } leading edge
NBK1 = Number of planform leading edge definition points within
the chord
K1 = First leading edge definition point within chord
L2 }
C2 } Same as above for trailing edge
R2 }
NBK2 }
K2 }

Output Value:

AREA = The desired box area, α

6. MODAL DATA PROCESSOR

Fortran Program MODES

AUTHOR: G. E. Keylon

PURPOSE: To read the modal input data, compute it by a least squares surface fitting routine or evaluate a polynomial equation with coefficients supplied by input and store this information on a scratch file.

METHOD: The information needed to determine the mode shapes is read in. The planform information is read from a scratch file created in the geometry section. The program then computes or reads the modal data at control points, orders the data and writes the data on a scratch file for use in following sections.

Subroutine ROPER is used to compute row pointers for storing box center modal values row-wise. Modal input from tape is handled by TAPMOD. If modal input option 2 was specified, FITTER is called to compute the surface fit polynomial coefficients. PRECOF is called if the coefficients from option 1 or 2 are to be printed. The coefficients are saved on a scratch file for future cycles, and the polynomial is evaluated at box centers, with the results stored on scratch file MODESC.

The program also has an option to read an array of Thickness slope function values derived from "Piston Theory" calculations. These values are input to an equation that computes the thickness correction factor.

$$\bar{Z}_{\tau}(x,y) = 1 + \frac{\gamma+1}{2} \cdot M \frac{\partial z_{\tau}}{\partial x} \quad (1)$$

where γ is ratio of specific heats for a perfect gas (1.40)

M is Mach number.

$\partial z_{\tau} / \partial x$ is the thickness slope function values.

$\bar{Z}_{\tau}(x,y)$ is the thickness correction factor.

Input Methods:

(1) Polynomial Coefficient Input

The degree of a surface polynomial equation and the coefficients are read in. The deflection is then computed by the following polynomial equation:

$$\text{Deflection} = a_{00} + \sum_{i=1}^{\text{\# of degrees}} (a_{i0} x^i y^0 + \dots + a_{0i} x^0 y^i) \quad (2)$$

where, point (X,Y) is the coordinates of a box center in the planform local coordinate system, and a is the array of polynomial coefficients read in. The slope is computed by taking the derivative of the deflection in the X direction.

$$\begin{aligned} \text{Slope} &= \sum_{i=0}^{\text{\# of degrees}} d(a_{i0} x^i y^0 + \dots + a_{0i} x^0 y^i) / dx \\ &= \sum_{i=1}^{\text{\# of degrees}} (i a_{i0} x^{i-1} y^0 + \dots + a_{0i} y^i) \end{aligned} \quad (3)$$

These equations are used to compute the deflection and slopes for all the planform boxes. The array of modal values is stored on a scratch file for use in the velocity potential and generalized forces sections of the program.

(2) Interpolation

The degree of a surface polynomial equation, the number of locations where deflections are to be given and the locations and deflections are read in. The deflections are perpendicular to the surface and the (X,Y) locations are input in the planform local coordinate system. The program uses this data to fit a surface polynomial expression in the least squares error approximation. The routine that performs the surface fit is subroutine FITTER. This routine sets up an upper triangular, augmented matrix that represents the set of simultaneous linear equations formed by taking the partial derivatives of each deviation equation squared and setting it to zero. It then solves the set of simultaneous linear equations by using the Choleski square root method given in Reference 1. The solution is an array of polynomial coefficients that are used to compute the modal values in the same manner as method (1).

(3) Modal Values at Box Centers

The values of the deflections and slopes are read in from cards or tape and stored on a scratch file for use in the velocity potential and generalized forces sections of the program. The values are stored in order of boxes within chord, and chords within planform. The order is fore to aft boxes, center most to tip chord and wing before tail. For card input, each chord begins on a new card. All of the mode shape for the wing will be read followed by all of the mode shapes for the tail.

USAGE:

The MODES program is the main program of a secondary overlay of the Mach Box program. It is called as follows:

CALL OVERLAY (6HAFMBØX, 1, 3, 0)

Input:

Uses labeled common blocks:

/PROBLM/
/GEOMTY/
/GEOM2/
/FILES/
/IOCONT/
/TAPEIO/
/MODES/
/RWBUFF/

Uses the following files:

IGEOSC

Output:

Output is stored on file:

MODESC

Fortran Subprogram ROPER

AUTHOR: G. E. Keylon

PURPOSE: To compute the row pointers indicating location of planform boxes.

METHOD: The subprogram uses the column pointers and determines the row pointers. The subprogram will also calculate pointers for a tail surface with overlapped planform and store the pointers after the first planform pointers.

USAGE: CALL ROPER

General labeled common blocks used:

/GEOMTY/
/GEOM2/

LOCAL labeled common blocks used:

/INDEX/ IS(100), NOC(100), JS(50), JOC(50)

Common Input:

IS(J) - The ith index of the first planform box on chord J.

NOC(J) - The number of planform boxes on chord J.

Common Output:

JS(I) - The jth index of the first planform box on row I.

JOC(I) - The number of planform boxes on row I.

Fortran Subprogram FITTER

AUTHOR: G. E. Keylon

PURPOSE: To fit a surface in the least squares sense through a set of data points.

METHOD: The fitter routine is passed a set of ordered triplets and the degree of polynomial to fit. It is also given a scale factor if needed to scale the data to prevent the occurrence of arithmetic overflow or underflow. The program can fit real or complex data. The system of simultaneous linear equations that must be solved for employs the Choleski square root method (see Ref. 1). If the polynomial exceeds the maximum capability in either X or Y direction that degree is held and the other direction is allowed to use the full degree.

USAGE: CALL FITTER (M, N, X, Y, Z, C, CN, IDIM)

Input:

M - degree of polynomial equation
N - number of data points to fit curve through
X - Array of X coordinates
Y - Array of Y coordinates
Z - Array of Z coordinates
CN - scale factor
IDIM - Indicator of real or complex function
 = 1, function to fit is real
 2, function to fit is complex

Output:

C - Output polynomial coefficient array.

Fortran Subprogram MODOUT

AUTHOR: G. E. Keylon

PURPOSE: To print the mode shapes in a manner that the user can readily determine Mach box values of deflections and slopes.

METHOD: The mode shapes are rearranged in a print array so that one row or part of a row will be printed at a time. If there are more than 15 chords on the planform the program prints information for 15 chords, for all rows, and then prints for the next 15 chords until all information has been printed. The values may be scaled before printing to allow values to be printed under F mode Fortran format control. The scaling factor will be indicated in the title.

USAGE: CALL MODOUT (DEFSL, JS, JOC, NROWS, NM, IOVLAP)

Input:

DEFSL - Array of mode shapes
DEFSL(1,I) = deflection
DEFSL(2,I) = slope
JS - Array of pointers to first planform box on each row
JOC - Array of counters for the number of planform boxes on each row.
NROWS - number of rows
NM - Mode shape number
IOVLAP - Number of boxes of overlap between planforms for non-coplanar surfaces.

Output:

None

Fortran Subprogram PRECOF

AUTHOR: G. E. Keylon

PURPOSE: To print the polynomial coefficients used in evaluating mode shapes.

METHOD: The coefficients are printed with each coefficient having over it the corresponding powers of X and Y labeled. All the coefficients for a total power will on one line (i.e., line 1 - 0 power, line 2 - first power, line 3 - second power etc.).

USAGE: CALL PRECOF(IDEQ, A, IFR)

Input:

Labeled common block /FILES/

IDEQ - Degree of polynomial equation

A - Array of coefficients

IFR - Flag indicating how coefficients are obtained.

= 1, read from cards

= 2, computed by least squares surface fit.

Output:

None

7. AERODYNAMIC INFLUENCE COEFFICIENTS SECTION

Fortran Program VICMAIN

AUTHOR: G. E. Keylon

PURPOSE: To determine all aerodynamics influence coefficients (AIC's) that must be computed or retrieved for a specified reduced frequency.

METHOD: A parameter array is read from the geometry scratch file for each spatial AIC that is needed. The program then determines if an array already exists on permanent tape storage. If it exists the array is read in, expanded if necessary, and stored on scratch file IAICSC if spatial, or in blank common if planar. If calculation is necessary, subroutine KERNEL is called to control the actual computations. KERNEL in turn calls ROMBER to do the integrations of FUNCT and VFUNC.

USAGE: The VICMAIN program is the main program of a secondary overlay of the Mach Box program. It is called as follows:

CALL OVERLAY (6HAFMBØX 1, 4, 0)

Input:

Uses labeled common blocks

/KERN/
/KVAL/
/PROBLM/
/FILES/
/GEOMTY/
/IOCONT/
/ARRAYS/
/RWBUFF/
/TAPEIO/

Uses the following files

IGEOSC, OSPAIC (optional), OPLAIC (optional)

Output:

Output is stored on files:

NPLAIC, NSPAIC, IAICSC (all optional)

Fortran Subprogram KERNEL

AUTHOR: G. E. Keylon

PURPOSE: To determine the boxes to be integrated and the limits of integration prior to calling the integration routine.

METHOD: The program determines from a parameter array from the geometry scratch file, the intersection of the Mach cone with the planform boxes it is attempting to integrate. It determines what boxes on a row are to be integrated and breaks each box up into a set of integrable limits. It then passes the limits of integration to subroutine ROMBER for integration by the Romberg integration method described in Reference 2.

Box Patterns and Limits:

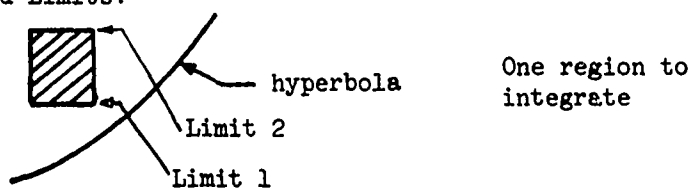


FIGURE 8 AIC Integration, Full Box

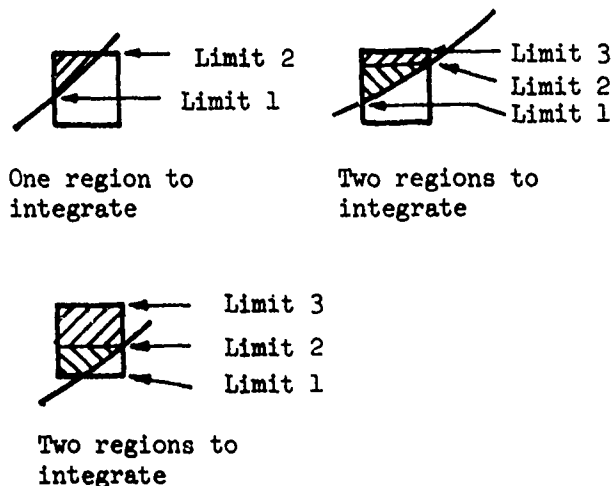


FIGURE 9 AIC Integration Edge Boxes

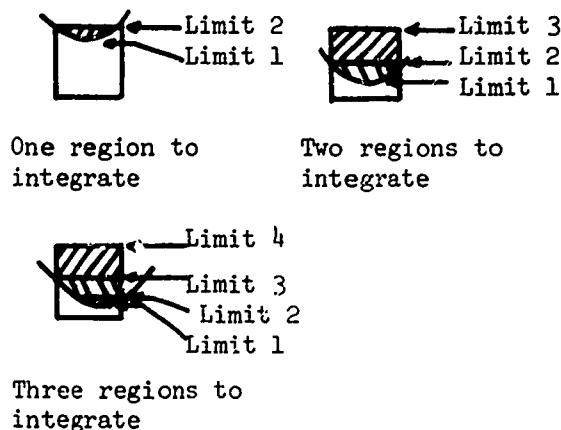


FIGURE 10 AIC Integration Apex Boxes

Box patterns and limits for boxes cut on the left side by the Mach hyperbola are computed in a like manner.

The functions integrated by subroutine ROMBER are those for the velocity potential aerodynamic influence coefficients ($C_{\bar{v}\bar{\mu}\bar{\lambda}}$), the upwash aerodynamic influence coefficients ($W_{\bar{v}\bar{\mu}\bar{\lambda}}$), and the sidewash aerodynamic influence coefficients ($V_{\bar{v}\bar{\mu}\bar{\lambda}}$).

The equations for $C_{\bar{v}\bar{\mu}\bar{\lambda}}$ and $W_{\bar{v}\bar{\mu}\bar{\lambda}}$ are:

$$C_{\bar{v}\bar{\mu}\bar{\lambda}} = -\frac{1}{\pi} \int_{\bar{\xi}_l}^{\bar{\xi}_u} e^{-i\bar{k}_1 \bar{\xi}} \left\{ J_0 \left(\frac{\bar{k}_1}{M} \sqrt{\bar{\xi}^2 - \ell^2} \right) \left[\sin^{-1} \left(\frac{\bar{\eta}_u}{\sqrt{\bar{\xi}^2 - \ell^2}} \right) \right. \right. \\ \left. \left. - \sin^{-1} \left(\frac{\bar{\eta}_L}{\sqrt{\bar{\xi}^2 - \ell^2}} \right) \right] + \sum_{r=1}^{\infty} \frac{(-1)^r}{r} J_{2r} \left(\frac{\bar{k}_1}{M} \sqrt{\bar{\xi}^2 - \ell^2} \right) \right. \\ \left. \left[\sin \left(2r \sin^{-1} \left(\frac{\bar{\eta}_u}{\sqrt{\bar{\xi}^2 - \ell^2}} \right) \right) - \sin \left(2r \sin^{-1} \left(\frac{\bar{\eta}_L}{\sqrt{\bar{\xi}^2 - \ell^2}} \right) \right) \right] \right\} d\bar{\xi}$$

$$\begin{aligned}
W_{\bar{\rho}, \bar{\mu}, \bar{\lambda}} = & \frac{\lambda}{\pi} \left[\int_{\bar{\xi}_L}^{\bar{\xi}_U} e^{-i\bar{k}_1 \bar{\xi}} \left\{ J_0 \left(\frac{\bar{k}_1}{M} \sqrt{\bar{\xi}^2 - \ell^2} \right) \left[\sin^{-1} \left(\frac{\bar{\eta}_U}{\sqrt{\bar{\xi}^2 - \ell^2}} \right) - \right. \right. \right. \\
& \left. \left. \sin^{-1} \left(\frac{\bar{\eta}_L}{\sqrt{\bar{\xi}^2 - \ell^2}} \right) \right] + \sum_{r=1}^{\infty} \frac{(-1)^r}{r} J_{2r} \left(\frac{\bar{k}_1}{M} \sqrt{\bar{\xi}^2 - \ell^2} \right) \cdot \right. \\
& \left. \left[\sin \left(2r \sin^{-1} \left(\frac{\bar{\eta}_U}{\sqrt{\bar{\xi}^2 - \ell^2}} \right) \right) - \sin \left(2r \sin^{-1} \left(\frac{\bar{\eta}_L}{\sqrt{\bar{\xi}^2 - \ell^2}} \right) \right) \right] \right\} \frac{1+i\bar{k}_1 \bar{\xi}}{\bar{\xi}^2} d\bar{\xi} \\
& + \left| \frac{e^{-i\bar{k}_1 \bar{\xi}}}{\bar{\xi}} \left\{ J_0 \left(\frac{\bar{k}_1}{M} \sqrt{\bar{\xi}^2 - \ell^2} \right) \left[\sin^{-1} \left(\frac{\bar{\eta}_U}{\sqrt{\bar{\xi}^2 - \ell^2}} \right) - \sin^{-1} \left(\frac{\bar{\eta}_L}{\sqrt{\bar{\xi}^2 - \ell^2}} \right) \right] \right. \right. \\
& + \sum_{r=1}^{\infty} \frac{(-1)^r}{r} J_{2r} \left(\frac{\bar{k}_1}{M} \sqrt{\bar{\xi}^2 - \ell^2} \right) \left[\sin \left(2r \sin^{-1} \left(\frac{\bar{\eta}_U}{\sqrt{\bar{\xi}^2 - \ell^2}} \right) \right) \right. \\
& \left. \left. - \sin \left(2r \sin^{-1} \left(\frac{\bar{\eta}_L}{\sqrt{\bar{\xi}^2 - \ell^2}} \right) \right) \right] \right\} \right]_{\bar{\xi}_L}^{\bar{\xi}_U} + e^{-i\bar{k}_1 \bar{\xi}} \frac{\pi}{\ell} \quad (5)
\end{aligned}$$

The $C_{\bar{\rho}, \bar{\mu}, \bar{\lambda}}$ equations is valid for all types of boxes, which are shown in Figures 8, 9, and 10. The $W_{\bar{\rho}, \bar{\mu}, \bar{\lambda}}$ equation, as written, is valid for the region of an apex box shown in Figure 10, that has the hyperbola as a boundary on both sides. The last term is zero for regions that have the hyperbola as a boundary for one side and the box edge as the other side boundary. The last term and the terms evaluated at the integration limits are zero for full boxes or regions bounded on both sides by the box edges. The values of the integrand used in ROMBER are computed by subroutine FUNCT which also calls subroutines RANGE and BESSEL to evaluate the Bessel functions.

The function $V_{\bar{\rho}, \bar{\mu}, \bar{\lambda}}$ is considerably different and holds for all regions to be integrated. The evaluation of the integral is done by subroutine VFUNCT. The equation is:

$$\begin{aligned}
v_{\bar{\nu}\bar{\mu}\bar{\lambda}} = & -\frac{M}{\pi \bar{k}_1} \left[\left| \frac{-i\bar{k}_1\bar{\xi}}{\bar{\xi}} \frac{1+i\bar{k}_1\bar{\xi}}{\bar{\xi}^2} \left(\sin\left(\frac{\bar{k}_1}{M}\sqrt{\bar{\xi}^2 - \bar{\eta}_u^2 - l^2}\right) \right. \right. \right. \\
& \left. \left. - \sin\left(\frac{\bar{k}_1}{M}\sqrt{\bar{\xi}^2 - \bar{\eta}_L^2 - l^2}\right) \right) \right]_{\bar{\xi}_L}^{\bar{\xi}_u} + \int_{\bar{\xi}_L}^{\bar{\xi}_u} \left[e^{-i\bar{k}_1\bar{\xi}} \frac{1+i\bar{k}_1\bar{\xi}}{\bar{\xi}^2} \right. \\
& \left. \left(\sin\left(\frac{\bar{k}_1}{M}\sqrt{\bar{\xi}^2 - \bar{\eta}_u^2 - l^2}\right) - \sin\left(\frac{\bar{k}_1}{M}\sqrt{\bar{\xi}^2 - \bar{\eta}_L^2 - l^2}\right) \right) \right] d\bar{\xi} \quad (6)
\end{aligned}$$

For cases where $\bar{k}_1 = 0$, the equation becomes

$$v_{\nu\mu\lambda} = -\frac{1}{\pi} \left[\cosh^{-1} \frac{\bar{\xi}}{\sqrt{\bar{\eta}_u^2 + \bar{\lambda}^2}} - \cosh^{-1} \frac{\bar{\xi}}{\sqrt{\bar{\eta}_L^2 + \bar{\lambda}^2}} \right] \left[\right]_{\bar{\xi}_L}^{\bar{\xi}_u} \quad (7)$$

USAGE: CALL KERNEL(XMACH,K1,ERR,C,W,V)

Input:

XMACH Mach number
K1 Reduced frequency
ERR Convergence criteria (relative not absolute)
Labeled Common Block /VICPAR/

Output:

C Velocity potential aerodynamic influence coefficients.
W Upwash aerodynamic influence coefficients.
V Sidewash aerodynamic influence coefficients.

Fortran Subprogram ROMBER

AUTHOR: G. E. Keylon

PURPOSE: To integrate the aerodynamic influence coefficient functions.

METHOD: The program uses the Romberg integration technique (Ref. 2). The technique is a modified trapezoidal area method with an extrapolation method added. For analytical cases the sidewash aerodynamic influence coefficient will be solved by an analytic equation, not by numerical approximation.

USAGE: CALL ROMBER (XILL, XILU, IUC, ERR, IFLAG, K1BAR, YMUBAR, EL, XMACH
C, W, V)

Input:

XILL - Lower limit of integration
XILU - Upper limit of integration
IUC - Flag indicating type of box or edge condition of interval to be integrated = 0, full box
1, left side of box is edge of Mach hyperbola.
2, right side of box is edge of Mach hyperbola.
3, both sides of box are edges of Mach hyperbola.

ERR - Convergence criteria (relative, not absolute)
IFLAG - Indicator of real or imaginary parts
= 0, real part
= 1, imaginary part
K1BAR - Function of reduced frequency and Mach number, $= \frac{K_1 M^2}{(M^2 - 1)}$

YMUBAR - Parallel offset of pulse sending box.
EL - Normal offset of receiving point from sending plane.
XMACH - Mach number

Output:

C - Velocity potential aerodynamics influence coefficient, $C_{\bar{v}\bar{\mu}\bar{\lambda}}$
W - Upwash aerodynamic influence coefficient, $W_{\bar{v}\bar{\mu}\bar{\lambda}}$
V - Sidewash aerodynamic influence coefficient, $V_{\bar{v}\bar{\mu}\bar{\lambda}}$

Fortran Subprogram FUNCT

AUTHOR: G. E. Keylon

PURPOSE: To evaluate the velocity potential and upwash aerodynamic influence coefficient functions for a set of independent variables.

METHOD: An array XI of independent variables is passed to the program through the calling sequence. The program evaluates the function at each point first checking for boundary conditions where the function approaches a singularity. Routines to find the range of and value of Bessel functions are called in the evaluation of the function.

USAGE: CALL FUNCT (K, XI, FXIC, FXIW, IFLAG, KIBAR, EL, YMUBAR, IUC, XMACH, BESSY)

Input:

K - Number of functions to evaluate
XI - Array of independent variables
IFLAG - Indicator of real or imaginary part:
= 0, real part
= 1, imaginary part
KIBAR - Function of reduced frequency and Mach number, $K, M^{\infty} / (K^2 - 1)$
EL - Normal offset of receiving point from sending plane.
YMUBAR - Parallel offset of pulse sending box.
IUC - Flag indicating type of box or edge condition of interval to be integrated.
XMACH - Mach number

Output:

FXIC - Function values for Velocity Potential AIC.
FXIW - Function values for Upwash AIC.
BESSY - Evaluation at end points for upwash AIC.

Fortran Subprogram BESSEL

AUTHOR: G. E. Keylon

PURPOSE: To evaluate the Bessel functions for a given argument over a range of orders.

METHOD: The argument and range (# of terms or order) is passed to the routine. The routine then calculates the required terms and places them in an array and returns.

USAGE: CALL BESSEL (K12, AV, NA)

Input:

K12 - The argument, a function of independent variable, Mach number and reduced frequency.

NA - Highest order of the Bessel function to be evaluated.

Output:

AV - Array containing the Bessel functions.

Fortran Subprogram RANGE

AUTHOR: G. E. Keylon

PURPOSE: To determine the range (or order) of a Bessel function with a given argument.

METHOD: An order, or equation for an order, is given for various increments of arguments. This routine determines which interval the argument is in and computes the order.

USAGE: CALL RANGE (K12, NA)

Input:

K12 - The argument, function of independent variable, Mach number and reduced frequency.

Output:

NA - Highest order of the Bessel function to be evaluated.

Fortran Subprogram VFUNC

AUTHOR: G. E. Keylon

PURPOSE: To evaluate the sidewash aerodynamic influence coefficient function for a set of independent variables.

METHOD: An array of independent variables is passed to the program through the calling sequence. The program evaluates the function at each point, first checking for boundary conditions where the function approaches a singularity.

USAGE: CALL VFUNC (K, XI, FXIV, IFLAG, K1BAR, EL, YMUBAR, INC, XMACH, IND, VT)

Input:

K - Number of values to calculate
XI - Array of independent variables
IFLAG - Flag indicating real or complex part
 = 0, real part
 = 1, imaginary part
K1BAR - Function of reduced frequency and Mach number, $K_1 M^2 / (M^2 - 1)$
EL - Normal offset of receiving box above sending plane.
YMUBAR - Parallel offset of pulse sending box.
INC - Flag indicating type of box or edge condition of interval to be integrated.
XMACH - Mach number
IND - Indicator to calculate VT terms
 = 0, do not calculate
 = 1, calculate

Output:

FXIV - Function values for sidewash AIC.
VT - Extra terms calculated at the limits of integration.

8. NORMAL-WASHES AND VELOCITY POTENTIALS

Fortran Program NWLPT

AUTHOR: G. D. Kramer

PURPOSE: To compute normal washes and associated velocity potentials for each oscillatory mode shape at box centers. Wake sampling of upwash, sidewash and longitudinal wash is also provided.

METHOD: The necessary box patterns and other geometric items are first read in from the scratch file IGEO SC. The mode shape and velocity potential pointer array IPNTRM is read from scratch file MODE SC, and a pointer array for normal-washes, IPNTDW, is generated by subroutine POINTR. These pointer arrays serve to associate a box location in a sparsely filled rectangular array with the corresponding mode, velocity potential or normal wash value in a singly dimensioned, densely filled array.

A loop on mode shapes is entered next. The box center deflections and shapes are read from MODE SC into array DEFSL. Subroutine VELPOT is called for the wing to compute N_{RUW} , N_{RW} , and $\Delta\bar{\phi}$ at box centers, and trailing edge $\Delta\bar{\phi}$ values in array TVP. If a tail is being analyzed as well, the contributing wing normal-washes are determined and VELPOT is called again. Optional printing of N_{RUW} , etc. and $\Delta\bar{\phi}$ is done in routine PRINTR.

If sampling of wake washes is desired, subroutine SMPLW is called to compute and print these results.

The $\Delta\bar{\phi}$ array VELPOT and the TVP array are written on scratch file IVPSC for each mode shape.

USAGE: The DWLPT program is the main program of a secondary overlay of the Mach box program. It is called as follows:

CALL OVERLAY (62AFMB0X, 1, 5, 0)

Input: Uses labelled common blocks

/CONTRL/	/FILES/
/PROBLM/	/IOCONT/
/GEOMTY/	/TAPEIO/
/GEOM2/	/MODES/
/KERN/	/ARRAYS/
/KVAL/	/SAMPLW/

Uses scratch files

IGEO SC
IMODE SC

Fortran Subroutine POINTR

AUTHOR: G. E. Keylon, G. D. Kramer

PURPOSE: To generate part or all of a pointer array which indexes another array of box associated values (modes, normal-washes, etc.) stored compactly, row-wise.

METHOD: The box codes are scanned to determine the first box of interest and the number of boxes of interest on each row. From this, the pointer array is generated such that IPNTR(1,1) = the location of the first box value for row 1, and IPNTR (2, 1) = the chord number of the first box value for row 1.

USAGE: The routine is called by:

```
DIMENSION IBOX (LBXCD, # chords/10), IPNTR (2, MXIR)
LOGICAL DIAPH, SUBD, WING
```

```
CALL POINTR (IX, MX, MYB, IOVLAP, SUBD, DIAPH, IBOX, LBXCD, MXIR,
             IPOINT, IPNTIN, IPNTR)
```

Input Parameters:

IX = First row of the box pattern for which the pointer array is desired.
MX = Number of rows desired.
MYB = Maximum row length
IOVLAP = Number of rows to allow for overlap (tail only).
SUBD = .T., a pointer array for subdivided boxes is desired
 = .F., only unsubdivided box information is desired.
DIAPH = .T., boxes in diaphragm areas are to be included.
 = .F., only on-planform boxes are of interest though space may be left within a row if imbedded diaphragm areas occur.
IBOX = Array of subdivided box codes generated in the geometry section.
LBXCD = Length of box code array.
MXIR = Length of IPNTR array, used to control end-around buildup of the array.
IPOINT = Value to be used for first pointer; 1 if IX=1, else the next location available in the array "pointed to" for row IX.

In/out Parameters:

IPNTIN = Location of next available cell in the IPNTR array. This will be incremented for each row processed until MXIR is reached, when it is reset to 1.

IPNTR (2, MXIR) = The pointer array, see Method above.

FORTRAN Subroutine GETAIC

Author: G. D. Kramer

Purpose: To get the desired Aerodynamic Influence Coefficient (AIC) arrays from scratch file IAICSC.

Method: From the calling sequence, the location of the desired AIC array(s) is determined. If they are in core, the routine returns. If there are none, the error flag is set. Otherwise, the disk file is positioned, and the desired arrays are read into local common block AICS.

Usage: EL, YBAR, NROWS, MUAIC (2,50) are in a common block, MUAICS for output from GETAIC
 NWWAIC, NTTAIC, NRWTAIC, NLWTAIC and PAIC (4,50) are in a common block /PAICS/, for use by the routine.
 CALL GETAIC (JUCENT, ITYPE, ICODE, IR)

Input Parameters:

JUCENT = receiving chord number
 IYPE = 1, 2, 3, 4 indicating wing-wing, tail-tail, right-wing-tail, or left-wing-tail AIC's desired
 ICODE = 0, C,V,W desired
 1, V,W desired
 2, V desired

Common Input:

$$\left. \begin{array}{l} \text{NWWK} \\ \text{NTTK} \\ \text{NRWTK} \\ \text{NLWTK} \end{array} \right\} = \text{Number of AIC arrays available for influence.}$$

$$\left. \begin{array}{l} \text{wing-wing} \\ \text{tail-tail} \\ \text{right wing-tail} \\ \text{left wing-tail} \end{array} \right\}$$

PAIC (4,50) = Table of contents for the AIC's.
 PAIC (I,J) indicates where the AIC's for the Ith form of influence (see above) on the Jth chord are located.

Output Parameters:

IR = D, Success
 1, C not found
 2, C and W not found
 3, Nothing found

Common Output:

$$C = C_{\frac{v_{\mu\omega}}{v_{\mu\omega}}}$$

$$W = W_{\frac{v_{\mu\omega}}{v_{\mu\omega}}}$$

$$V = V_{\frac{v_{\mu\omega}}{v_{\mu\omega}}}$$

Fortran Subroutine VELPOT

AUTHOR: G. D. Kramer

PURPOSE: To compute normal wash and velocity potential values for one mode shape.

METHOD: This routine calculates the following equations:

$$\frac{D\bar{f}_j^{n,m}}{Dt} = \left[i k_j f_j^{n,m} + b_j \frac{\partial f_j^{n,m}}{\partial x} \right] \quad (8)$$

(1) For the wing:

$$N_{RUW}^{n,m} = \frac{D\bar{f}_j}{Dt} - \hat{N}_{RUW}^{n,m} \quad , \quad N_{RLW}^{n,m} = -\frac{D\bar{f}_j}{Dt} - \hat{N}_{RLW}^{n,m} \quad (9)$$

where

$$\hat{N}_{RUW}^{n,m} = \sum_{\substack{\text{left wing} \\ + \text{diaphragm}}} \left[\cos 2\psi_w W_{\bar{\mu}\bar{\lambda}}^{(RW)} - \sin 2\psi_w V_{\bar{\mu}\bar{\lambda}}^{(RW)} \right] N_{RUW}^{\mu,\mu} * \text{SYM} \quad (10)$$

and

$$\hat{N}_{RLW}^{n,m} = - \sum_{\substack{\text{left wing} \\ + \text{diaphragm}}} \left[\cos 2\psi_w W_{\bar{\mu}\bar{\lambda}} - \sin 2\psi_w V_{\bar{\mu}\bar{\lambda}} \right] N_{RLW}^{\mu,\mu} * \text{SYM} \quad (11)$$

and $\text{SYM} = \begin{cases} +1.0, & \text{symmetric} \\ -1.0, & \text{antisymmetric} \end{cases}$

(2) For leading edge or tip diaphragm boxes:

$$\begin{aligned} (N_{RUS} - N_{RLS})^{n,m} = & \frac{1}{C_{\text{geo}}} \left[- \sum_{\substack{\text{right wing} \\ + \text{diaphragm}}} C_{\bar{\mu}\bar{\lambda}0} (N_{RUW}^{\mu,\mu} - N_{RLW}^{\mu,\mu}) \right. \\ & \left. + \text{SYM} * \sum_{\substack{\text{left wing} \\ + \text{diaphragm}}} C_{\bar{\mu}\bar{\lambda}\lambda} (N_{RUW}^{\mu,\mu} - N_{RLW}^{\mu,\mu}) \right] \quad (12) \end{aligned}$$

(3) For wake diaphragm boxes:

$$(N_{RUS} - N_{RLS})^{n,m} = \frac{1}{C_{000}} \left[\Delta \bar{\phi}_j^{n,m} - \sum_{\substack{\text{right wing} \\ + \text{diaphragm}}} C_{j\bar{\mu}\bar{\lambda}0} (N_{RUW} - N_{RLW})^{\nu\mu} \right. \\ \left. + \text{SYM} + \sum_{\substack{\text{left wing} \\ + \text{diaphragm}}} C_{j\mu\lambda} (N_{RUW} - N_{RLW})^{\nu\mu} \right] \quad (13)$$

where

$$\Delta \bar{\phi}_j^{n,m} = \Delta \phi_{TE}^m e^{-ik_1 \left(\frac{X_n - X_{TEm}}{b_1} \right)} \quad (14)$$

$$(N_{RUS}^{n,m} + N_{RLS}^{n,m}) = - \left(\hat{N}_{RUS}^{n,m} + \hat{N}_{RLS}^{n,m} \right) \quad (15)$$

(4) For the tail:

$$(N_{RUT} - N_{RLT})^{n,m} = 2 \left(\frac{Df^{n,m}}{Dt} \right) + \left(\hat{N}_{PLT}^{n,m} - \hat{N}_{RUT}^{n,m} \right) \\ - 2 \hat{N}_{RUT}^{n,m} - 2 \hat{N}_{RUT}^{n,m} \quad (16)$$

where $\hat{N}_{RUT}^{n,m}$ and $\hat{N}_{RLT}^{n,m}$ are computed as in Equation (11) and (12).

$$\hat{N}_{RUT}^{n,m} = \sum_{\substack{\text{rt wing} \\ + \\ \text{diaph.}}} \left[\cos(\psi_T - \psi_W) W_{\bar{\nu}\bar{\mu}\bar{\lambda}}^{(RT)} - \sin(\psi_T - \psi_W) V_{\bar{\nu}\bar{\mu}\bar{\lambda}} \right] N_{RW}^{\nu\mu} \quad (17)$$

$$\hat{N}_{RUT}^{n,m} = \text{SYM} \sum_{\substack{\text{left wing} \\ + \\ \text{diaph.}}} \left[\cos(\psi_T + \psi_W) W_{\bar{\nu}\bar{\mu}\bar{\lambda}}^{(LT)} - \sin(\psi_T + \psi_W) V_{\bar{\nu}\bar{\mu}\bar{\lambda}} \right] N_{LW}^{\nu\mu} \quad (18)$$

(5) Velocity potentials:

$$\Delta \bar{\phi}_j^{n,m} = \sum_{\substack{\text{right surface} \\ \text{and diaphragm}}} C_{\bar{\mu}0} (N_{RUS}^{\bar{\mu}} - N_{RLS}^{\bar{\mu}}) + \sum_{\substack{\text{left surface} \\ \text{and diaphragm}}} C_{\bar{\mu}1} (N_{RUS}^{\bar{\mu}} - N_{RLS}^{\bar{\mu}}) * SYM \quad (19)$$

Because the equations involve summations over unknown values, the order of calculation is very critical. The routine computes normal washes and velocity potentials in parallel, one row at a time, inboard-most box first. If the subdivision option is on, each subdivided box must have a set of normal washes computed as well, using equations similar to those above.

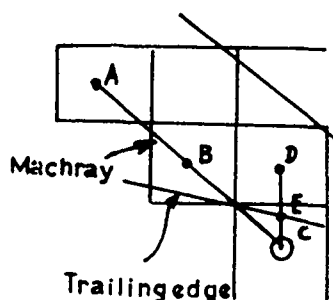
For each box, the $\hat{N}$ terms are first zeroed out. If spatial contribution is present, subroutine GETAIC is called to get the necessary AIC arrays from scratch file IAICSC, and the proper summation is computed over the forward Mach hyperbola. This is first done for $\hat{N}_{RUS}$ and $\hat{N}_{RLS}$ and stored in variables ENRULU, ENRLLL, LUS, LLS

If the surface is a tail, a similar procedure of getting AIC arrays and computing the proper summation for $\hat{N}_{RUT}$ and $\hat{N}_{RLT}$ is followed. The results are stored in ENRURW and ENRULW.

If the box being considered is a planform box, the normal wash values are next computed from Equations (9) or (16).

Function B is called to compute the planar $\Delta \bar{\phi}$ contribution, except for the contribution of the box to itself. This is stored in variable DELPH. If the box is on planform, the out-of-plane contribution is added, yielding Equation (19). If the box is on a diaphragm, DELPH is used in Equation (12) or (13) to eventually yield the normal wash values at the diaphragm box center.

Trailing edge velocity potentials, array TVP, are computed whenever a trailing edge box is encountered. The computation is normally linear extrapolation from the last two box center values. In the event there is only one box on the tip chord, a Mach ray extrapolation is first done, followed by chord-wise linear interpolation. See Figure 11.



Values at A and B are extrapolated to C. Then the values at C and D are interpolated to give a value at E, the desired trailing edge value.

FIGURE 11 Tip Chord Trailing Edge Velocity Potential Calculation

The subdivision option causes the following:

- (1) All row and column loops are on subdivided boxes.
- (2) Any necessary $\hat{N}$ terms are calculated once per control point, and stored in temporary arrays for use on all subdivided boxes within the unsubdivided box. $\hat{N}$ terms and spatial contribution of left surface to $\Delta \bar{\phi}$ are not calculated using subdivided values.
- (3) Function B and $\Delta \bar{\phi}$ are not computed for on-planform subdivided boxes which do not contain a control point.
- (4) Function B, when called, applies two equations - one within the "effective area" of subdivision, and the other outside this area. It is within function B that the subdivision refinement actually takes place.
- (5) Any unsubdivided box which has one or more off-planform subdivided boxes has its normal wash values computed as the average of all subdivided values within its bounds, i.e.

$$N_{RUS}^{n,m} = \left(\sum_{\substack{\text{all subdivided} \\ \text{boxes on box } n,m}} N_{RUS}^s \right) / NSUBDV$$

JSAGE: The subroutine is called by:
CALL VELPOT(IBOX,LBXCD,PKERNL,SKERNL,WING,DIHS)

Input Parameters:

IBOX Array of box codes for the surface.
LBXCD Length of the box code array.
PKERNL Primary (unsubdivided) $C_{\overline{V}\overline{\mu}0}$ array.
SKERNL Subdivided $C_{\overline{V}\overline{\mu}0}^{k_1/N_s}$ array.
WING .TRUE., the surface is a wing.
 .FALSE., the surface is a tail.
DIHS .TRUE., any surface dihedral is to be accounted for.
 .FALSE., any surface dihedral may be ignored.

Input Common Variables:

Global common blocks used:

/GEOMTY/
'GEOM2/
/MODES/
/FILES/
/CHECKPR/
Blank Common for C

Local common values:

/MUAICS/YBAR	Parallel offset
EL	Normal offset
MUAIC(2,50)	AIC pointer array determined in the geometry section.
NROWS	Number of rows defined for the AIC set.
/AICS/ XKVL	Current value of K_1
C	C
W	W
V	V
/DELTAP/{TEXLOC}	{leading } edge X-locations at chord centers
{FEXLOC}	{trailing }
IPNTRM	Pointer array for modes and velocity potentials.
DEFSL	Mode shape array - equivalenced to velocity potential array.
IOVLAP	Measure of tail overlap of wing, box mode shapes.
/NWASHES/ IPNTDW	Pointer array for normal wash values.
/BXCDES/ IBOXW	Wing box codes

Output Common Variables:

/DELTAP/	DELPHI	$\Delta \bar{\phi}$ array
	TVP	$\Delta \bar{\phi}_{TE}$ array
/NWASHES/	ENRUS	N_{RUW} or N_{RUT}
	ENRLS	N_{RLW} or N_{RLT}
	IOVLAPH	Measure of tail overlap of wing diaphragm, for normal washes.
/SNWASH/	IPNTSD	Pointer array for subdivided normal washes
	ENSUBD	$N_{RUS}^{(S)}$ and $N_{RLS}^{(S)}$
	IPNTIN	{ End-around pointers for array IPNTSD }
	IPNTOT	
	IPNTLS	

Fortran Function B

AUTHOR: G. D. Kramer

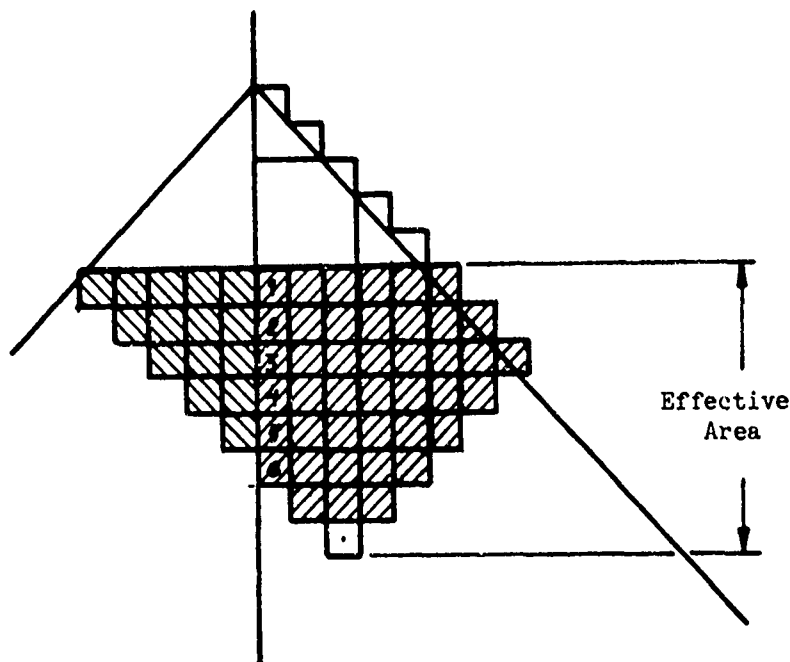
PURPOSE: Given the location of a Mach box, to compute the planar contribution of the rest of the surface to the velocity potential difference for the box.

METHOD: The routine has two sections, one for computing the subdivided contribution within the "effective area," and the other to compute the unsubdivided contribution from ahead of the "effective area". If the subdivision option is off, the second section is used for the full contribution.

In the first section, the summation performed is

$$B_s = \sum_{\bar{v}} \sum_{\bar{\mu}} C_{\bar{v}\bar{\mu}0}^{k_1/N_s} (N_{RUS}^{(s)} - N_{RLS}^{(s)}) \quad (21)$$

where the summation limits are as shown in Figure 12.



 Always contributes to the summation
 Contributes if the dihedral angle $\gamma = 0$

FIGURE 12 Subdivided "Effective Area"

The second section starts up where the first leaves off, and computes the second summation in Equation (22)

$$B = B_s + \sum \sum C_{\bar{v}/\bar{\mu}_0} (N_{RUS} - N_{RLS}) \quad (22)$$

where the summation limits are as shown in Figure 13 and the AIC array and normal wash values are now unsubdivided, computed at control points.

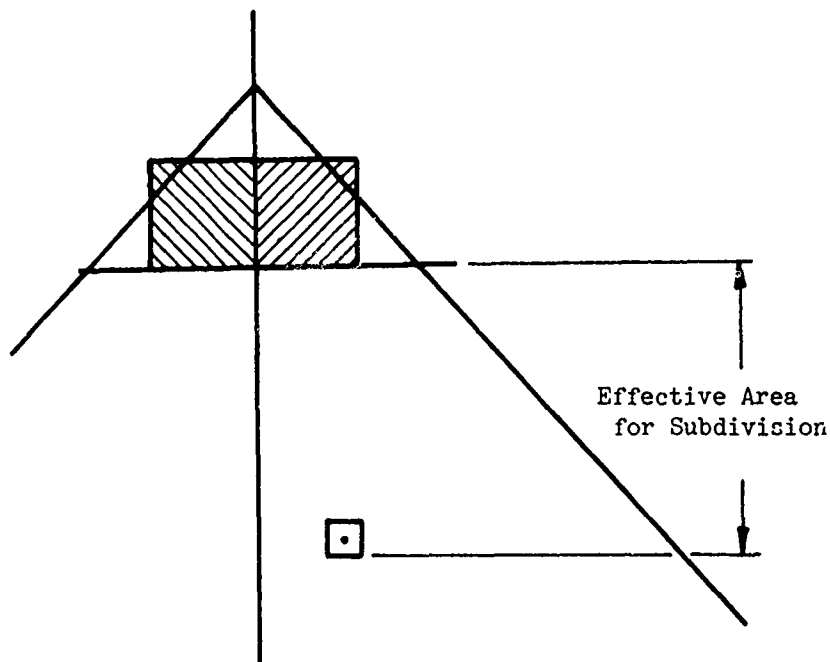


FIGURE 13. Unsubdivided Boxes Outside the "Effective Area"

USAGE:

The function is called by:

DELPH = B(IROW,JCOL,PKERNL,SKERNL,IBOX,LBXCD,WING,DIH)

Input Parameters:

IROW JCOL	} Location of receiving point, subdivided
PKERNL	Primary (unsubdivided) $C_{\bar{v}\bar{\mu}0}$ array
SKERNL	Subdivided $C_{\bar{v}\bar{\mu}0}^{k_i/N_i}$ array
IBOX	Box code array
LBXCD	Length of box/code array
WING	.TRUE., the surface is the wing .FALSE., the surface is the tail
DIH	.TRUE., leftside is to be ignored .FALSE., Include left side.

Input Common Parameters:

See subroutine VELPOT. Both subdivided and unsubdivided values are used.

Output:

The function value, B, in this case stored in DELPH, is the result of the summations described under METHOD.

Fortran Subroutine SMPLW

AUTHOR: G. D. Kramer

PURPOSE: To compute and print upwashes, sidewashes, and longitudinal washes at arbitrary chord locations in the wake of a wing.

METHOD: This routine is called once for each sampling chord. For each box on the chord, the right wing contribution is summed as

$$WSUM = \sum_{rt. wing} W_{\bar{\rho} \bar{\mu} \bar{\lambda}} * N_{RW}^{\nu \mu} \quad (23)$$

$$VSUM = \sum_{rt. wing} V_{\bar{\rho} \bar{\mu} \bar{\lambda}} * N_{RW}^{\nu \mu} \quad (24)$$

$$PHISUM = \sum_{rt. wing} C_{\bar{\rho} \bar{\mu} \bar{\lambda}} * N_{RW}^{\nu \mu} \quad (25)$$

where $N_{RW} = \begin{cases} N_{RUW} & \text{if the chord is above the wing,} \\ N_{RLW} & \text{if the chord is below the wing.} \end{cases}$

These sums are then combined as:

$$UW_R = \frac{u}{U} = \left(1/b_1 \right) * \left(\cos \psi_w * WSUM + \sin \psi_w * VSUM \right) \quad (26)$$

$$SW_R = \frac{v}{U} = \left(1/b_1 \right) * \left(\cos \psi_w * VSUM - \sin \psi_w * WSUM \right) \quad (27)$$

$$PHI_R = PHISUM$$

The left wing contributing summations are identical to Equations (23), (24), and (25), with N_{RW} replaced by N_{LW} .

The results are then combined by

$$UW_{complete} = UW_R + \left(1/b_1 \right) * \left(\cos \psi_w * WSUM - \sin \psi_w * VSUM \right) * SYM \quad (28)$$

$$SW_{complete} = SW_R + \left(1/b_1 \right) * \left(\cos \psi_w * VSUM - \sin \psi_w * WSUM \right) * SYM \quad (29)$$

$$\text{PHI}_{\text{complete}} = \text{PHI}_R + \text{PHISUM} * \text{SYM} \quad (30)$$

The printed upwash and sidewash is given by Equations (28) and (29). For longitudinal wash, the PHI values computed in Equation (30) are used in

$$\text{LW}(I) = \frac{1}{2b_1/\beta} \left[\text{PHI}(I+1) - \text{PHI}(I-1) \right] \quad (31)$$

USAGE:

The routine is called by:

CALL SMPLW(IBOX,LBXCD,JCHRD,JT,IFRST,ILAST)

Input Parameters:

IBOX	Array of wing box codes
LBXCD	Length of box code array
JCHRD	Sample wash chord number, with reference to the order specified in the card data
JT	The y-location of the chord
IFRST	Number of the first sample box desired
ILAST	Number of the last sample box desired

9. VELOCITY POTENTIAL SMOOTHING SECTIONS

FORTRAN Program SMTH

Author: G. E. Keylon

Purpose: To smooth the velocity potentials by using a least squares surface fitting technique.

Method: The velocity potentials are read in from a disk file and smoothed with a least squares fit by subroutine FITTER, previously described. The polynomial equation derived from the fit is then used to compute an array of velocity potentials at planform box centers.

Usage: The SMTH program is the main program of a secondary overlay of the Mach box program. It is called as follows:

```
CALL OVERLAY (6HAFMBØX, 1, 6, 0)
```

Input:

USES LABELLED COMMON BLOCKS

```
/ARRAYS/  
/FILES/  
/IOCONT/  
/PROBLM/  
/KVAL/  
/GEOMTY/  
/GEOM2/  
/TAPEIO/  
/RWBUFF/
```

Uses the following files
MODESC, IGEOSC, IVPSC

Output:

Output is stored on file IWTFSC which is changed to IVPSC.

FORTTRAN Program CRDFIT

Author: G. E. Keylon

Purpose: To smooth the velocity potentials by using a least squares curve fit along each chord.

Method: The velocity potentials are read in from a disk file. The values for each chord are then separated into an array. The values are then changed to the numerical slope between the midpoint average values. Subroutine CURVE is then called to fit a least squares polynomial curve through these slopes. The polynomial equation is then integrated at each box on the chord and the integral value becomes the velocity potential at that box.

Usage: The CRDFIT program is the main program of a secondary overlay of the Mach box program. It is called as follows:

CALL OVERLAY (6HAFMBOX, 1, 7, 0)

Input:

USES LABELED COMMON BLOCKS

/ARRAYS/
/FILES /
/IOCONT/
/PROBLM/
/KVAL /
/GEOMTY/
/GEOM2 /
/TAPEIO/
/RWBUFF/

Uses the following files

MODESC, IGEOSC, IVPSC

Output:

Output is stored on file IWTFSC which is changed to IVPSC.

FORTRAN Subprogram CURVE

Author: G. E. Keylon

Purpose: To fit a curve in the least squares sense through a set of data points.

Method: The CURVE routine is passed a set of ordered complex pairs and the degree of polynomial to fit. The system of simultaneous linear equations is solved employing the Choleski square root method (see Ref. 1). If the polynomial degree exceeds the limits possible to fit the degree is reduced to a lower level.

Usage: CALL CURVE (M,N,X,Z,C)

Input:

- M - degree of polynomial equation
- N - number of data points to fit curve through
- X - Array of X coordinates (independent variable)
- Z - Array of Z coordinates (dependent variable, complex)

Output:

- C - output polynomial coefficient array, complex

10. GENERALIZED AIR FORCES SECTION

Fortran Program FORCES

AUTHOR: G. E. Keylon, G. D. Kramer

PURPOSE: To calculate the boxlifts, section lifts, and generalized air forces for a problem.

METHOD: Planform information is first read from the geometry and modes scratch files. The outer-most loop on thickness slope functions is then entered. One set of thickness slope functions, defined at box centers by Equation (1), is read in from scratch file ITSLSC. Next a loop on mode shapes, used as weighting functions for the generalized forces calculations, is entered. One mode shape is read from scratch file MODESC.

The third loop entered is on velocity potentials. The $\Delta\phi$ array is read into DELPHI and $\Delta\phi_{TE}$ into array TVP from scratch file IVPSC. The box pattern for each surface is then passed over, one row at a time. For each box the following values are computed:

$$\bar{L}_j^{nm} = \text{BXLIFT}(IDC) = \frac{2}{\beta} \left[\Delta\bar{\phi}_{jTE}^{nm} - \Delta\bar{\phi}_{jLE}^{nm} + i\alpha^{nm} k_1 \Delta\bar{\phi}_j^{nm} \right] \bar{Z}_\tau^{nm} \quad (3.1)$$

$$\Delta C_{Pj}^{nm} = \text{DELCP}(IDC) = \bar{L}_j^{nm} / (\alpha^{nm} * b_1) \quad (3.2)$$

$$\bar{Q}_{ij}^{nm} \beta_{1/2} = \left[\left(f_{TE}^{nm} * \Delta\bar{\phi}_{TE}^{nm} - f_{LE}^{nm} * \Delta\bar{\phi}_{LE}^{nm} \right) - b_1 \alpha^{nm} \frac{\partial f}{\partial x} \Delta\bar{\phi}_j^{nm} + i\alpha^{nm} k_1 f^{nm} \Delta\bar{\phi}_j^{nm} \right] \bar{Z}_\tau^{nm} \quad (3.4)$$

The $\bar{Q}_{ij}^{nm} \beta_{1/2}$ terms are summed as calculated, and stored as

$$\text{AFROW}(JVP) = \sum_m \sum_n Q_{ij}^{nm} \beta_{1/2} \quad (3.5)$$

After all boxes have been processed, if boxlifts and section lifts are desired and this is the first mode shape, box lifts are printed, section lifts are computed and printed, and total lift is printed:

$$\bar{L}_j^m = \text{SLIFT}(JCOL) = \sum_n \bar{L}_j^{nm} \quad (3.6)$$

$$\bar{L}_j = \text{TLIFT} = \sum_m \bar{L}_j^m \quad (3.7)$$

After this has been done for all velocity potentials, one row of the final generalized air forces arrays is computed as:

$$\bar{Q}_{ij} = \text{GENAF}(IJ) = 2/\beta \text{ AFROW}(JVP) \quad (38)$$

$$Q'_{ij} = -b_1/s^3/\beta \text{ Re} [\bar{Q}_{ij}] \quad (39)$$

$$Q''_{ij} = -b_1^2/(k_1 s^4/\beta) \text{ Im} [\bar{Q}_{ij}] \quad (40)$$

The program does the above for all mode shapes, prints the results, optionally writes them on tape, then terminates. Printing is done in routines PRNTBL, PRNTSL, and PRNTAF.

For Equations (32) and (34) box leading and trailing edge values are needed. Several geometric conditions exist:

- (1) Box leading or trailing edge is internal to the planform:
Linear interpolation is used,

$$f_{LE}^{nm} = \frac{1}{2} (f^{n-1,m} + f^{n,m}) \quad (41)$$

$$\Delta \bar{\phi}_{LE}^{nm} = \frac{1}{2} (\Delta \bar{\phi}^{n-1,m} + \Delta \bar{\phi}^{n,m}) \quad (42)$$

and similarly for the box trailing edge.

- (2) Box is cut by the planform leading edge:

$$f_{LE}^{nm} = f^{nm} - (X_n - X_{LE}) * \frac{\partial f^{nm}}{\partial x} \quad (\text{point-slope}) \quad (43)$$

$$\Delta \phi_{LE}^{nm} = \begin{cases} 0 & \text{for wing or spatial tail} \\ \Delta \bar{\phi}_{TE \text{ of wing}} * e^{-i \frac{(X_{LE_{tail}}^m - X_{TE_{wing}}^m)}{b_1}} k_1 & \end{cases} \quad (44)$$

(3) Box is cut by the planform trailing edge

$$f_{TE}^{nm} = f^{nm} + (X_{TE} - X_n) \cdot \frac{\partial f^{nm}}{\partial x} \quad (45)$$

$\Delta \bar{\phi}_{TE}$ Computed planform trailing edge value, TVP, as described under subroutine VELPOT. This normally is a linear extrapolation using the forward adjacent box center and the current one for the two necessary $\Delta \bar{\phi}$ values.

USAGE: The FORCES program is the main program of a secondary level overlay of the Mach Box program. It is called as follows:

CALL OVERLAY(6HAFMBØX, 1, 7, 0)

Input:

Uses labeled common blocks:

/ARRAYS/
/FILES/
/IOCONT/
/KERN/
/KVAL/
/PROBLM/
/MODES/
/GEOMTY/
/GEOM2/
/TAPEIO/
/RWBUFF/

Uses the following files:

MODESC
IPNTRM
IVPSC
ITSLSC

Output:

Printer and tape NOUTP (optional).

11. COMMON BLOCK ORGANIZATION

The basic geometric and program control parameters are stored in a set of labeled common blocks which are loaded with the primary level overlay and thus are available to any secondary overlay. Some of the variables come directly from card input values (see Part I, Section III of this report), and others are internally computed.

	Default
/CONTRL/ PREVEX, OMACH, TITLE(8), PRVGEOM, PRVMODE, DIHW, DIHT, DEFAULT	
PREVEX Tested for code word in the data preprocessor link to determine whether defaults should be set or prior status maintained (recycle)	--
OMACH Mach # from previous cycle, compared on recycle to determine whether planform geometry needs changing.	0.
TITLE(8) One-line title for all printed headings	blank
PRVGEOM .T. previous geometry is to be used this cycle .F. New geometry is to be read	.F.
PRVMODE .T. previous modes are to be used this cycle .F. new modes are to be read	.F.
DIHW} .T. {Wing} dihedral is to be used computing DIHT} {Tail} influence on itself	.F.
.F. The {wing} {tail} is to be considered flat in computing influence on itself, but dihedral will be used in wing/tail calculations	
DEFAULT .T. All parameters on Card C are to be set to their default values .F. Do not set parameters to default.	.F.
/PROBLM/ XMACH, NMODES, NTSLOP, NKVALS, SMOOTH, NDEG, CRDFIT, EXAIC, SUBDV, PLYWOOD	
XMACH = Mach number for current cycle	no default
NMODES = Number of input modes to use	no default
NTSLOP = Number of thickness slope functions to be used	0
NKVALS = Number of reduced frequencies to be used	0
SMOOTH = .T., Velocity potentials surface smoothing desired .F., No velocity potential surface smoothing desired	.F.
NDEG = Maximum order for smoothing polynomial	0, program will determine
CRDFIT = .T., Chordwise velocity potential smoothing desired .F., No chordwise smoothing desired	.F.
EXAIC = .T., Integration accuracy of 10^{-4} desired .F., Integration accuracy of 10^{-2} desired	.F.
SUBDV = .T., Subdivision is to be applied .F., No subdivision is desired	.F.
PLYWOOD = .T., Full box areas to be used in box lifts .F., Planform box areas to be used.	

Default

```

/GEOMTY/ COPLAN, NSUBDV, XSUBDV, NSUBD2, NSUBCN, NSURF,
          B1, B1BETA, B1S, B1BTAS, WLAX, WLAZ, PSIW, MXBW, MXBBW,
          MYBW, MYBBW, MXBSW, MYBSW, MYBBSW, IXBW, XCENR

/GEOM2/  TLAX, TLAZ, PSIT, MXBT, MYBT, MYBBT, MXBST,
          MYBST, MYBBST, IXBT, IXBST, CAPL

COPLAN  .T. PSIW = PSIT and CAPL = 0. One box array  --
        is used
        .F. The 2 surfaces are not coplanar, or only
        one surface is defined
NSUBDV  Number of subdivided rows (columns) per box  1
XSUBDV  = Float (NSUBDV) 1.0
NSUBD2  = NSUBDV/2 0
NSUBCN  = NSUBD2 + 1 = center location of first chord 1
NSURF   Number of surfaces, 1 or 2 1
B1       Box length = B1BETA *  $\sqrt{M^2 - 1}$  --
B1BETA   b1/β, box width, = YWLE(NWLE)/MYBW --
B1S      } Subdivided box {length = B1/XSUBDV --
B1BTAS}   {width = B1BETA/XSUBDV} --
WLAX}    {Wing} local axis location, in global X co- 0.
TLAX}    {Tail} ordinate 0.
WLAZ}    {Wing} local axis location, in global Z co- 0.
TLAZ}    {Tail} ordinate 0.
PSIW}    {Wing} dihedral angle, input in degrees but 0.
PSIT}    {Tail} immediately changed to radians. 0.
MXBW}    Number of rows to aftmost portion of the {wing} --
MXBT}    measured in the nc coordinate {tail} --
MXBBW   Number of rows to aftmost wing diaphragm --
        box, nc coordinate
MYBW}    Number of chords on the {wing}, mc coordinate {=NCHRDS} --
MYBT}    {tail} --
MYBBW}   Number of {wing} chords, including tip --
MYBBT}   {tail} diaphragm --
MXBSW}   Subdivided MXB {W} count --
MXBST}   {T} --

```


		Default
MYBSW } MYBST }	Subdivided MYB { $\begin{smallmatrix} W \\ T \end{smallmatrix}$ } count	-- --
MYBBSW } MYBBST }	Subdivided MYBB { $\begin{smallmatrix} W \\ T \end{smallmatrix}$ } count	-- --
IXBW } IXBT }	Subdivided grid X-location of the first unsubdivided { $\begin{smallmatrix} \text{wing} \\ \text{tail} \end{smallmatrix}$ } box center	-- --
IXBST	Subdivided grid X-location of the first subdivided tail box	--
XCENR	X_w location of the center of the first box on the wing	No default
CAPL	Non-dimensionalized vertical distance be- tween centerlines of the wing and tail	0.
/KERN/ ERR, MXSKRN, IPKERN, NPLKRN, NSPATK, NR ~~O~~ WEA		
ERR	Integration accuracy in AIC calculations	.01
MXSKRN	Size of the subdivided AIC, array (number of rows)	
IPKERN	Location in array SKERNL where PKERNL(1) would be if it were not overlaid by the subdivided $C_{\nabla_{po}}$ array.	1
NPLKRN	Size of the planar AIC array (number of rows)	--
NSPATK	Number of spatial AIC arrays necessary	0
NR ~~O~~ WEA	Number of rows for the subdivided effective area	--
/KVAL/ IKVAL, XKVAL(20), XKS (20)		
IKVAL	Current k-value number being solved	
XKVAL	Array of reduced frequencies, k_1 , based on box length, b_1	
XKS	Array of reduced frequencies, k_s , based on semispan, s .	
/FILES/ NT5, NT6, INTAPE, INFSP, NPLAIC, NSPAIC, HOUTP, IOUFSP, MODESC, IVPSC, IGEOSC, IWFSC, IAICCC		
NT5	Card file (INPUT)	1
NT6	Print file (OUTPUT)	1
INTAPE	Binary input tape number, If 0 or 1 card input will be used	1

Default

INFSP	Initial file spacing on the input tape	0
NPLAIC	Tape number for the { planar } AIC arrays	0
NSPAIC		0
NOUTP	Binary output tape number. If 0, none written	0
IOUFSP	Initial file spacing on tape NOUTP	0
MODESC, IVPSC	} Internal scratch files	
IGEOSC, IUTFSC		
IAICSC		

/IOCONT/ OPLAIC, OSPAIC,
 WTGEOM, WTGNAF, WTSI, WTBL, PRBOX, PRPAIC,
 PRSAIC, PRMODS, PRCOEF, PRUW, PRSW, PRVP, PRBL,
 PRDCP, PRGNAF, PRGNAC, PRSL, PRLW, PRNW

OPLAIC	} .T., an old { planar } AIC tape is being used	.T.
OSPAIC		.F., a new { spatial } AIC tape is being used
WTGEOM	Not used	.F.
WTGNAF	.T., Write generalized air forces on tape	.T.
WTSI	.T., Write section lifts on tape	.F.
WTBL	.T., Write box lifts on tape	.F.
PRBOX	.T., Print the box code pattern(s)	.F.
PRPAIC	} .T., Print the { planar } AIC arrays	.F.
PRSAIC		.F.
PRMODS	.T., Print modal deflections and slopes	.F.
PRCOEF	.T., Print modal polynomial coefficients, if available	.F.
PRUW	} .T., for wake wash sampling, print { upwashes side washes longitudinal washes	.F.
PRSW		.F.
PRLW		.F.
PRVP	.T., Print velocity potential differences	.F.
PRBL	.T., Print box lifts, $\bar{L}_j^{n,m}$	.F.
PRDCP	.T., Print change in pressure, $\Delta C_{p_j}^{n,m}$	.F.
PRGNAF	.T., Print generalized airforces, $\bar{Q}_{ij}$	.T.
PRGNAC	.T., Print generalized aerodynamic coefficients, Q' and Q''	.F.
PRCM	.T. Print sectional generalized airforces, $\bar{Q}_{ij}^m$	.F.

		Default
PRSL	.T., Print section lifts, $\bar{L}_j^m$	.F.
PRNW	.T., Print normal washes, N_{RUW} , N_{RUW} , etc.	.F.
/TAPEIO/ NFS, NMS, LS, NMR, ID(20), NID, ITYPE, LRS, LWS, M, N, PARM(10), IRR		
DIMENSION IPARM(10)		
EQUIVALENCE (PARM, IPARM)		
NFS }	{File }	0
NMS }	{Matrix } spacing	0
LS }	Not used	
NMR }		
ID	ID array for the matrix	
NID	Number of words in the ID array on tape	1
ITYPE	Matrix type - MIXED, COMPLEX	
LRS }	Not used	
LWS }		
M }	Matrix dimensions	--
N }		--
PARM	Numerical parameters for the matrix	--
IRR	Error return	-0-
/MODES/ SYM, SYMT, MTYPEW, MTYPET		
SYM	1, Symmetric modes	1
	-1, Antisymmetric modes	
	0, Left surface contribution will be ignored	
SYMT	As above, for a non-planar tail. Differs only for vertical tail	SYM
MTYPEW	1, Polynomial coefficients will be read for the wing	2
	2, Deflections at arbitrary locations will be read	
	3, Box center values will be read	
MTYPET	1, Same as above for the tail	
	2,	2
	3,	
/ARRAYS/ KBXCDW, LBXCDW, LBOXC, KBXCDT, LBXCDT, KJALPH, LJALPH, KALPHA, KKERNL, LKERNL, KPNTRM, LPNTRM, KDEFSL, KELPHI, LMODES, KPNTSD, LPNTSD, KSDW, LSDW, KPNTDW, LPNTDW, KDW, LDW, KTV, LTV		

Locations and limits for arrays:

Variable	Array affected	Value
KBXCDW } LBXCDW } LBOXC }	IBOXW (LBXCDW, LBOXC)	{ Not used 150 8
KBXCDT } LBXCDT }	IBOXT (LBXCDT, LBOXC)	{ Not used 90
KJALPH } LJALPH }	IJALPH (LJALPH)	{ Not used 200
KALPHA	ALPHA (LJALPH)	Not used
KKERNL } LKERNL }	SKERNL (LKERNL), PKERNL	{ 1 1640
KPNTRM } LPNTRM }	IPNTRM (2, LPNTRM)	{ Not used 100
KDEFSL	DEFSL (2, LMODES)	Not used
KELPHI } LMODES }	DELPHI (LMODES), complex	{ Not used 500
KPNTSD } LPNTSD }	IPNTSD (2, LPNTSD)	{ Not used 50
KSDW } LSDW }	ENSUBD (2, LSDW)	{ Not used 600
KPNTDW } LPNTDW }	IPNTDW (2, LPNTDW)	{ Not used 100
KDW } LDW }	ENRUS (LDW), ENRLS (LDW)	{ No. used 1275
KTVP } LTVP }	TVP (LTVP), TEXLOC (LTVP), FEXLOC (LTVP)	{ Not used 250

De Gault

```
/RWBUFF/  BFCODE, IBFCNT, BUFF (3280)

      BFCODE = Code word
      IBFCNT = Size of buffer
      BUFF   = Buffer array for use by READMX and WRITEMX

8HBUFFSIZE
3280

/SAMPLW/  ISMPLW, ICHORD(10), IBOXF(10), IBOXL(10), ZLOC(10)
      ISMPLW  Number of chords specified for wash sampling      0
      ICHORD  Chord number for sampling                          -
      IBOXF   First box on chord to be sampled                  -
      IBOXL   Last box on chord to be sampled                   -
      ZLOC    Z-location of sampling chord, transformed         0.
              internally to correspond to wing coordinates

/PLANXY/  NWLE, NWTE, NTLE, NTTE, XWLE(10), YWLE(10),
          XWTE(10), YWTE(10), XTLE(10), YTLE(10),
          XTTE(10), YTTE(10)

      NWLE }      Number of wing {leading edge } definition
      NWTE }      {trailing edge} points

      NTLE }      Number of tail {leading edge } definition
      NTTE }      {trailing edge} points

      XWLE }      Wing leading edge definition points
      YWLE }

      XWTE }      Wing trailing edge definition points
      YWTE }

      XTLE }      Tail leading edge definition points
      YTLE }

      XTTE }      Tail trailing edge definition points
      YTTE }
```

COMMON/ CHECKPR/ DPPCPR, GEOCPR, AICCPR, NWSCPR,
SMCPR, GAFCPR

These variables are all typed logical. They control whether
or not internal checkout print statements will be
executed. They will be read from Card C, default .FALSE.

```
DPPCPR  Data preprocessor check-print
GEOCPR  Geometry check-print
AICCPR  AIC section check-print
NWSCPR  Normal wash & velocity potential check-print
SMCPR   Velocity potential smoothing check-print
GAFCPR  Generalized Airforces check-print
```

12. ARRAY STORAGE

In order to conserve storage, a number of arrays are used as pointers for sparse arrays. All unusual array usage is described below.

a. Arrays Generated in the Geometry Section

IBOXW - Subdivided box pattern for first planform, or both if "coplanar"

IBOXT - Subdivided box pattern for 2nd planform, if non-"coplanar"

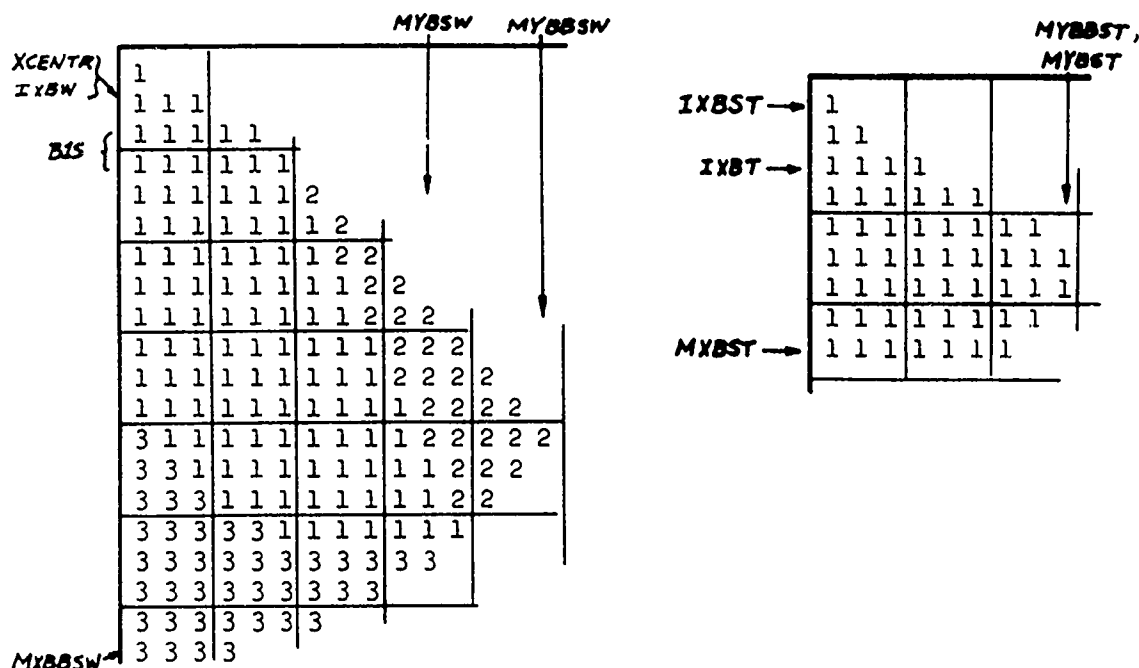
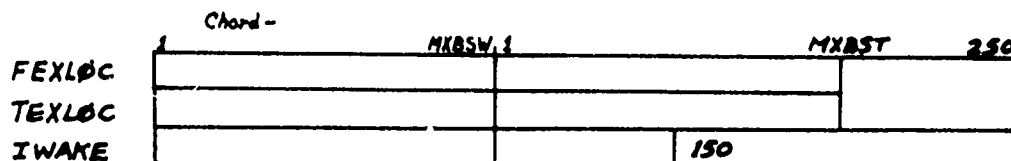


FIGURE 14. Box Code Arrays

The box code arrays are packed twenty numbers to a word, so IBOXW(1,1) contains codes for box (1,1) through box (1,20), IBOXW(2,1) contains codes for box (2,1) through box (2,20), etc.



FEXLOC(I) = The location of the leading edge at chord I, normalized to BIS with 1.0 corresponding to the center of the 1st (subdivided) row.

TEXLOC(I) = Same for trailing edge.

IWAKE(I) = Aftmost subdivided wing wake box needed by the tail.

FIGURE 15. Leading and Trailing Edge Arrays

	1	NWLE	10
XWLE			
YWLE		NTTE	
XWTE			
YWTE			NTLE
XTLE			
YTLE		NTTE	
XTTE			
YTTE			

Planform edge definition, Non-dimensionalized and shifted after reading.

FIGURE 16. Planform Edge Definitions

	1	2	3	...	...	...	...	...	...	NCHRDS
KPTWW										NWWK
KPTTT							NTTK			
KPTRWT									NRNTK	
KPTLWT									NLWTK	

For each AIC array needed:

											NROWS
MUAIC	1										
	2										

MUAIC(1,J) = first box needed in row j
MUAIC(2,J) = last box needed in row j

FIGURE 17. AIC Array Pointers

The four KPT-- arrays indicate the location on scratch file IATSSC of the desired AIC array set. For example, KPTT(3) is the AIC set number (4 matrices per set) of the AIC's for the influence of the left tail on right tail chord 3.

b. Arrays generated in the Modes Section

IPNTRM Pointer array for planform boxes on a row.

	J = 1 2 3 4				N N+1			
IPNTRM(1,J)	1	2	5	10		81	89	
IPNTRM(2,J)	1	1	1	1		3	0	

J Normally the row number for which the pointer value is being computed. If there are 2 surfaces that are noncoplanar, the value of J representing the first row of the second planform is MYBW+IOVLAP. IOVLAP is the number of rows on the tail planform that have same x coordinates as rows on the wing planform. If there are no rows with this condition IOVLAP is zero.

IPNTRM(1,J) The sequential count + 1 of all boxes, planform or wake region, that are on or between the first and last planform box of all rows forward of the one J represents

IPNTRM(2,J) The chord number of the first planform box on the row represented by J.

FIGURE 18. Row Pointers

c. Arrays Generated in the AIC Section

The $C_{D\mu\lambda}$, $W_{D\mu\lambda}$ and $V_{D\mu\lambda}$ arrays are stored in a one dimensional matrix. For planar AIC's the $W_{D\mu\lambda}$ and $V_{D\mu\lambda}$ are not computed and the $C_{D\mu\lambda}$ array is calculated for 1/2 of the Mach cone since it will be symmetrical. If subdivision is applied then 2 planar arrays are calculated with the subdivided array overlaying part of the un-subdivided array.

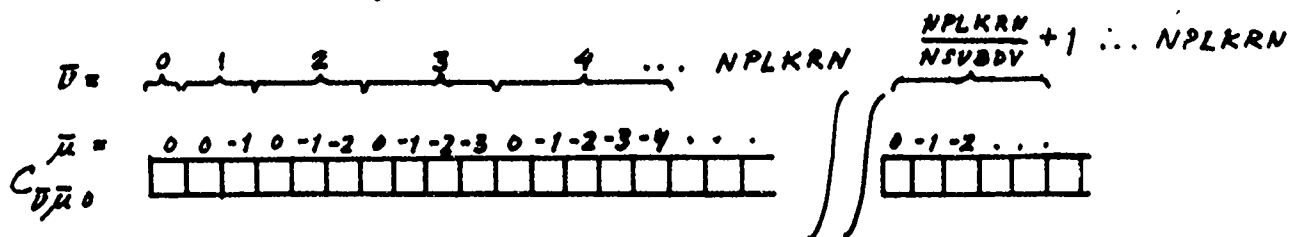


FIGURE 19. Planar AIC

For unsubdivided cases the array ends when $\bar{v} = \text{NPLKRN}$. For subdivided cases when $\bar{v} = \text{NPLKRN}$ the array contains the subdivided AIC calculated at $k_1 = k_1 / \text{NSUBDV}$. $\bar{v}$ then is reduced to $\frac{\text{NPLKRN}}{\text{NSUBDV}} + 1$ and is allowed to increase again until it reaches NPLKRN or the number of rows to cover the planform.

Because of the possible condition where the receiving point of a planform may not be in alignment with boxes on other planforms the spatial AIC's must be calculated on both sides.

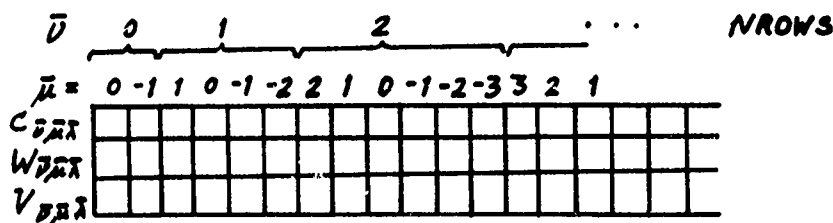


FIGURE 20. Spatial AIC's

13. INTERNAL SCRATCH FILES

a. Matrix Format

All arrays written on disk or tape files are formatted as variable sized matrices. Each matrix consists of two logical records of binary information, the first one being a 16-word matrix identification record, and the second containing the contents of the matrix. The reading/writing of these matrices is done by subroutines READMX and WRTEMX. In the following tape maps, each matrix is a separate box.

Matrix Identification Record - 16₁₀ words

Word	Contents
1	One-word ID label, an integer or label
2	M, the number of rows in the matrix
3	N, the number of columns in the matrix
4	Not used, = 0
5	Not used, = 0
6	Number of words in matrix record
7	K_1
8	Mach Number
9	} User parameters, array PARM
10	
11	
to	
16	

Matrix Record - variable length (word 6 above)

Ordered consecutively by row, left to right within each row.

b. Geometry Scratch File IGEO SC

This file is generated in the geometry processor and contains all large geometry arrays. The space after the two geometry files is used for temporary scratch during mode shape processing.

		Matrix Dimensions	Parameter Array
Present only if if NSURF=2 & COPLAN=.F.	IBOXW	M = MXBBW*NSUBDV N = (MYBBSW-1)/NBWRD+1	PARM(1) = 0. PARM(2) = XMACH
	IBOXT	M = MXEBST-IXBST+1 N = (MYBBST-1)/NBWRD+1	
	FEXLOC	M = 1 N = MYBSW+MYBST	
	TEXLOC	M = 1 N = MYBSW+MYBST	
Present only if NSPATK 0	ALPHA	M = 1 N = NAL	IPARM(3) = NALPHW
	IJALPH	M = 1 N = NAL	
	KPT	M = 1,2,3 or 4 N = max. # of AIC's needed in the 4 categories	IPARM(3) = NWWK IPARM(4) = NTTK IPARM(5) = NRWTK IPARM(6) = NLWTK
	EOF		
	MUAIC	M = 2 N = NROWS	PARM(4) = YBAR PARM(5) = EL IPARM(6) = 0; C,W,V needed
	MUAIC	M = 2 N = NROWS	1; W,V needed
			2; V needed
	MUAIC	M = 2 N = NROWS	
	EOF		

(File 2 is first built on IVPSC by GEOMBX, then copied to IGEO SC.)

c. Modes Scratch File MODESC

This file is generated in the modal data processor. The deflections and slopes are given at all box centers.

Matrix Dimensions

IPNTRM	M = 2 N = NPNTRS = $\begin{cases} \text{MXBW}+1 & \text{for single planform} \\ \text{MXBT}+\text{IOVLAP}+1 & \text{otherwise} \end{cases}$	IPARM(3) = 10VLA!
DEFSL Mode 1	M = 2 N = IPNTRM(1,NPNTRS)-1	
DEFSL Mode 2	M = 2 N = IPNTRM(1,NPNTRS)-1	
.		
.		
.		
.		
DEFSL Mode NMODES		
EOF		

d. Thickness Slopes Scratch File ITSLSC

This file is equivalenced to IWTFSC, which is first used in GEOMBX for temporary scratch while building the MUAIC arrays. The thickness slope functions are then written on the file at the end of the modal data processor. If $NTSLOP = 0$, one matrix of ones will be written, corresponding to $\frac{\partial z}{\partial x} = 0$.

Matrix Dimensions

EOF	TSLFN No. 1
	TSLFN No. 2
	TSLFN No. NTSLOP

M = 1
N = IPNTRM(1,NPNTRS)-1

e. Spatial AIC Scratch File IAICSC

This file is first used for internal scratch during calculation of polynomial coefficients by the modal data processor. In the AIC section it is written with all spatial AIC's needed for one reduced frequency, as determined in the geometry section. IAICSC is re-written for each new reduced frequency.

	Matrix Dimensions	Parameters
MUAIC ₁	M = 2 N = NROWS ₁	PARM(1) = k ₁ PARM(2) = XMACH ₁ PARM(4) = YBAR ₁ PARM(5) = EL ₁
C _{$\bar{v}\bar{\mu}\bar{\lambda}$}	M = 2 N = (NROWS ₁ +1)(NROWS ₁ /2)	
W _{$\bar{v}\bar{\mu}\bar{\lambda}$}		
V _{$\bar{v}\bar{\mu}\bar{\lambda}$}		
MUAIC ₂	M = 2 N = NROWS ₂	PARM(4) = YBAR ₁ PARM(5) = EL ₁
C _{$\bar{v}\bar{\mu}\bar{\lambda}$}	M = 2 N = (NROWS ₂ +1)(NROWS ₂ /2)	
W _{$\bar{v}\bar{\mu}\bar{\lambda}$}		
V _{$\bar{v}\bar{\mu}\bar{\lambda}$}		
MUAIC _{NSPATK}		
C _{$\bar{v}\bar{\mu}\bar{\lambda}$}		
W _{$\bar{v}\bar{\mu}\bar{\lambda}$}		
V _{$\bar{v}\bar{\mu}\bar{\lambda}$}		
EOF		

f. Velocity Potentials Scratch File IVPSC

This file is first used for internal scratch by the geometry processor while assembling MUAIC arrays. It is later used in the modal data processor as temporary storage for the wing mode shapes to be merged with the tail modes, and again for the same purpose when working with thickness slope functions. In the normal wash and velocity potentials section it is written with the $\Delta\phi$ and $\Delta\phi_{12}$ arrays for each mode.

<u>Matrix Dimensions</u>	
DELPHI ₁	M = 2 N = IPNTRM(1,NPNTRS)-1
TVP ₁	M = 2 N = { MYBSW if wing only MYBSW+MYBST otherwise
DELPHI ₂	
TVP ₂	
.	
.	
.	
.	
DELPHI _{NMODES}	
TVP _{NMODES}	

14. OUTPUT FILES

The program generates three optional output files. Two of them, the AIC files, are designed for reuse with the program during subsequent executions. The program automatically searches these files and updates them with any new AIC's generated.

The optional final output file is designed to pass the generalized air-forces matrices on for flutter or dynamic loads analyses. It is written optionally in the forces section of the program.

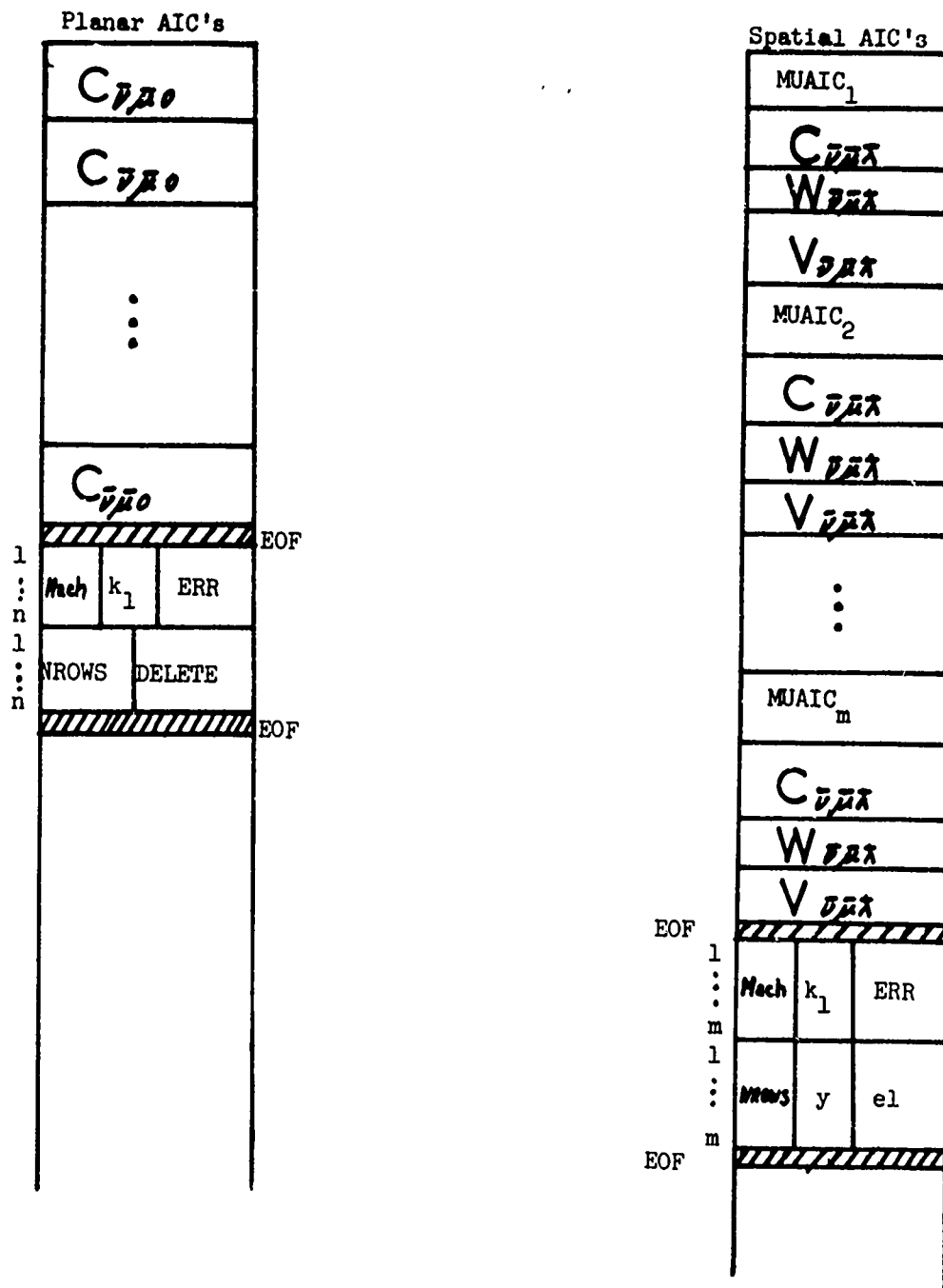


FIGURE 11. Tape Storage of AIC Arrays

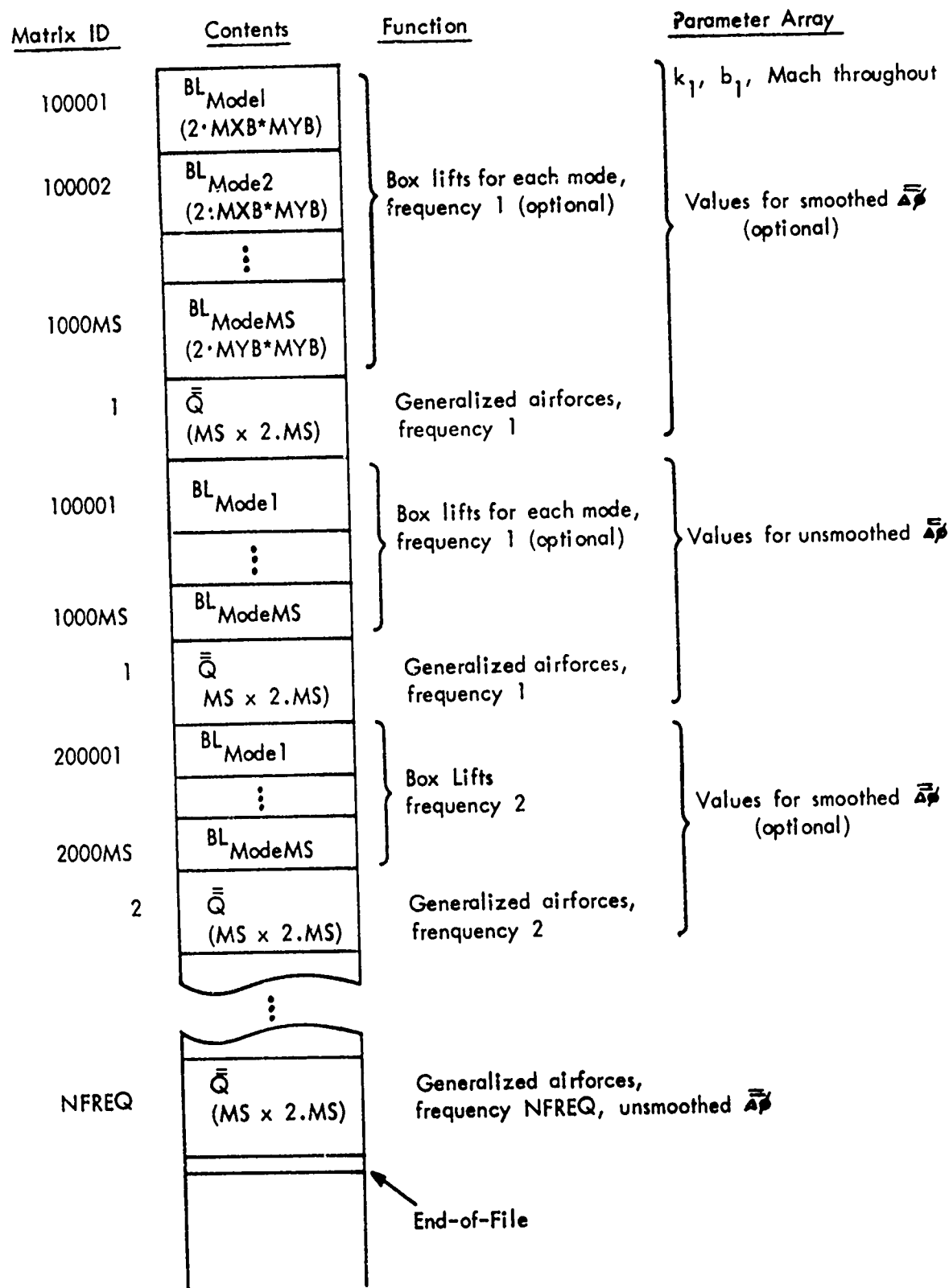


FIGURE 22. TAPE MAP OF FORCES OUTPUT TAPE

15. IMPLEMENTATION AND DEBUGGING

a. Update

The source program is written and maintained using the CDC 6600 SCOPE operating system UPDATE feature. Under UPDATE, all coding is either part of a *COMDECK or a *DECK. A *COMDECK may be replicated many times throughout the other decks. This feature is used for all global labeled common blocks and for most local common blocks, to insure that all routines needing them have identical common statements. A few subroutines which are needed in more than one overlay are also set up as *COMDECKs. The names of the *COMDECKs and *DECKs correspond as closely as possible to their Fortran identifiers - program name, subroutine name or common block name.

b. Open-ended Features

The writers of the program feel that most potential users probably have unique system features which may be utilized to optimize the execution of the program beyond its release status. With this in mind, numerous "hooks" have been coded in to make other features easy to implement.

1. All references to disk or tape files are by name, rather than by number. All file names are together in one common block, /FILES/. The internal scratch files are defined in one DATA statement in the zero overlay DRIVER, and the input, output and AIC files are defined via card input data.
2. All reading and writing of internal and external scratch files is handled by subroutines READMX and WRTEMX. These routines have several calling parameters which are unused, but available if it is desired to make use of labeling, random I.O., or level numbers. Because READMX and WRTEMX use BUFFERIN and BUFFEROUT, all files may share a common buffer area, allowing for a considerable savings in storage requirements.
3. Subroutine FLUSH is always called when a fatal error is encountered. This routine may be written to make use of any system error recovery procedure available. The release version prints a comment, flushes the OUTPUT file, and terminates with a Mode 1 error.
4. Subroutine DTIME is called between each secondary overlay. The release version returns CP time only; however, provision is made for PP time if the implementing system has that capability.

c. Debugging

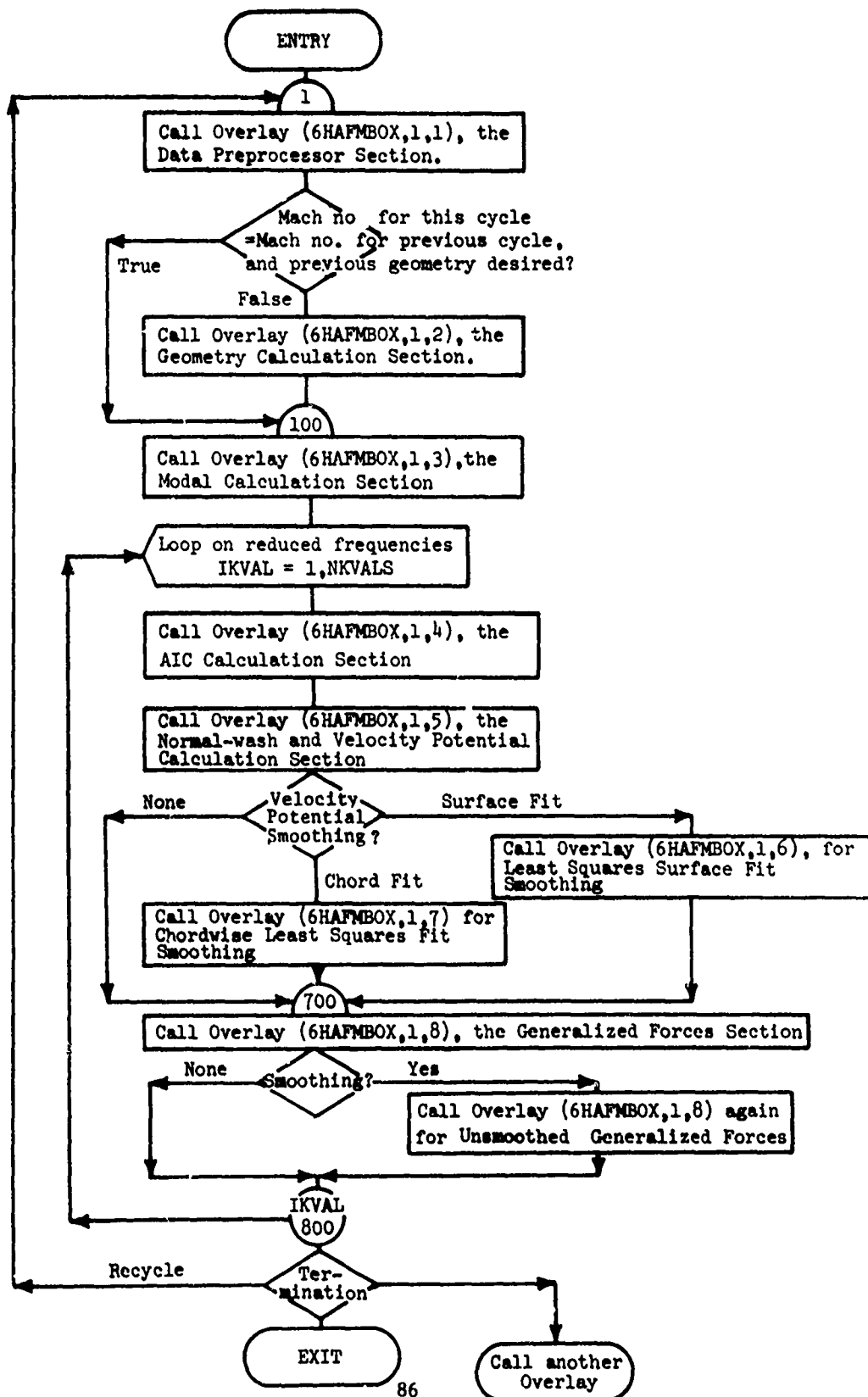
It is recommended that a new user first run one of the sample data cases, to familiarize himself with the program features and to insure

that the program gives correct answers at his installation.

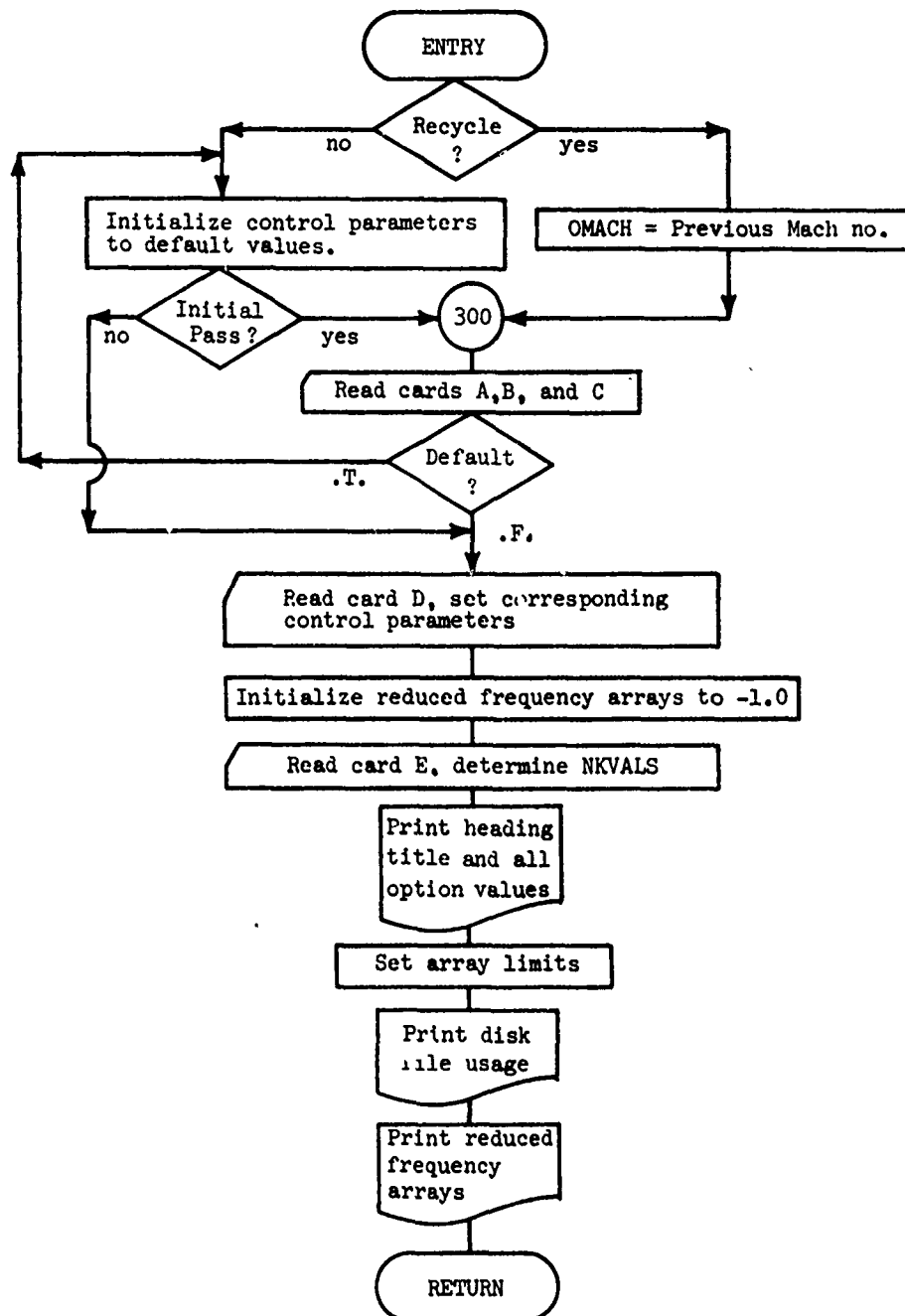
In the event the program fails "hard" (mode error, time limit, etc.), standard use of listings, load maps and dumps will usually pinpoint the cause.

If the program executes but seems to give bad numbers, additional intermediate printout may be helpful. The variables in common block /CHECKPR/ are designed to control the printing of additional check values. Each variable controls printing from one secondary overlay, so only the suspected area need be printed. The check prints provided are rudimentary, so for given problems additional prints would probably have to be written, but if they are made conditional on the common variables, they can be left in for future needs. The CHECKPR variables are all read from Card C of the data, or may be set in an executable statement after the call to DATAPP.

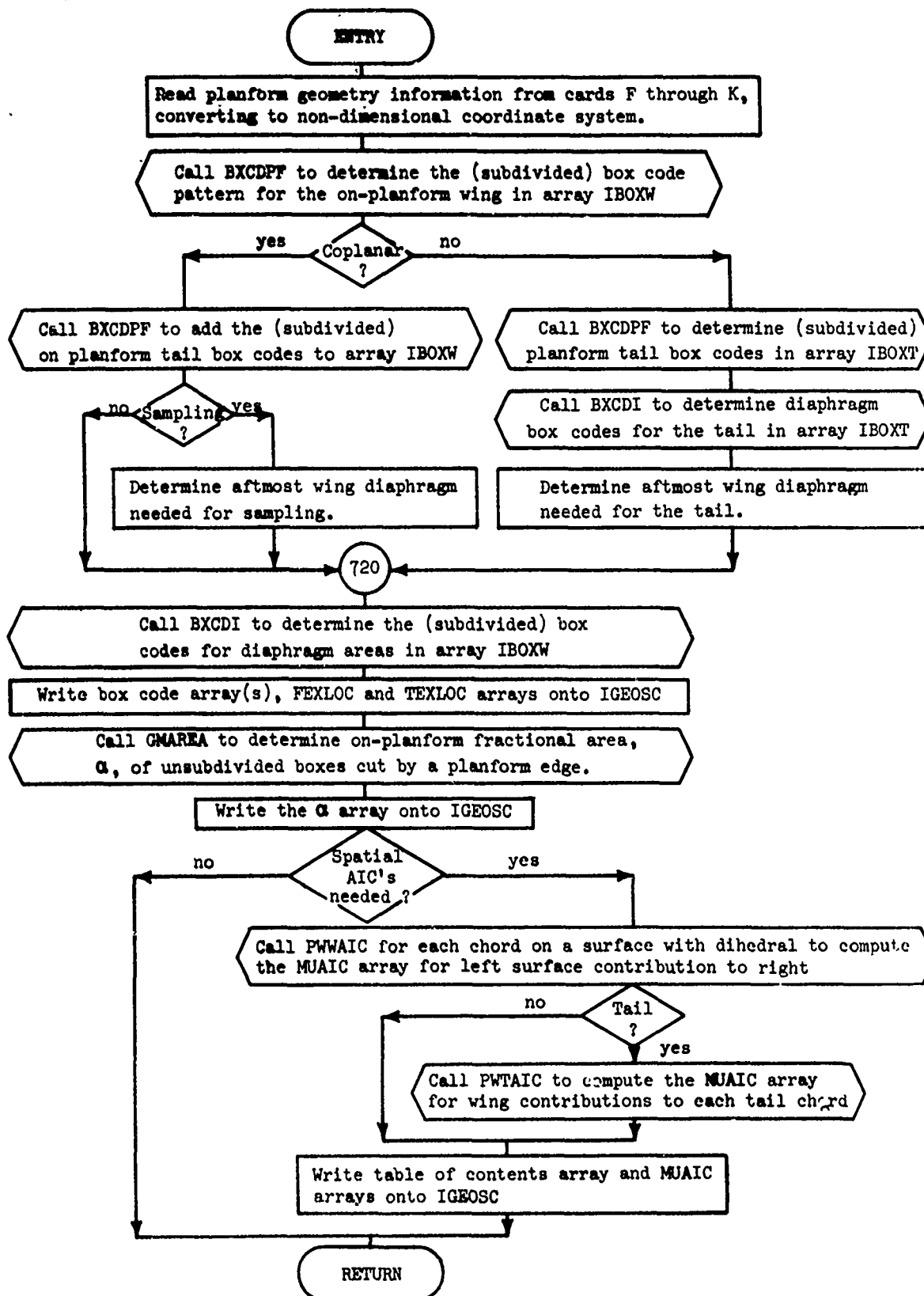
Program CONTROL — Primary level overlay which controls the program flow



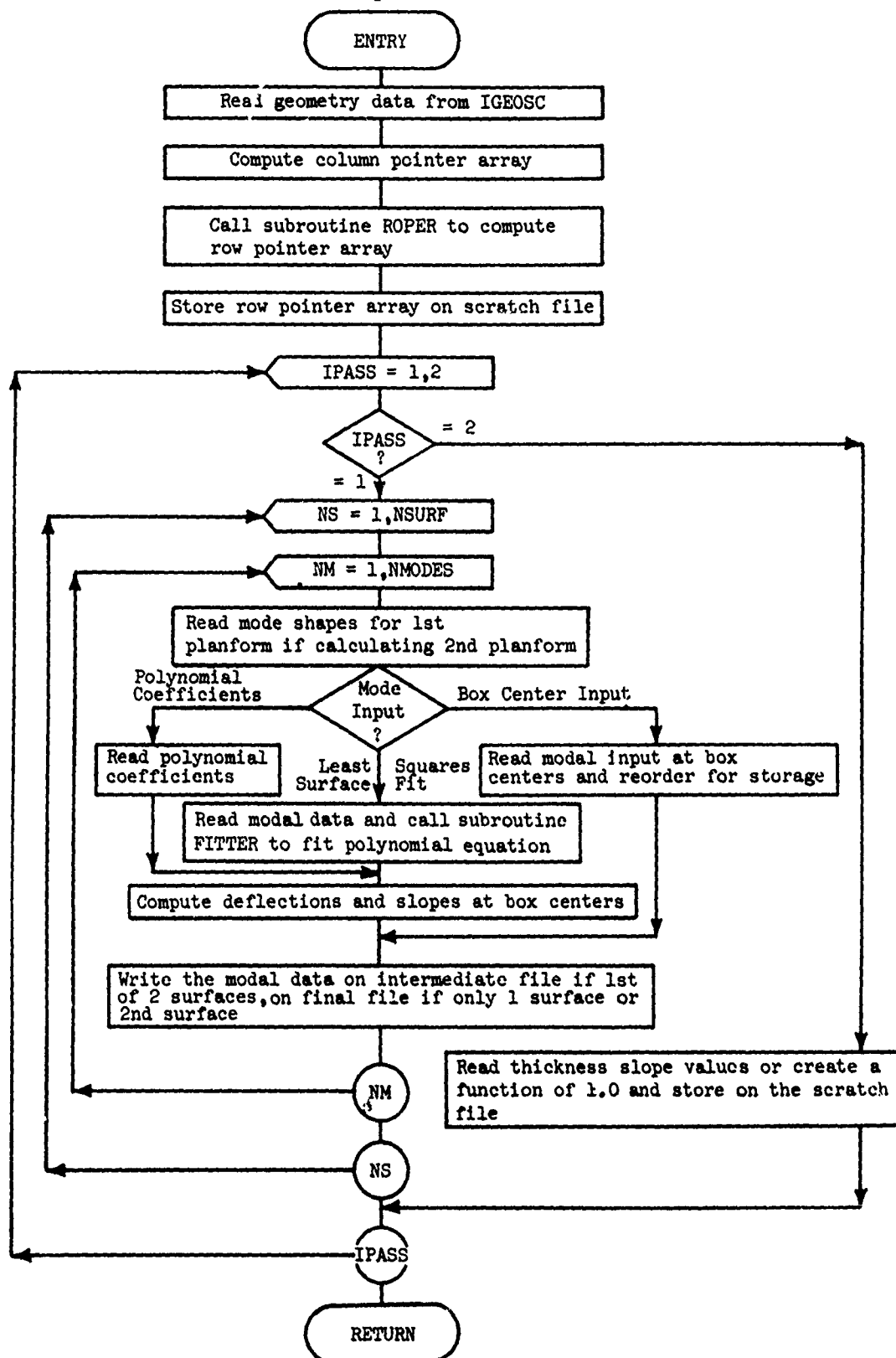
Program DATAPP — Secondary overlay which initiates control parameters as a function of defaults or card data.



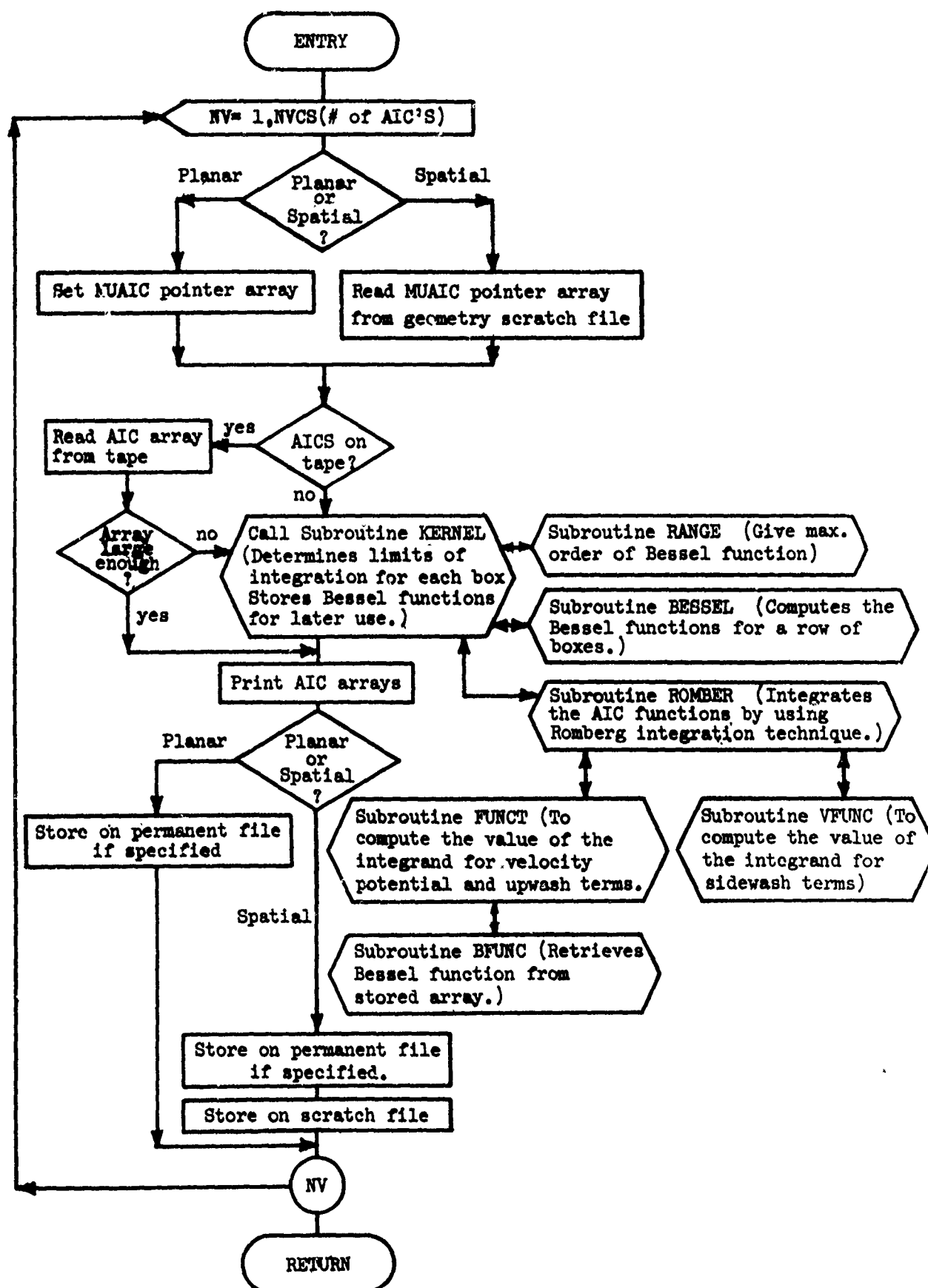
Program GEOSK — Secondary level overlay which computes planform geometry.



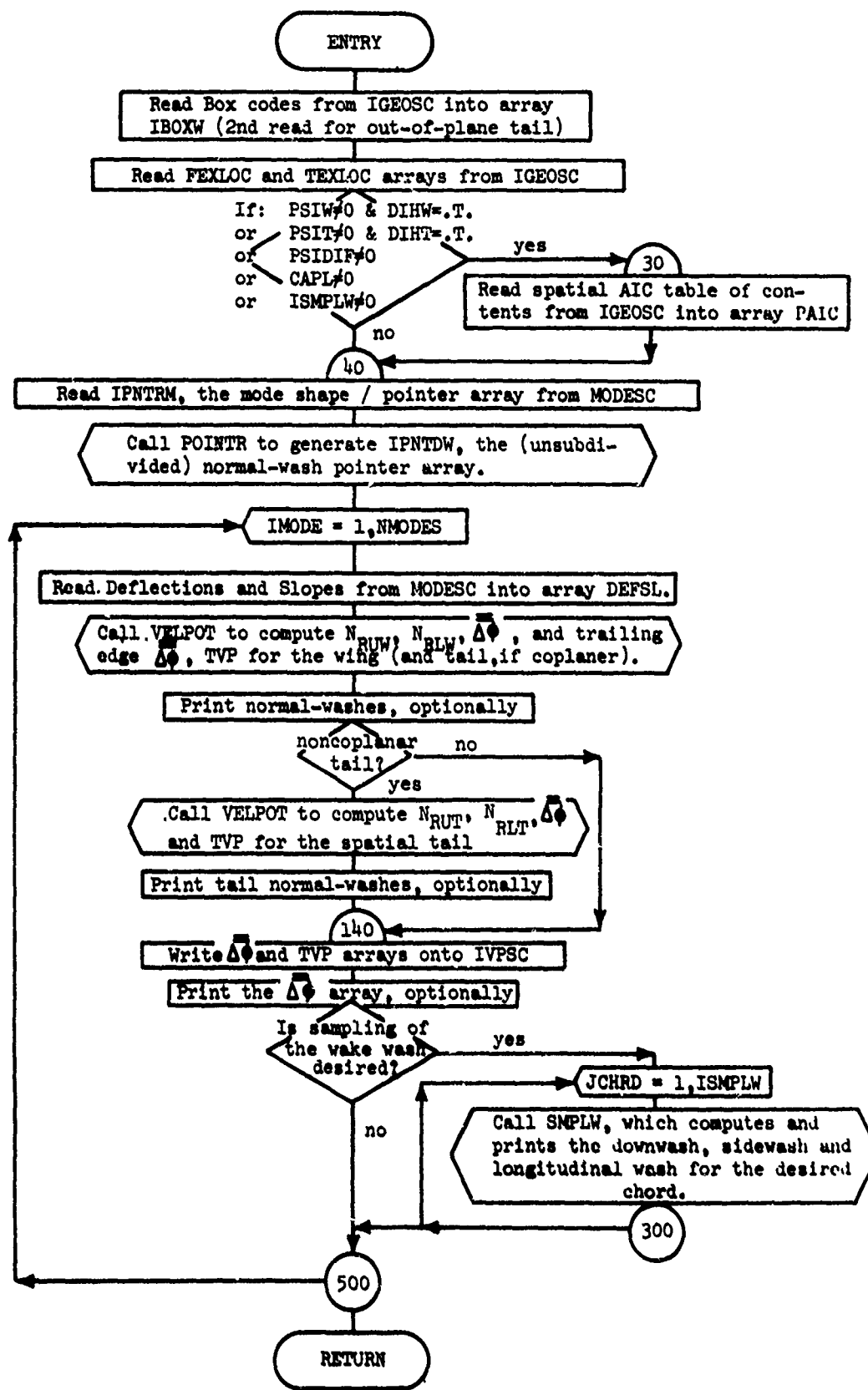
Program MODES— Secondary level overlay which reads mode shapes and thickness slope functions



Program VICMAIN — Secondary level overlay which calculates all
Aerodynamic Influence Coefficients

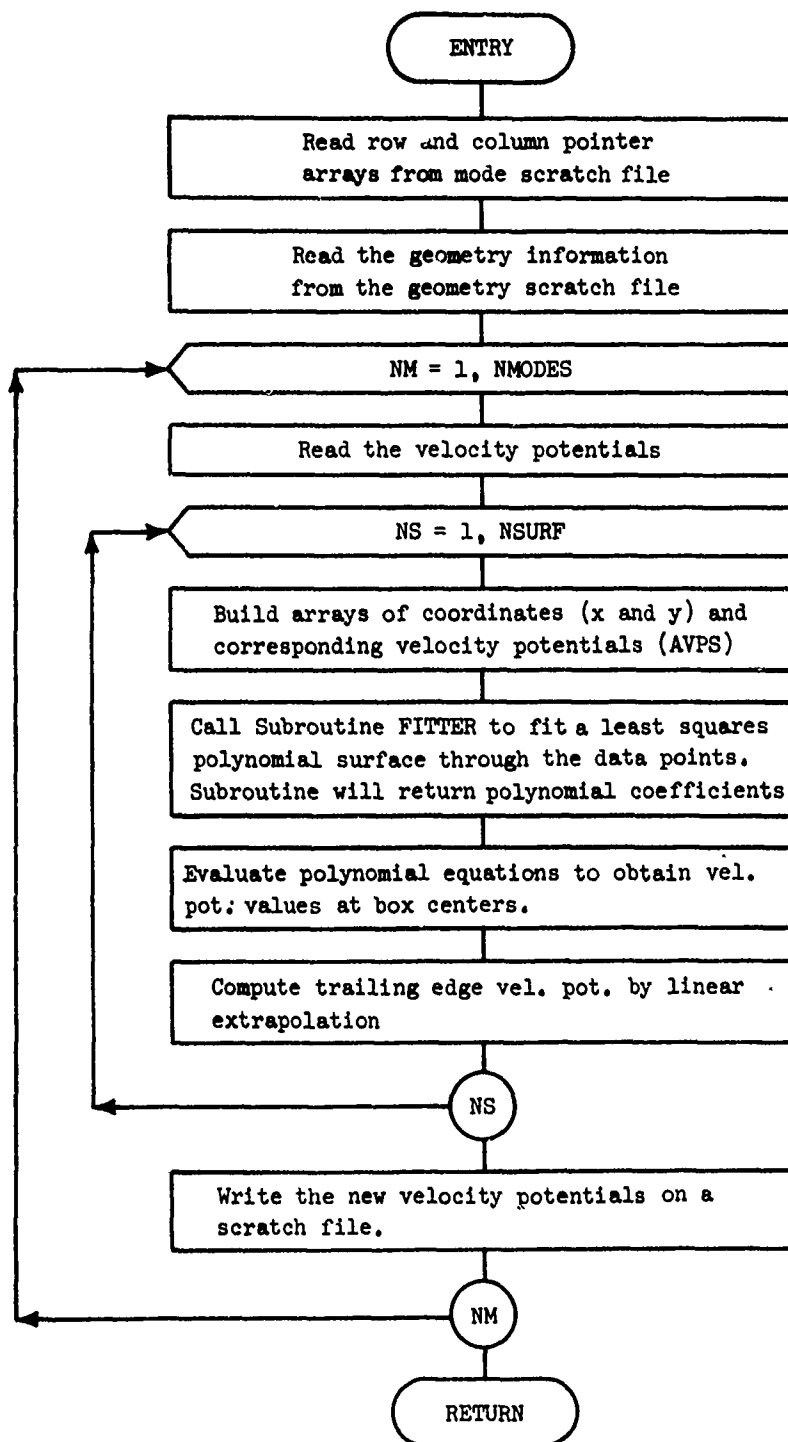


Program NWVirt Secondary overlay which calculates normal washes and velocity potentials.

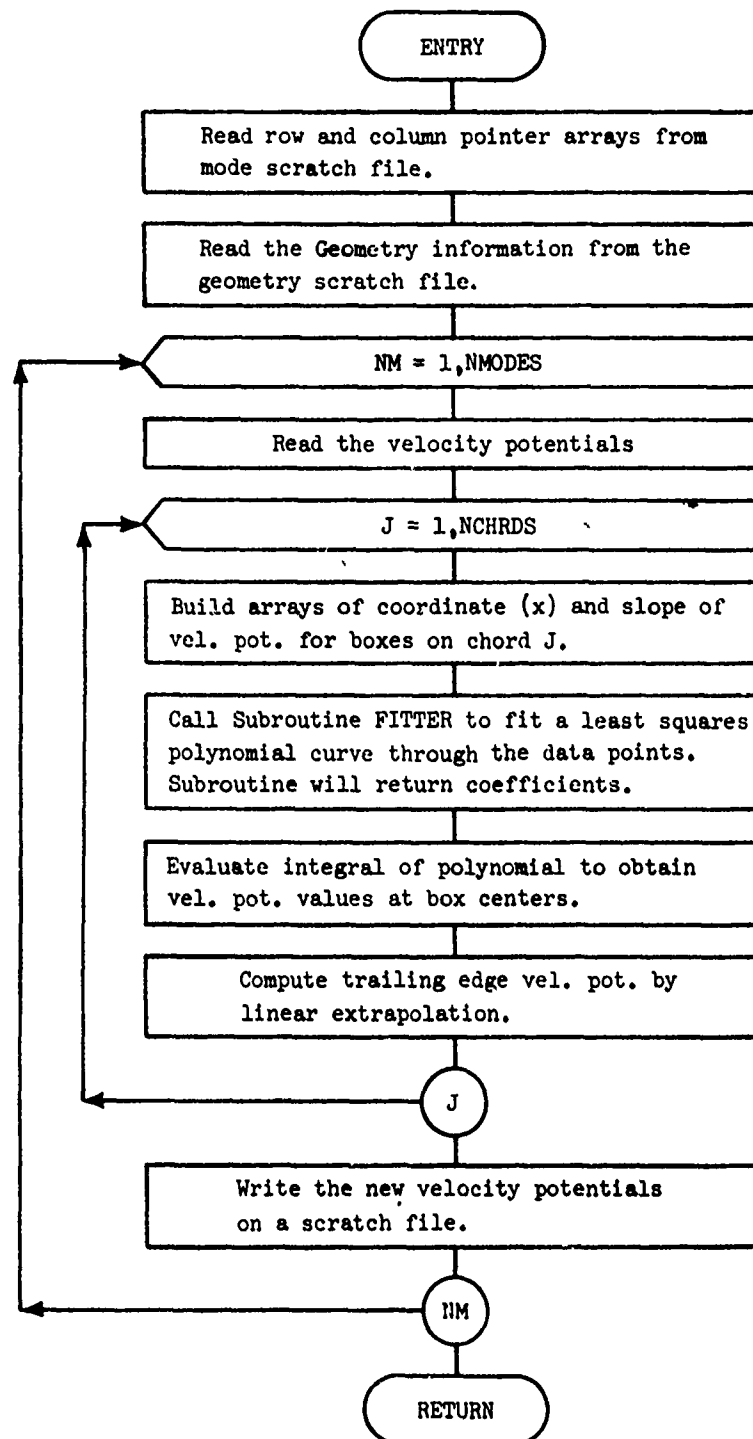


Program SMTH

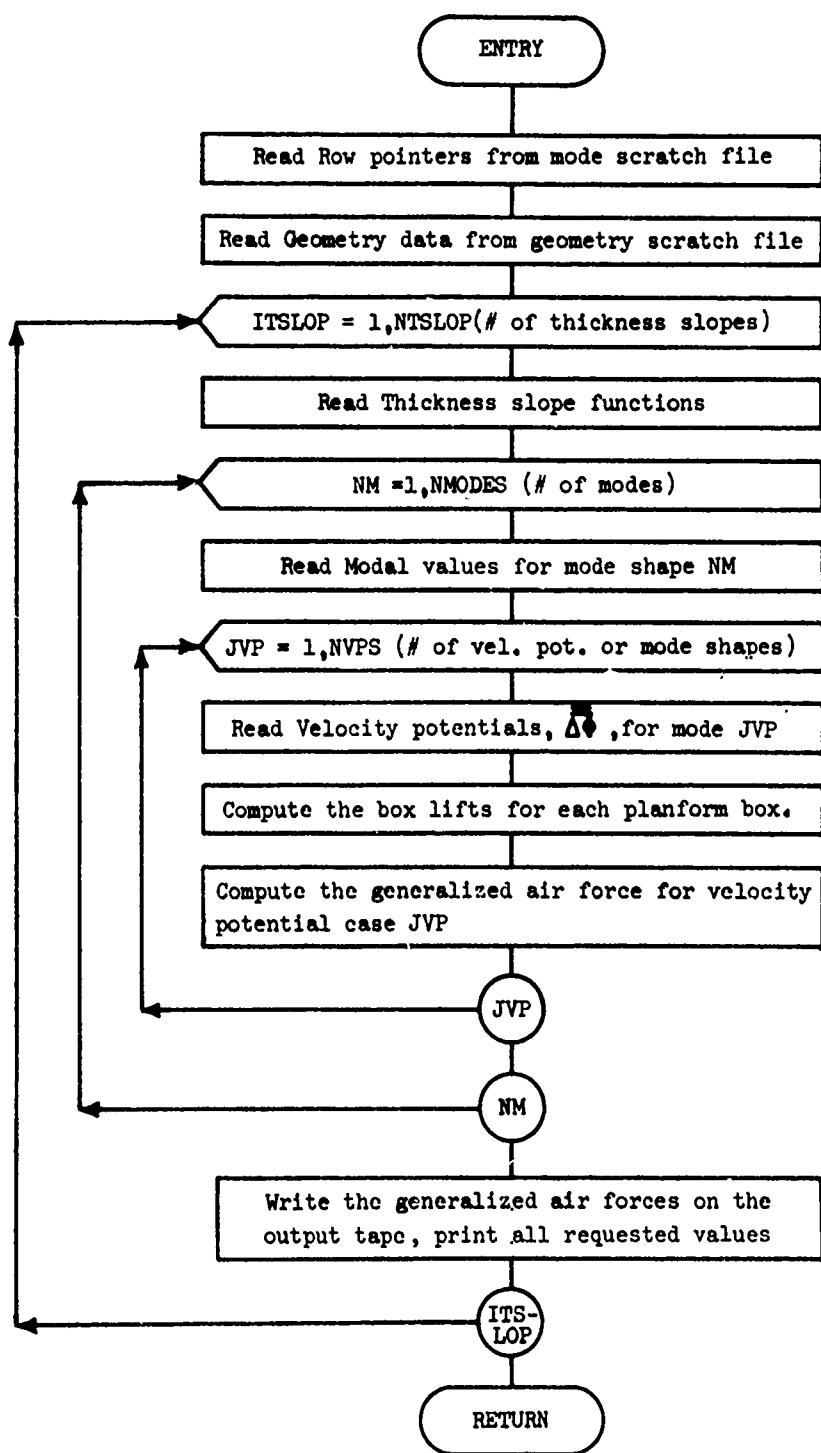
Secondary overlay which smooths velocity potentials
by fitting a least squares surface through them



Program CHORDF — Secondary overlay which smooths velocity potentials by fitting a least squares curve along each chord.



Program FORCES Secondary overlay which computes box lifts
section lifts, and generalized air forces



REFERENCES

1. Faddeeva, V. N., Computational Methods of Linear Algebra, Dover Publications, Inc., 1959.
2. Ralston, A., And Wilf, H., Mathematical Methods for Digital Computers, Vol. 2, New York, John Wiley and Sons, 1967, pp. 133-137.

APPENDIX A

SAMPLE INPUT AND OUTPUT DATA

A simple spatial configuration, shown in Figure 23, was chosen as a sample problem for the demonstration of the card data input and a selection of the printed output. The planform used is a pair of identical rectangular surfaces (wing and tail) with small horizontal and vertical separation.

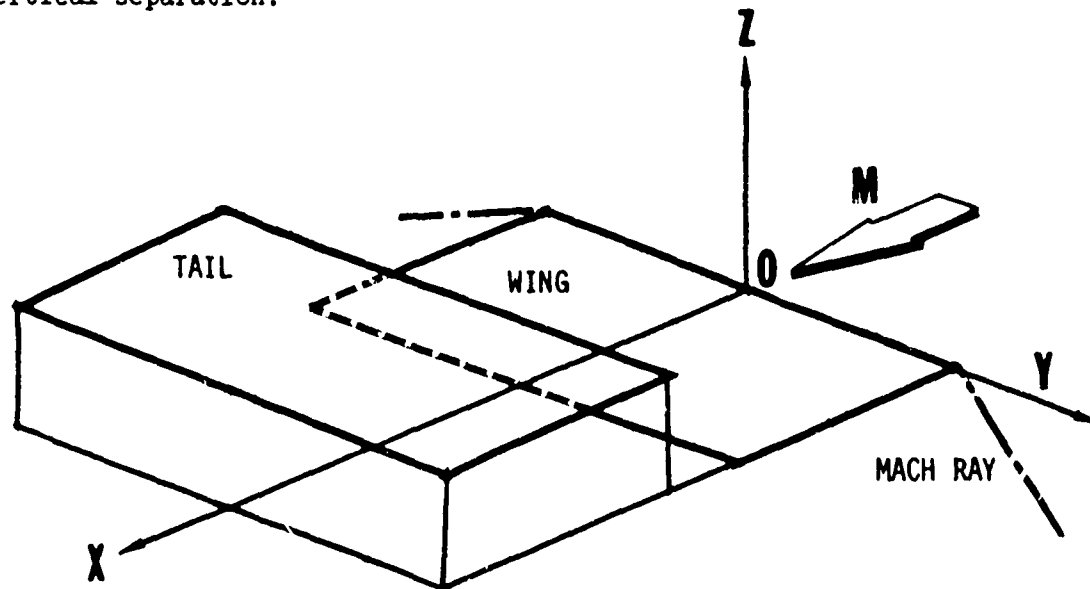


FIGURE 23 SAMPLE PROBLEM CONFIGURATION

The configuration was analyzed at Mach 1.2 for a reduced frequency (based on semi-span) of .5. Only the wing surface was allowed to oscillate, in plunging motion for mode 1 and in pitch about the wing leading edge for mode 2. These two modes were input on cards as polynomials. Chordwise velocity potential smoothing was requested.

In the interest of space the printout was edited to give samples only. A few pages of one spatial AIC array and the planar AIC are included, as well as most of the computations for mode 2 (wing pitch). Since for this configuration the upper and lower surface normal wash differs only in sign, only the upper normal washes are included. The generalized force calculations at the end are for smoothed velocity potentials.

Card Input Data

```

AFMDOX      1      0
SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
$CARDD XMACH=1.20      $
$CARDC SYM=1.0,MTYPEW=1,MTYPEF=1,NSURF=2,WTGNF=.T.,WTDL=.T.,
PRGNAC=.T., PRDCF=.T.,
PLYWOOD = .T.,
CRDFIT=.T.,NDEG=4,
PRGNAF=.T.,PRDL=.T.,PRSL=.T.,PRPAIC=.T.,PRSAIC=.T.,PRCOEF=.T.,
PRDOX=.T.,PRVF=.T.,EXAIL=.F.,FRNW=.T., PRMODS=.T. $
$CARDD      $
$CARDE XKS(1) =      .50      $
$CARDF TLAX=1.20, TLAZ=.40 $
$CARDG NCHRDS=10, XEDGE=0.00      $
  2      2      2      2
0.      0.      0.      1.0
1.0      0.0      1.0      1.0
0.      0.      0.      1.0
1.0      0.0      1.0      1.0
$CARDH NMDES=2 $
  2
1.      0.      0.      0.      0.      0.
  2
0.      1.      0.      0.      0.      0.
  1
0.      0.      0.
  1
0.      0.      0.

```

CARD H
 CARD I
 CARD J
 CARD K
 CARD L

 WING-1
 WING-2

 TAIL-1
 TAIL-2


```

*****
*
* UNSTEADY AERODYNAMICS OF WING-HORIZONTAL TAIL
* CONFIGURATIONS IN SUPERSONIC FLOW
*
*
* PREPARED UNDER CONTRACT NO. AF 33615-70-C-1126
* PROJECT NO. 1370
*
* FOR DEPARTMENT OF THE AIR FORCE
* AERONAUTICAL SYSTEMS DIVISION
* AIR FORCE FLIGHT DYNAMICS LABORATORY
* WRIGHT-PATTERSON AIR FORCE BASE
*
* BY THE BOEING COMPANY
* COMMERCIAL AIRPLANE DIVISION
* SEATTLE, WASHINGTON
*
*****

```

- TITLE

SAMPLE CASE --- TWO AR-2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

TITLE -

A 3

THE FOLLOWING OPTIONS ARE REQUESTED -

```

MACH NUMBER = 1.200000
SYMMETRIC ANALYSIS
FLYMODE OPTION IS USED. (PLATFORM BOUNDARY DETERMINED BY DOX PATTERN.)
BASIC (UNSUBDIVIDED) ANALYSIS WILL BE USED
ANALYSIS FOR 2 PLATFORMS
APPROXIMATE KERNELS WILL BE USED
VELOCITY POTENTIALS WILL BE SMOOTHED BY A LEAST SQUARES
POLYNOMIAL CHORDWISE FIT, OF ORDER 4.
(0 = PROGRAM DETERMINED.)

PRINT THE DOX PATTERN
PRINT WIDE SHAPE POLYNOMIAL COEFFICIENTS, IF AVAILABLE
PRINT WIDE SHAPES USED
PRINT THE FLANAR AIC ARRAYS USED
PRINT THE SPATIAL AIC ARRAYS USED
PRINT NORMAL WASHES
PRINT 1/C VELOCITY POTENTIALS
PRINT THE DOX LIFTS
PRINT THE SECTION LIFTS
PRINT PRESSURE DIFFERENCE COEFFICIENTS

```

PRINT GENERALIZED AERODYNAMIC COEFFICIENTS
 PRINT GENERALIZED AIR FORCES
 WRITE BOX LIFTS ON TAPE
 WRITE GENERALIZED AIR FORCES ON TAPE
 MODAL INPUT FOR WING IS POLYNOMIAL COEFFICIENTS
 MODAL INPUT FOR TAIL IS POLYNOMIAL COEFFICIENTS
 DIHEDRAL WING INFLUENCE CALCULATED
 DIHEDRAL TAIL INFLUENCE CALCULATED

THE FOLLOWING TAPE SETUP IS REQUESTED -

OLD AIC TAPE = 0
 NEW AIC TAPE = 0
 OLD SPATIAL AIC TAPE = 0
 NEW SPATIAL AIC TAPE = 0
 INPUT DATA TAPE = 0 SPACED 0 FILES,
 OUTPUT TAPE = 1 SPACED 0 FILES,

A 4

THE FOLLOWING IS THE REDUCED FREQUENCY ARRAY BASED ON WING SEMI-SPAN

.50000

ENTERING PROGRAM GEOM CURRENT ELAPSED TIME IS CP = .224, FP = 37.757

----- GEOMETRIC PARAMETERS -----

CARD#	LOCAL AXES DEFINITION-	X-LOCATION	Z-LOCATION	DIHEDRAL ANGLE (FST)
WING		0.000	0.000	0.00 DEGREES
TAIL		1.200	.400	0.00 DEGREES

CARD#	BOX PATTERN DEFINITION-	NCHRS	XCENTR	OR	XEDGE
		10	#0000.0000		0.0000

CARD#	FLANFORM DEFINITION POINT COUNTS-	LEADING EDGE	TRAILING EDGE
	WING	2	2
	TAIL	2	2

CARD#	TO CARCL FLANFORM DEFINITIONS-	X	Y	(LOCAL AXES)
	WING L.E.	0.000	0.000	
		0.000	1.000	

WING T.C.	1.000	0.000
	1.000	1.000
TAIL L.E.	0.000	0.000
	0.000	1.000
TAIL T.E.	1.000	0.000
	1.000	1.000

-BOX DIMENSIONS- D1 (LENGTH) = 6.63324958E-02 D1/BETA (WIDTH) = 1.00000000E-01

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

BOX CODE PATTERN

MACH 1.2000000

CODE - 1 = PLATFORM BOX
2 = DIAPHRAGM BOX
3 = WAKE BOX

	2	4	6	8	10	12	14	16
19	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	2	
22	1	1	1	1	1	1	2	2
23	1	1	1	1	1	1	2	2
24	1	1	1	1	1	1	2	2
25	1	1	1	1	1	1	2	2
26	1	1	1	1	1	1	2	2
27	1	1	1	1	1	1	2	2
28	1	1	1	1	1	1	2	2
29	1	1	1	1	1	1	2	2
30	1	1	1	1	1	1	2	2
31	1	1	1	1	1	1	2	2
32	1	1	1	1	1	1	2	
33	1	1	1	1	1	1	1	1

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

BOX CODE PATTERN
MACH 1.2000000

CODE - 1 = PLANFORM BOX
2 = DIAPHRAGM BOX
3 = WAKE BOX

	2	4	6	8	10	12	14	16	18	20	22	24	26
1	1	1	1	1	1	1	1						
2	1	1	1	1	1	1	1	2					
3	1	1	1	1	1	1	1	2	2				
4	1	1	1	1	1	1	1	2	2	2			
5	1	1	1	1	1	1	1	2	2	2	2		
6	1	1	1	1	1	1	1	2	2	2	2	2	
7	1	1	1	1	1	1	1	2	2	2	2	2	2
8	1	1	1	1	1	1	1	2	2	2	2	2	2
9	1	1	1	1	1	1	1	2	2	2	2	2	2
10	1	1	1	1	1	1	1	2	2	2	2	2	2
11	1	1	1	1	1	1	1	2	2	2	2	2	2
12	1	1	1	1	1	1	1	2	2	2	2	2	2
13	1	1	1	1	1	1	1	2	2	2	2	2	2
14	1	1	1	1	1	1	1	2	2	2	2	2	2
15	1	1	1	1	1	1	1	2	2	2	2	2	2
16	3	3	3	3	3	3	3	2	2	2	2	2	2
17	3	3	3	3	3	3	3	2	2	2	2	2	2
18	3	3	3	3	3	3	3	2	2	2	2	2	2
19	3	3	3	3	3	3	3	2	2	2	2	2	2
20	3	3	3	3	3	3	3	2	2	2	2	2	2
21	3	3	3	3	3	3	3	2	2	2	2	2	2
22	3	3	3	3	3	3	3	2	2	2	2	2	2
23	3	3	3	3	3	3	3	2	2	2	2	2	2
24	3	3	3	3	3	3	3	2	2	2	2	2	2
25	3	3	3	3	3	3	3	2	2	2	2	2	2
26	3	3	3	3	3	3	3	2	2	2	2	2	2
27	3	3	3	3	3	3	3	2	2	2	2	2	2
28	3	3	3	3	3	3	3	2	2	2	2	2	2
29	3	3	3	3	3	3	3	2	2	2	2	2	2

ENTERING PROGRAM HOSES CURRENT ELAPSED TIME IS CP = 1.894, FP = 40.123

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

MODE SHAPE NUMBER 2
MACH NUMBER = 1.200000

FROM CARD INPUT

MODAL POLYNOMIAL COEFFICIENTS

CONSTANT

0.

X ##1Y ##0 X ##0Y ##1

1.0200E+00 0.

X ##2Y ##0 X ##1Y ##1 X ##0Y ##2

0.

MODAL POLYNOMIAL COEFFICIENTS

FROM CARD INPUT

CONSTANT

0.

X ##1Y ##0 X ##0Y ##1

0.

DEFLECTIONS X 1.0E 1

	1	2	3	4	5	6	7	8	9	10
1	.332	.332	.332	.332	.332	.332	.332	.332	.332	.332
2	.995	.995	.995	.995	.995	.995	.995	.995	.995	.995
3	1.658	1.658	1.658	1.658	1.658	1.658	1.658	1.658	1.658	1.658
4	2.322	2.322	2.322	2.322	2.322	2.322	2.322	2.322	2.322	2.322
5	2.985	2.985	2.985	2.985	2.985	2.985	2.985	2.985	2.985	2.985
6	3.648	3.648	3.648	3.648	3.648	3.648	3.648	3.648	3.648	3.648
7	4.312	4.312	4.312	4.312	4.312	4.312	4.312	4.312	4.312	4.312
8	4.975	4.975	4.975	4.975	4.975	4.975	4.975	4.975	4.975	4.975
9	5.638	5.638	5.638	5.638	5.638	5.638	5.638	5.638	5.638	5.638

ENTERING PROGRAM AIC CURRENT ELAPSED TIME IS CF = 2.555, FP = 44.910

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

AIC CALCULATIONS

MACH = 1.20000 K1 = .0331662 ERR = 1.00000E-02 EL = 4.00 YBAR = 0.00

NU	MU	C			W			V		
		VELOCITY	POTENTIAL	COEFFICIENT	UPWASH	COEFFICIENT	SIDEMASH	COEFFICIENT	COEFFICIENT	COEFFICIENT
0	0	0.	0.	0.	0.	0.	0.	0.	0.	0.
1	1	0.	0.	0.	0.	0.	0.	0.	0.	0.
1	0	0.	0.	0.	0.	0.	0.	0.	0.	0.
1	-1	0.	0.	0.	0.	0.	0.	0.	0.	0.
2	2	0.	0.	0.	0.	0.	0.	0.	0.	0.
2	1	0.	0.	0.	0.	0.	0.	0.	0.	0.
2	0	0.	0.	0.	0.	0.	0.	0.	0.	0.
2	-1	0.	0.	0.	0.	0.	0.	0.	0.	0.
2	-2	0.	0.	0.	0.	0.	0.	0.	0.	0.
3	3	0.	0.	0.	0.	0.	0.	0.	0.	0.
3	2	0.	0.	0.	0.	0.	0.	0.	0.	0.
3	1	0.	0.	0.	0.	0.	0.	0.	0.	0.
3	0	0.	0.	0.	0.	0.	0.	0.	0.	0.
3	-1	0.	0.	0.	0.	0.	0.	0.	0.	0.
3	-2	0.	0.	0.	0.	0.	0.	0.	0.	0.
3	-3	0.	0.	0.	0.	0.	0.	0.	0.	0.
4	4	0.	0.	0.	0.	0.	0.	0.	0.	0.
4	3	0.	0.	0.	0.	0.	0.	0.	0.	0.
4	2	-3.47636677E-02	1.80224368E-02	1.97268081E-01	-1.00363164E-01	9.19532598E-02	-4.69384667E-02	-4.69384667E-02	-4.69384667E-02	-4.69384667E-02
4	1	-1.17760029E-01	5.85388468E-02	1.72703458E-01	-7.56330976E-02	4.39228951E-02	-1.94380883E-02	-1.94380883E-02	-1.94380883E-02	-1.94380883E-02
4	0	-1.39653064E-01	6.82985240E-02	1.59648216E-01	-6.50348287E-02	0.	0.	0.	0.	0.
4	-1	-1.17760029E-01	5.85388468E-02	1.72703458E-01	-7.56330976E-02	-4.39228951E-02	1.94380883E-02	1.94380883E-02	1.94380883E-02	1.94380883E-02
4	-2	-3.47636677E-02	1.80224368E-02	1.97268081E-01	-1.00363164E-01	-9.19532598E-02	4.69384667E-02	4.69384667E-02	4.69384667E-02	4.69384667E-02
4	-3	0.	0.	0.	0.	0.	0.	0.	0.	0.
4	-4	0.	0.	0.	0.	0.	0.	0.	0.	0.
5	5	0.	0.	0.	0.	0.	0.	0.	0.	0.
5	4	-1.26580111E-02	8.45618922E-03	7.57086924E-02	-4.99682786E-02	6.96685870E-02	-4.60082949E-02	-4.60082949E-02	-4.60082949E-02	-4.60082949E-02
5	3	-1.1608951E-01	7.25678775E-02	1.10641214E-01	-6.02286065E-02	8.38256078E-02	-4.59949061E-02	-4.59949061E-02	-4.59949061E-02	-4.59949061E-02
5	2	-1.45526451E-01	8.55703207E-02	-9.83107500E-02	5.93665069E-02	-4.25224207E-02	2.62689346E-02	2.62689346E-02	2.62689346E-02	2.62689346E-02
5	1	-9.96538896E-02	5.92204335E-02	-6.88337551E-02	3.97750794E-02	-1.80453789E-02	1.04152589E-02	1.04152589E-02	1.04152589E-02	1.04152589E-02
5	0	-9.18644239E-02	5.46812863E-02	-5.23885439E-02	3.04815076E-02	0.	0.	0.	0.	0.
5	-1	-9.96538896E-02	5.92204335E-02	-6.88337551E-02	3.97750794E-02	1.80453789E-02	-1.04152589E-02	-1.04152589E-02	-1.04152589E-02	-1.04152589E-02

5	-2	-1.45526451E-01	8.55703207E-02	-9.83107500E-02	5.93665069E-02	4.25224207E-02	-2.62689346E-02
5	-3	-1.16088951E-01	7.25678775E-02	1.10641214E-01	-6.10228606E-02	-8.38256078E-02	4.59949061E-02
5	-4	-1.26580111E-02	8.45618922E-03	7.57086924E-02	-4.99682786E-02	-6.96685870E-02	1.60082949E-02
5	-5	0.	0.	0.	0.	0.	0.
6	6	0.	0.	0.	0.	0.	0.
6	5	-3.99220293E-02	3.25890518E-02	7.99613225E-02	-6.33037310E-02	9.80050144E-02	-7.77761878E-02
6	4	-1.21957220E-01	9.22166540E-02	-6.92810630E-03	9.28625574E-03	-7.56361388E-04	4.98241475E-03
6	3	-7.78882128E-02	5.86123874E-02	-4.00640646E-02	2.94590784E-02	-3.10389954E-02	2.28078724E-02
6	2	-6.07096937E-02	4.58905554E-02	-1.87613792E-02	1.39987792E-02	-9.54362988E-03	7.11917814E-03
6	1	-5.45262136E-02	4.12685264E-02	-1.40850668E-02	1.05505943E-02	-3.56932071E-03	2.67317908E-03
6	0	-5.28116078E-02	3.99887087E-02	-1.29745459E-02	9.72827152E-03	0.	0.
6	-1	-5.45232136E-02	4.12685264E-02	-1.40850668E-02	1.05505943E-02	3.56932071E-03	-2.67317908E-03
6	-2	-6.07096937E-02	4.58905554E-02	-1.87613792E-02	1.39987792E-02	9.54362988E-03	-7.11917814E-03
6	-3	-7.78882128E-02	5.86123874E-02	-4.00640646E-02	2.94590784E-02	-3.10389954E-02	-2.28078724E-02
6	-4	-1.21957220E-01	9.22166540E-02	-6.92810630E-03	9.28625574E-03	7.56361388E-04	-4.98241475E-03
6	-5	-3.99220293E-02	3.25890518E-02	7.99613225E-02	-6.33037310E-02	-9.80050144E-02	7.77761878E-02
6	-6	0.	0.	0.	0.	0.	0.
7	7	0.	0.	0.	0.	0.	0.
7	6	-5.56430474E-02	5.54322101E-02	6.75641236E-02	-6.43952032E-02	1.02001591E-01	-9.75864803E-02
7	5	-9.31170389E-02	8.69860745E-02	-3.18547641E-02	3.11225707E-02	-3.78738988E-02	3.73390401E-02
7	4	-5.41942395E-02	5.10205307E-02	-1.66531945E-02	1.54504087E-02	-1.69471710E-02	1.57190079E-02
7	3	-4.33217879E-02	4.08912504E-02	-9.2000806E-03	8.60658871E-03	-6.97408521E-03	6.52337914E-03
7	2	-3.83083684E-02	3.61961147E-02	-6.86176794E-03	6.43842195E-03	-3.45857670E-03	3.24490770E-03
7	1	-3.58922648E-02	3.39267375E-02	-5.92213712E-03	5.56399165E-03	-1.49063517E-03	1.40038141E-03
7	0	-3.51605591E-02	3.32385738E-02	-5.65911048E-03	5.31886725E-03	0.	0.
7	-1	-3.58922648E-02	3.39267375E-02	-5.92213712E-03	5.56399165E-03	1.49063517E-03	-1.40038141E-03
7	-2	-4.33217879E-02	4.08912504E-02	-9.2000806E-03	8.60658871E-03	-6.97408521E-03	6.52337914E-03
7	-3	-5.41942395E-02	5.10205307E-02	-1.66531945E-02	1.54504087E-02	-1.69471710E-02	1.57190079E-02
7	-4	-9.31170389E-02	8.69860745E-02	-3.18547641E-02	3.11225707E-02	-3.78738988E-02	-3.73390401E-02
7	-5	-5.56430474E-02	5.54322101E-02	6.75641236E-02	-6.43952032E-02	-1.02001591E-01	9.75864803E-02
7	-6	0.	0.	0.	0.	0.	0.
7	-7	0.	0.	0.	0.	0.	0.
8	8	-6.15398979E-02	7.51511780E-02	5.40298073E-02	-6.23899846E-02	9.63065278E-02	-1.11734727E-01
8	6	-6.73784527E-02	7.80656965E-02	-3.25854394E-02	3.74499354E-02	-4.95508943E-02	5.69967196E-02
8	5	-4.01354093E-02	4.70355698E-02	-9.19908547E-03	1.06595010E-02	-1.16230536E-02	1.34631701E-02
8	4	-3.22017048E-02	3.78151778E-02	-5.39386189E-03	6.29013614E-03	-5.42663640E-03	6.32648970E-03
8	3	-2.81043413E-02	3.30316117E-02	-4.00938435E-03	4.68713806E-03	-3.02277022E-03	3.53340678E-03
8	2	-2.57852135E-02	3.03201599E-02	-3.36133177E-03	3.93376837E-03	-1.69069170E-03	1.97892725E-03
8	1	-2.45666888E-02	2.88944759E-02	-3.05520421E-03	3.57709645E-03	-7.69124580E-04	9.00832388E-04
8	0	-2.41845845E-02	2.84472780E-02	-2.9637479E-03	3.47043774E-03	0.	0.
8	-1	-2.45666888E-02	2.88944759E-02	-3.05520421E-03	3.57709645E-03	7.69124580E-04	-9.00832388E-04
8	-2	-2.57852135E-02	3.03201599E-02	-3.36133177E-03	3.93376837E-03	-1.69069170E-03	-1.97892725E-03
8	-3	-2.81043413E-02	3.30316117E-02	-4.00938435E-03	4.68713806E-03	-3.02277022E-03	-3.53340678E-03

8	-4	-3.22017040E-02	3.70151778E-02	-5.39386189E-03	6.29013614E-03	5.42663640E-03	-6.32648970E-03
8	-5	-4.01354093E-02	4.70355698E-02	-9.19908547E-03	1.06595010E-02	1.16230336E-02	-1.34631701E-02
8	-6	-6.73784527E-02	7.80656965E-02	-3.25854394E-02	3.74993354E-02	4.95508943E-02	-5.69967196E-02
8	-7	-6.15398979E-02	7.51517780E-02	5.40298073E-02	-6.23899846E-02	-9.63065278E-02	1.11734727E-01
8	-8	0.	0.	0.	0.	0.	0.
9	9	-1.91016241E-03	3.15651003E-03	1.15527028E-02	-1.89580107E-02	2.47730250E-02	-4.06551126E-02
9	8	-5.86648472E-02	8.86101458E-02	3.03279382E-02	-4.02846159E-02	6.10248666E-02	-8.14001675E-02
9	7	-4.70533389E-02	6.85290523E-02	-2.81140335E-02	3.91520514E-02	-5.09766547E-02	7.09297045E-02
9	6	-3.00275403E-02	4.41209153E-02	-5.61154846E-03	8.16167676E-03	-8.49377385E-03	1.23522917E-02
9	5	-2.40234205E-02	3.53650017E-02	-3.39587159E-03	4.96731463E-03	-4.27067600E-03	6.24658295E-03
9	4	-2.07188137E-02	3.05236820E-02	-2.52526002E-03	3.70316867E-03	-2.53739828E-03	3.72083317E-03
9	3	-1.87015916E-02	2.75632483E-02	-2.09005804E-03	3.06910987E-03	-1.57403477E-03	2.31130132E-03
9	2	-1.74665924E-02	2.57489776E-02	-1.85517066E-03	2.72626338E-03	-9.31089214E-04	1.36824996E-03
9	1	-1.67912920E-02	2.47563781E-02	-1.73608954E-03	2.55227390E-03	-4.35582028E-04	6.40348400E-04
9	0	-1.65760411E-02	2.44396386E-02	-1.69936650E-03	2.49868926E-03	0.	0.
9	-1	-1.67912920E-02	2.47563781E-02	-1.73608954E-03	2.55227390E-03	4.35582028E-04	-6.40348400E-04
9	-2	-1.74665924E-02	2.57489776E-02	-1.85517066E-03	2.72626338E-03	9.31089214E-04	-1.36824996E-03
9	-3	-1.87015916E-02	2.75632483E-02	-2.09005804E-03	3.06910987E-03	1.57403477E-03	-2.31130132E-03
9	-4	-2.07188137E-02	3.05236820E-02	-2.52526002E-03	3.70316867E-03	2.53739828E-03	-3.72083317E-03
9	-5	-2.40234205E-02	3.53650017E-02	-3.39587159E-03	4.96731463E-03	4.27067600E-03	-6.24658295E-03
9	-6	-3.00275403E-02	4.41209153E-02	-5.61154846E-03	8.16167676E-03	8.49377385E-03	-1.23522917E-02
9	-7	-4.70533389E-02	6.85290523E-02	-2.81140335E-02	3.91520514E-02	5.09766547E-02	-7.09297045E-02
9	-8	-5.86648472E-02	8.86101458E-02	3.03279382E-02	-4.02846159E-02	-6.10248666E-02	8.14001675E-02
9	-9	-1.91016241E-03	3.15651003E-03	1.15527028E-02	-1.89580107E-02	2.47730250E-02	-4.06551126E-02
10	10	-3.58627167E-03	7.62974542E-03	1.09058621E-02	-2.28652507E-02	2.62746278E-02	5.50987357E-02
10	9	-4.93988861E-02	9.47024542E-02	9.04988294E-03	-1.38232848E-02	2.17318192E-02	-3.36454123E-02
10	8	-3.33746722E-02	6.23050284E-02	-1.09548255E-02	1.99039991E-02	-2.23138428E-02	4.05252958E-02
10	7	-2.21584032E-02	4.16558268E-02	-3.55573550E-03	6.61879222E-03	-6.26813064E-03	1.16666287E-02
10	6	-1.76192213E-02	3.31835546E-02	-2.19659456E-03	4.11200149E-03	-3.31123875E-03	6.19830177E-03
10	5	-1.50088829E-02	2.82895513E-02	-1.63392090E-03	3.06670134E-03	-2.05037048E-03	3.84821408E-03
10	4	-1.33396297E-02	2.51540349E-02	-1.34059486E-03	2.51982266E-03	-1.34506727E-03	2.52817137E-03
10	3	-1.22400154E-02	2.30862162E-02	-1.17199983E-03	2.20484437E-03	-8.81650890E-04	1.65859036E-03
10	2	-1.15371879E-02	2.17636565E-02	-1.07358237E-03	2.02074685E-03	-5.39313642E-04	1.01322333E-03
10	1	-1.11435376E-02	2.10226062E-02	-1.02145635E-03	1.92317085E-03	-2.56065080E-04	4.82105711E-04
10	0	-1.10166619E-02	2.07835674E-02	-1.00506137E-03	1.89252180E-03	0.	0.
10	-1	-1.11435376E-02	2.10226062E-02	-1.02145635E-03	1.92317085E-03	2.56065080E-04	-4.82105711E-04
10	-2	-1.15371879E-02	2.17636565E-02	-1.07358237E-03	2.02074685E-03	5.39313642E-04	-1.01322333E-03
10	-3	-1.22400154E-02	2.30862162E-02	-1.17199983E-03	2.20484437E-03	8.81650890E-04	-1.65859036E-03
10	-4	-1.33396297E-02	2.51540349E-02	-1.34059486E-03	2.51982266E-03	1.34506727E-03	-2.52817137E-03
10	-5	-1.50088829E-02	2.82895513E-02	-1.63392090E-03	3.06670134E-03	-2.05037048E-03	3.84821408E-03
10	-6	-1.76192213E-02	3.31835546E-02	-2.19659456E-03	4.11200149E-03	3.31123875E-03	-6.19830177E-03
10	-7	-2.21584032E-02	4.16558268E-02	-3.55573550E-03	6.61879222E-03	6.26813064E-03	-1.16666287E-02
10	-8	-3.33746722E-02	6.23050284E-02	-1.09548255E-02	1.99039991E-02	2.23138428E-02	-4.05252958E-02
10	-9	-4.93988861E-02	9.47024542E-02	9.04988294E-03	-1.38232848E-02	-2.17318192E-02	3.36454123E-02

10	-10	-3.58627167E-03	7.62974673E-03	1.09058621E-02	-2.28632507E-02	-2.62746278E-02	5.50987357E-02
11	11	-4.18639975E-03	1.20663134E-02	8.56562525E-03	-2.40799525E-02	2.28776961E-02	-6.43417919E-02
11	10	-3.78975536E-02	9.62892728E-02	3.27966726E-03	-4.93045524E-03	9.24313571E-03	-1.47902829E-02
11	9	-2.33397754E-02	5.79961587E-02	-6.08233328E-03	1.47178448E-02	-1.38752164E-02	3.35630000E-02
11	8	-1.57211284E-02	3.93316120E-02	-2.24058933E-03	5.54728295E-03	-4.50866281E-03	1.11615796E-02
11	7	-1.23591352E-02	3.10821435E-02	-1.40397486E-03	3.49756058E-03	-2.46718435E-03	6.14591930E-03
11	6	-1.04283274E-02	2.61643294E-02	-1.04472019E-03	2.61029182E-03	-1.57217571E-03	3.92804894E-03
11	5	-9.12851150E-03	2.29137463E-02	-8.52110504E-04	2.13263015E-03	-1.06807889E-03	2.67308894E-03
11	4	-8.23503168E-03	2.06763761E-02	-7.37344164E-04	1.84731796E-03	-7.39176940E-04	1.85187834E-03
11	3	-7.62074435E-03	1.91368493E-02	-6.65957676E-04	1.66957404E-03	-5.00633487E-04	1.25508447E-03
11	2	-7.21772822E-03	1.81262757E-02	-6.22218396E-04	1.56056085E-03	-3.11809922E-04	7.82028633E-04
11	1	-6.98853946E-03	1.75513977E-02	-5.98386689E-04	1.50112902E-03	-1.49927951E-04	3.76109275E-04
11	0	-6.91414032E-03	1.73646556E-02	-5.90791626E-04	1.48221511E-03	0.	0.
11	-1	-6.98853946E-03	1.75513977E-02	-5.98386689E-04	1.50112902E-03	1.49927951E-04	-3.76109275E-04
11	-2	-7.21772822E-03	1.81262757E-02	-6.22218396E-04	1.56056085E-03	3.11809922E-04	-7.82028633E-04
11	-3	-7.62074435E-03	1.91368493E-02	-6.65957676E-04	1.66957404E-03	5.00633487E-04	-1.25508447E-03
11	-4	-8.23503168E-03	2.06763761E-02	-7.37344164E-04	1.84731796E-03	7.39176940E-04	-1.85187834E-03
11	-5	-9.12851150E-03	2.29137463E-02	-8.52110504E-04	2.13263015E-03	1.06807889E-03	-2.67308894E-03
11	-6	-1.04283274E-02	2.61643294E-02	-1.04472019E-03	2.61029182E-03	1.57217571E-03	-3.92804894E-03
11	-7	-1.23591352E-02	3.10821435E-02	-1.40397486E-03	3.49756058E-03	2.46718435E-03	-6.14591930E-03
11	-8	-1.57211284E-02	3.93316120E-02	-2.24058933E-03	5.54728295E-03	4.50866281E-03	-1.11615796E-02
11	-9	-2.33397754E-02	5.79961587E-02	-6.08233328E-03	1.47178448E-02	1.38752164E-02	-3.35630000E-02
11	-10	-3.78975536E-02	9.62892728E-02	-6.27966726E-03	-4.93045524E-03	-9.24313571E-03	1.47902829E-02
11	-11	-4.18639975E-03	1.20663134E-02	8.56562525E-03	-2.40799525E-02	-2.28776961E-02	6.43417919E-02
12	12	-3.72776720E-03	1.20663134E-02	8.56562525E-03	-2.40799525E-02	1.70333830E-02	-7.04581208E-02
12	11	-2.62917539E-02	9.55891131E-02	1.02580352E-03	4.09961357E-05	3.50152290E-03	-2.07692808E-02
12	10	-1.52506711E-02	5.43187266E-02	-3.38304633E-03	1.16880687E-02	-8.55318683E-03	2.95400203E-02
12	9	-1.03153072E-02	3.70290494E-02	-1.33820525E-03	4.74458707E-03	-3.02684968E-03	1.07305559E-02
12	8	-6.05661541E-03	2.89904861E-02	-8.46758430E-04	3.02522663E-03	-1.69955143E-03	6.07168542E-03
12	7	-5.76180714E-03	2.07657237E-02	-5.11573400E-04	2.25950416E-03	-1.10566569E-03	3.96486125E-03
12	6	-5.10897888E-03	1.84179437E-02	-4.39299702E-04	1.83850727E-03	-7.69132814E-04	2.76406283E-03
12	5	-4.64105922E-03	1.67332808E-02	-3.92716140E-04	1.58092128E-03	-5.50263143E-04	1.98021174E-03
12	4	-4.31064695E-03	1.55428127E-02	-3.62239121E-04	1.41454784E-03	-3.93483559E-04	1.41729147E-03
12	3	-4.05014224E-03	1.47479566E-02	-3.42948235E-04	1.30554646E-03	-2.72196355E-04	9.81011826E-04
12	2	-3.96345727E-03	1.42911603E-02	-3.32229523E-04	1.23648864E-03	-1.71797790E-04	6.19403321E-04
12	1	-3.92213375E-03	1.41420631E-02	-3.28781103E-04	1.19809594E-03	-8.32140693E-05	3.00086475E-04
12	0	-3.96345727E-03	1.42911603E-02	-3.32229523E-04	1.18576433E-03	0.	0.
12	-1	-4.09014224E-03	1.47479566E-02	-3.42948235E-04	1.19809594E-03	8.32140693E-05	-3.00086475E-04
12	-2	-4.31064695E-03	1.55428127E-02	-3.62239121E-04	1.23648864E-03	1.71797790E-04	-6.19403321E-04
12	-3	-4.64105922E-03	1.67332808E-02	-3.92716140E-04	1.30554646E-03	2.72196355E-04	-9.81011826E-04
12	-4	-5.10897888E-03	1.84179437E-02	-4.39299702E-04	1.41454784E-03	3.93483559E-04	-1.41729147E-03
12	-5	-5.76180714E-03	2.07657237E-02	-5.11573400E-04	1.58092128E-03	5.50263143E-04	-1.98021174E-03
12	-6	-6.05661541E-03	2.89904861E-02	-8.46758430E-04	1.83850727E-03	7.69132814E-04	-2.76406283E-03
12	-7	-6.98853946E-03	3.10821435E-02	-1.04472019E-03	2.25950416E-03	1.10566569E-03	-3.96486125E-03

12	-8	-8.05661541E-03	2.89954861E-02	-8.46758430E-04	3.02522663E-03	1.69955143E-03	-6.07168542E-03
12	-9	-1.03153072E-02	3.70290494E-02	-1.33820525E-03	4.74458787E-03	3.02684968E-03	-1.07305559E-02
12	-10	-1.52506711E-02	5.43187266E-02	-3.38304633E-03	1.16880687E-02	8.53318683E-03	-2.95400203E-02
12	-11	-2.62917539E-02	9.5091131E-02	1.02580352E-03	4.09961357E-05	-3.50152909E-03	2.07692808E-03
12	-12	-3.72776720E-03	1.60534916E-02	5.82061141E-03	-2.40293415E-02	-1.70533830E-02	7.04581208E-02
13	13	-2.37982443E-03	1.94553031E-02	3.11216511E-03	-2.32790623E-02	9.91317750E-03	-7.42776469E-02
13	12	-1.52050771E-02	9.33755139E-02	3.4077951E-04	3.01418639E-03	1.40306647E-03	7.16239221E-03
13	11	-8.47075744E-03	5.08228967E-02	-1.67259381E-03	9.62191041E-03	-4.64421110E-03	2.67048472E-02
13	10	-5.71152866E-03	3.46955701E-02	-6.88665739E-04	4.10942874E-03	-1.72968514E-03	1.03199772E-02
13	9	-4.40961260E-03	2.68824055E-02	-4.37916073E-04	2.64354961E-03	-9.88367581E-04	5.96597701E-03
13	8	-3.60689733E-03	2.20219343E-02	-3.25424985E-04	1.97579521E-03	-6.52372674E-04	3.96063362E-03
13	7	-3.05762254E-03	1.86812736E-02	-2.63066930E-04	1.60257641E-03	-4.61255474E-04	2.80981791E-03
13	6	-2.66227705E-03	1.62702707E-02	-2.24440739E-04	1.37021537E-03	-3.37239926E-04	2.05880126E-03
13	5	-2.37139376E-03	1.44931720E-02	-1.98987432E-04	1.21657840E-03	-2.49140101E-04	1.52317358E-03
13	4	-2.15730557E-03	1.31836639E-02	-1.81720132E-04	1.11210912E-03	-1.82014008E-04	1.11389114E-03
13	3	-2.00343382E-03	1.22417036E-02	-1.70025667E-04	1.04124164E-03	-1.27728500E-04	7.82204177E-04
13	2	-1.89954413E-03	1.16053736E-02	-1.5151508E-04	9.95292673E-04	-8.13619251E-05	4.98476564E-04
13	1	-1.83943247E-03	1.12370608E-02	-1.58183254E-04	9.69381543E-04	-3.96132471E-05	2.42757056E-04
13	0	-1.81574877E-03	1.1164365E-02	-1.56803425E-04	9.61002342E-04	0.	0.
13	-1	-1.83943247E-03	1.12370608E-02	-1.58183254E-04	9.69381543E-04	3.96132471E-05	-2.42757056E-04
13	-2	-1.89954413E-03	1.16053736E-02	-1.5151508E-04	9.95292673E-04	8.13619251E-05	-4.98476564E-04
13	-3	-2.00343382E-03	1.22417036E-02	-1.70025667E-04	1.04124164E-03	1.27728500E-04	-7.82204177E-04
13	-4	-2.15730557E-03	1.31836639E-02	-1.81720132E-04	1.11210912E-03	1.82014008E-04	-1.11389114E-03
13	-5	-2.37139376E-03	1.44931720E-02	-1.98987432E-04	1.21657840E-03	2.49140101E-04	-1.52317358E-03
13	-6	-2.66227705E-03	1.62702707E-02	-2.24440739E-04	1.37021537E-03	3.37239926E-04	-2.05880126E-03
13	-7	-3.05762254E-03	1.86812736E-02	-2.63066930E-04	1.60257641E-03	4.61255474E-04	-2.80981791E-03
13	-8	-3.60689733E-03	2.20219343E-02	-3.25424985E-04	1.97579521E-03	6.52372674E-04	-3.96063362E-03
13	-9	-4.40961260E-03	2.68824055E-02	-4.37916073E-04	2.64354961E-03	9.88367581E-04	-5.96597701E-03
13	-10	-5.71152866E-03	3.46955701E-02	-6.88665739E-04	4.10942874E-03	1.72968514E-03	-1.03199772E-02
13	-11	-8.47075744E-03	5.08228967E-02	-1.67259381E-03	9.62191041E-03	4.64421110E-03	-2.67048472E-02
13	-12	-1.52050771E-02	9.33755139E-02	3.4077951E-04	3.01418639E-03	-1.40306647E-03	-7.16239221E-03
13	-13	-2.37982443E-03	1.94553031E-02	3.11216511E-03	-2.32790623E-02	9.91317750E-03	7.42776469E-02
14	14	-3.7797820E-04	2.21734002E-02	6.13399072E-04	-2.21130237E-02	2.09519971E-03	-7.62485125E-02
14	13	-4.88029980E-03	9.00444575E-02	4.24329426E-04	4.79966496E-03	1.53358715E-03	1.40154796E-02
14	12	-2.67813939E-03	4.75259637E-02	-5.08248412E-04	8.08657408E-03	-1.53853599E-03	2.44548289E-02
14	11	-1.76752479E-03	3.23150690E-02	-2.05463417E-04	3.58583402E-03	-5.67491190E-04	9.90055402E-03
14	10	-1.34114504E-03	2.47538405E-02	-1.28969852E-04	2.32330469E-03	-3.23336994E-04	5.82348819E-03
14	9	-1.07828293E-03	1.99791133E-02	-9.49001168E-05	1.73739101E-03	-2.13963810E-04	3.91659915E-03
14	8	-6.97576116E-04	1.66547183E-02	-7.60369332E-05	1.40549463E-03	-1.52321872E-04	2.81527909E-03
14	7	-7.66319627E-04	1.42210380E-02	-6.43051829E-05	1.19606537E-03	-1.12693768E-04	2.09592267E-03
14	6	-6.68289510E-04	1.23939742E-02	-5.64950716E-05	1.05265558E-03	-8.48566207E-05	1.58493940E-03
14	5	-5.94379382E-04	1.10115304E-02	-5.10936127E-05	9.57210891E-04	-6.39538869E-05	1.19809510E-03
14	4	-5.39068248E-04	9.97437849E-03	-4.73045206E-05	8.8803239E-04	-4.73719298E-05	8.89327316E-04
14	3	-4.98856938E-04	9.21954634E-03	-4.46777593E-05	8.39990809E-04	-3.35568745E-05	6.30935449E-04

SAMPLE CASE --- TWO AR=2 SURF
ORIENTAL AND VERTICAL SEPARATIONS

AIC CALCULATIONS

MACH = 1.20000 K1 = .0331662 ERR = 1.00000E-02 EL = 0.00 YEAR = 0.00

NU	MU	C			W			V		
		VELOCITY	POTENTIAL	COEFFICIENT	UPWASH	COEFFICIENT	SIDEWASH	COEFFICIENT	COEFFICIENT	COEFFICIENT
0	0	-4.99669441E-01	1.35612131E-02	0.	0.	0.	0.	0.	0.	0.
1	0	-3.83780888E-01	3.72653793E-02	0.	0.	0.	0.	0.	0.	0.
1	-1	-3.02537716E-01	3.51774045E-02	0.	0.	0.	0.	0.	0.	0.
2	0	-1.58274582E-01	3.41050490E-02	0.	0.	0.	0.	0.	0.	0.
2	-1	-1.98775746E-01	4.21384433E-02	0.	0.	0.	0.	0.	0.	0.
2	-2	-2.06053452E-01	4.75173836E-02	0.	0.	0.	0.	0.	0.	0.
3	0	-9.83703960E-02	3.28543730E-02	0.	0.	0.	0.	0.	0.	0.
3	-1	-1.05489007E-01	3.51833148E-02	0.	0.	0.	0.	0.	0.	0.
3	-2	-1.47785247E-01	4.86337200E-02	0.	0.	0.	0.	0.	0.	0.
3	-3	-1.62295376E-01	5.66261105E-02	0.	0.	0.	0.	0.	0.	0.
4	0	-6.81525261E-02	3.13856404E-02	0.	0.	0.	0.	0.	0.	0.
4	-1	-7.08290748E-02	3.26038898E-02	0.	0.	0.	0.	0.	0.	0.
4	-2	-8.09774972E-02	3.72153085E-02	0.	0.	0.	0.	0.	0.	0.
4	-3	-1.18703126E-01	5.39547333E-02	0.	0.	0.	0.	0.	0.	0.
4	-4	-1.34046042E-01	6.38424648E-02	0.	0.	0.	0.	0.	0.	0.
5	0	-4.93550501E-02	2.96149940E-02	0.	0.	0.	0.	0.	0.	0.
5	-1	-5.06323716E-02	3.03790933E-02	0.	0.	0.	0.	0.	0.	0.
5	-2	-5.50235203E-02	3.29920612E-02	0.	0.	0.	0.	0.	0.	0.
5	-3	-6.50871190E-02	3.89671498E-02	0.	0.	0.	0.	0.	0.	0.
5	-4	-9.81522509E-02	5.81981438E-02	0.	0.	0.	0.	0.	0.	0.
5	-5	-1.12751838E-01	6.96207600E-02	0.	0.	0.	0.	0.	0.	0.
6	0	-3.63050688E-02	2.75563287E-02	0.	0.	0.	0.	0.	0.	0.
6	-1	-3.70065878E-02	2.80858986E-02	0.	0.	0.	0.	0.	0.	0.
6	-2	-3.92959019E-02	2.98145968E-02	0.	0.	0.	0.	0.	0.	0.
6	-3	-4.35915269E-02	3.32964611E-02	0.	0.	0.	0.	0.	0.	0.
6	-4	-5.31898899E-02	4.02714174E-02	0.	0.	0.	0.	0.	0.	0.
6	-5	-8.19406102E-02	6.14878343E-02	0.	0.	0.	0.	0.	0.	0.
6	-6	-9.52385400E-02	7.41890253E-02	0.	0.	0.	0.	0.	0.	0.
7	0	-2.66803249E-02	2.52527061E-02	0.	0.	0.	0.	0.	0.	0.
7	-1	-2.75944500E-02	2.56432429E-02	0.	0.	0.	0.	0.	0.	0.
7	-2	-2.84124674E-02	2.68859576E-02	0.	0.	0.	0.	0.	0.	0.

7	-3	-3.09101106E-02	2.92368664E-02	0.	0.	0.	0.
7	-4	-3.52622140E-02	3.33509956E-02	0.	0.	0.	0.
7	-5	-4.35481149E-02	4.11045928E-02	0.	0.	0.	0.
7	-6	-6.82681173E-02	6.39065410E-02	0.	0.	0.	0.
7	-7	-8.00574122E-02	7.76904920E-02	0.	0.	0.	0.
8	0	-1.93534078E-02	2.27776232E-02	0.	0.	0.	0.
8	-1	-1.96065790E-02	2.30748510E-02	0.	0.	0.	0.
8	-2	-2.04006206E-02	2.40068433E-02	0.	0.	0.	0.
8	-3	-2.18546126E-02	2.57125717E-02	0.	0.	0.	0.
8	-4	-2.42349520E-02	2.85024106E-02	0.	0.	0.	0.
8	-5	-2.81504160E-02	3.30834487E-02	0.	0.	0.	0.
8	-6	-3.53149521E-02	4.14460143E-02	0.	0.	0.	0.
8	-7	-5.63226441E-02	6.55224972E-02	0.	0.	0.	0.
8	-8	-6.64812250E-02	8.02511141E-02	0.	0.	0.	0.
9	0	-1.36536158E-02	2.01395719E-02	0.	0.	0.	0.
9	-1	-1.38130648E-02	2.03743775E-02	0.	0.	0.	0.
9	-2	-1.43081589E-02	2.11032178E-02	0.	0.	0.	0.
9	-3	-1.51944494E-02	2.24075396E-02	0.	0.	0.	0.
9	-4	-1.65876175E-02	2.44565463E-02	0.	0.	0.	0.
9	-5	-1.87192445E-02	2.75883532E-02	0.	0.	0.	0.
9	-6	-2.20996834E-02	3.25455499E-02	0.	0.	0.	0.
9	-7	-2.81468341E-02	4.13902112E-02	0.	0.	0.	0.
9	-8	-4.55800825E-02	6.64147609E-02	0.	0.	0.	0.
9	-9	-5.40196170E-02	8.18676526E-02	0.	0.	0.	0.
10	0	-9.22935494E-03	1.74162743E-02	0.	0.	0.	0.
10	-1	-9.32978337E-03	1.76055805E-02	0.	0.	0.	0.
10	-2	-9.63939834E-03	1.81891284E-02	0.	0.	0.	0.
10	-3	-1.01853736E-02	1.92178729E-02	0.	0.	0.	0.
10	-4	-1.10215237E-02	2.07926504E-02	0.	0.	0.	0.
10	-5	-1.22469657E-02	2.30989010E-02	0.	0.	0.	0.
10	-6	-1.40517571E-02	2.64913718E-02	0.	0.	0.	0.
10	-7	-1.68462984E-02	3.17353784E-02	0.	0.	0.	0.
10	-8	-2.17694763E-02	4.09392936E-02	0.	0.	0.	0.
10	-9	-3.37973860E-02	6.66196905E-02	0.	0.	0.	0.
10	-10	-4.24590539E-02	8.26216243E-02	0.	0.	0.	0.
11	0	-5.83916779E-03	1.46664657E-02	0.	0.	0.	0.
11	-1	-5.90078868E-03	1.48212020E-02	0.	0.	0.	0.
11	-2	-6.08980302E-03	1.52928524E-02	0.	0.	0.	0.
11	-3	-6.41970067E-03	1.61239579E-02	0.	0.	0.	0.
11	-4	-6.91605609E-03	1.73694020E-02	0.	0.	0.	0.
11	-5	-7.62351803E-03	1.91435636E-02	0.	0.	0.	0.
11	-6	-8.82033143E-03	2.16410619E-02	0.	0.	0.	0.
11	-7	-1.00333793E-02	2.52262670E-02	0.	0.	0.	0.
11	-8	-1.22367151E-02	3.06771025E-02	0.	0.	0.	0.

11	-9	-1.60386098E-02	4.01251198E-02	0.	0.	0.	0.
11	-10	-2.67874470E-02	6.61820553E-02	0.	0.	0.	0.
11	-11	-3.16568298E-02	8.25593435E-02	0.	0.	0.	0.
12	0	-3.31410387E-03	1.19482052E-02	0.	0.	0.	0.
12	-1	-3.34944725E-03	1.20757813E-02	0.	0.	0.	0.
12	-2	-3.45748920E-03	1.24657298E-02	0.	0.	0.	0.
12	-3	-3.64460663E-03	1.31409284E-02	0.	0.	0.	0.
12	-4	-3.92269497E-03	1.41440199E-02	0.	0.	0.	0.
12	-5	-4.31160686E-03	1.55460838E-02	0.	0.	0.	0.
12	-6	-4.84400056E-03	1.74640442E-02	0.	0.	0.	0.
12	-7	-5.57591213E-03	2.00974176E-02	0.	0.	0.	0.
12	-8	-6.61109057E-03	2.38148893E-02	0.	0.	0.	0.
12	-9	-8.17079482E-03	2.93981421E-02	0.	0.	0.	0.
12	-10	-1.08640675E-02	3.89711497E-02	0.	0.	0.	0.
12	-11	-1.84600241E-02	6.51450574E-02	0.	0.	0.	0.
12	-12	-2.15311213E-02	8.17247372E-02	0.	0.	0.	0.
13	0	-1.52635071E-03	9.31724946E-03	0.	0.	0.	0.
13	-1	-1.54357226E-03	9.42290476E-03	0.	0.	0.	0.
13	-2	-1.59608036E-03	9.74500408E-03	0.	0.	0.	0.
13	-3	-1.68652388E-03	1.02996621E-02	0.	0.	0.	0.
13	-4	-1.81975013E-03	1.11163399E-02	0.	0.	0.	0.
13	-5	-2.00360435E-03	1.22426482E-02	0.	0.	0.	0.
13	-6	-2.25044534E-03	1.37534277E-02	0.	0.	0.	0.
13	-7	-2.58009258E-03	1.57683366E-02	0.	0.	0.	0.
13	-8	-3.02600089E-03	1.84885050E-02	0.	0.	0.	0.
13	-9	-3.64383931E-03	2.22824510E-02	0.	0.	0.	0.
13	-10	-4.58288633E-03	2.79275002E-02	0.	0.	0.	0.
13	-11	-6.18762208E-03	3.75285221E-02	0.	0.	0.	0.
13	-12	-1.07562881E-02	6.35516015E-02	0.	0.	0.	0.
13	-13	-1.20383237E-02	8.01615653E-02	0.	0.	0.	0.
14	0	-3.71815945E-04	6.82555445E-03	0.	0.	0.	0.
14	-1	-3.76456183E-04	6.91313772E-03	0.	0.	0.	0.
14	-2	-3.90578748E-04	7.17962086E-03	0.	0.	0.	0.
14	-3	-4.14813348E-04	7.63663801E-03	0.	0.	0.	0.
14	-4	-4.50296488E-04	8.30514014E-03	0.	0.	0.	0.
14	-5	-4.98827722E-04	9.21818616E-03	0.	0.	0.	0.
14	-6	-5.63153510E-04	1.04259810E-02	0.	0.	0.	0.
14	-7	-6.47487118E-04	1.20250222E-02	0.	0.	0.	0.
14	-8	-7.58511667E-04	1.40755424E-02	0.	0.	0.	0.
14	-9	-9.07489381E-04	1.68376250E-02	0.	0.	0.	0.
14	-10	-1.11529580E-03	2.06556064E-02	0.	0.	0.	0.
14	-11	-1.42697726E-03	2.62948520E-02	0.	0.	0.	0.
14	-12	-1.97041278E-03	3.58216463E-02	0.	0.	0.	0.
14	-13	-3.61572184E-03	6.14451136E-02	0.	0.	0.	0.

14	-14	-3.16510175E-03	7.79147557E-02	0.	0.	0.	0.
15	0	2.39657385E-04	4.51992878E-03	0.	0.	0.	0.
15	-1	2.43895493E-04	4.59237037E-03	0.	0.	0.	0.
15	-2	2.56161447E-04	4.81246030E-03	0.	0.	0.	0.
15	-3	2.77124615E-04	5.18876549E-03	0.	0.	0.	0.
15	-4	3.07618702E-04	5.73652946E-03	0.	0.	0.	0.
15	-5	3.48931101E-04	6.47935636E-03	0.	0.	0.	0.
15	-6	4.02957604E-04	7.44521421E-03	0.	0.	0.	0.
15	-7	4.72467458E-04	8.70613507E-03	0.	0.	0.	0.
15	-8	5.61571159E-04	1.03179566E-02	0.	0.	0.	0.
15	-9	6.76593686E-04	1.24066678E-02	0.	0.	0.	0.
15	-10	8.27843443E-04	1.51689736E-02	0.	0.	0.	0.
15	-11	1.03363264E-03	1.89613281E-02	0.	0.	0.	0.
15	-12	1.33109416E-03	2.45299598E-02	0.	0.	0.	0.
15	-13	1.81457923E-03	3.38840698E-02	0.	0.	0.	0.
15	-14	2.90106485E-03	5.88699629E-02	0.	0.	0.	0.
15	-15	5.11245065E-03	7.50311840E-02	0.	0.	0.	0.
16	0	3.90947265E-04	2.44086871E-03	0.	0.	0.	0.
16	-1	4.00828484E-04	2.50047643E-03	0.	0.	0.	0.
16	-2	4.30815044E-04	2.68137753E-03	0.	0.	0.	0.
16	-3	4.81964054E-04	2.98997947E-03	0.	0.	0.	0.
16	-4	5.56137462E-04	3.43757361E-03	0.	0.	0.	0.
16	-5	6.56173024E-04	4.04138198E-03	0.	0.	0.	0.
16	-6	7.86172918E-04	4.82632687E-03	0.	0.	0.	0.
16	-7	9.51981429E-04	5.82796599E-03	0.	0.	0.	0.
16	-8	1.16105560E-03	7.09745539E-03	0.	0.	0.	0.
16	-9	1.44855805E-03	8.71031390E-03	0.	0.	0.	0.
16	-10	1.770756494E-03	1.07829270E-02	0.	0.	0.	0.
16	-11	2.21935250E-03	1.35065064E-02	0.	0.	0.	0.
16	-12	2.83027494E-03	1.72261705E-02	0.	0.	0.	0.
16	-13	3.71836382E-03	2.26622385E-02	0.	0.	0.	0.
16	-14	5.18722403E-03	3.17493620E-02	0.	0.	0.	0.
16	-15	8.88566854E-03	5.58716323E-02	0.	0.	0.	0.
16	-16	1.26955900E-02	7.15605086E-02	0.	0.	0.	0.
17	0	1.58787530E-04	6.21604294E-04	0.	0.	0.	0.
17	-1	1.72400938E-04	6.70254190E-04	0.	0.	0.	0.
17	-2	2.13682004E-04	8.17782706E-04	0.	0.	0.	0.
17	-3	2.83984664E-04	1.06903946E-03	0.	0.	0.	0.
17	-4	3.85675563E-04	1.43250287E-03	0.	0.	0.	0.
17	-5	5.22318763E-04	1.92094629E-03	0.	0.	0.	0.
17	-6	6.98980929E-04	2.55254234E-03	0.	0.	0.	0.
17	-7	9.2270363E-04	3.35261876E-03	0.	0.	0.	0.
17	-8	1.20337875E-03	4.35652728E-03	0.	0.	0.	0.
17	-9	1.55490318E-03	5.61442762E-03	0.	0.	0.	0.

SAMPLE CASE --- TWO AR-2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

WING UPPER SURFACE NORMAL WASH
(MACH 1.200 RED. FREQ. = .50000)

MODE SHAPE 2

ROW	CHORD 1		CHORD 2		CHORD 3		CHORD 4	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
1	6.63324958E-02	1.10000000E-03	6.63324958E-02	1.10000000E-03	6.63324958E-02	1.10000000E-03	6.63324958E-02	1.10000000E-03
2	6.63324958E-02	3.30000000E-03	6.63324958E-02	3.30000000E-03	6.63324958E-02	3.30000000E-03	6.63324958E-02	3.30000000E-03
3	6.63324958E-02	5.50000000E-03	6.63324958E-02	5.50000000E-03	6.63324958E-02	5.50000000E-03	6.63324958E-02	5.50000000E-03
4	6.63324958E-02	7.70000000E-03	6.63324958E-02	7.70000000E-03	6.63324958E-02	7.70000000E-03	6.63324958E-02	7.70000000E-03
5	6.63324958E-02	9.90000000E-03	6.63324958E-02	9.90000000E-03	6.63324958E-02	9.90000000E-03	6.63324958E-02	9.90000000E-03
6	6.63324958E-02	1.21000000E-02	6.63324958E-02	1.21000000E-02	6.63324958E-02	1.21000000E-02	6.63324958E-02	1.21000000E-02
7	6.63324958E-02	1.43000000E-02	6.63324958E-02	1.43000000E-02	6.63324958E-02	1.43000000E-02	6.63324958E-02	1.43000000E-02
8	6.63324958E-02	1.65000000E-02	6.63324958E-02	1.65000000E-02	6.63324958E-02	1.65000000E-02	6.63324958E-02	1.65000000E-02
9	6.63324958E-02	1.87000000E-02	6.63324958E-02	1.87000000E-02	6.63324958E-02	1.87000000E-02	6.63324958E-02	1.87000000E-02
10	6.63324958E-02	2.09000000E-02	6.63324958E-02	2.09000000E-02	6.63324958E-02	2.09000000E-02	6.63324958E-02	2.09000000E-02
11	6.63324958E-02	2.31000000E-02	6.63324958E-02	2.31000000E-02	6.63324958E-02	2.31000000E-02	6.63324958E-02	2.31000000E-02
12	6.63324958E-02	2.53000000E-02	6.63324958E-02	2.53000000E-02	6.63324958E-02	2.53000000E-02	6.63324958E-02	2.53000000E-02
13	6.63324958E-02	2.75000000E-02	6.63324958E-02	2.75000000E-02	6.63324958E-02	2.75000000E-02	6.63324958E-02	2.75000000E-02
14	6.63324958E-02	2.97000000E-02	6.63324958E-02	2.97000000E-02	6.63324958E-02	2.97000000E-02	6.63324958E-02	2.97000000E-02
15	6.63324958E-02	3.19000000E-02	6.63324958E-02	3.19000000E-02	6.63324958E-02	3.19000000E-02	6.63324958E-02	3.19000000E-02
16	4.69641560E-02	1.30749482E-02	4.93038417E-02	1.22854610E-02	5.00166418E-02	1.10011154E-02	5.13430376E-02	1.08606884E-02
17	5.43053085E-02	1.25650430E-02	5.35634354E-02	1.14413489E-02	5.36681729E-02	-8.06011302E-03	5.40440732E-02	-9.10498700E-03
18	5.43053085E-02	2.02172665E-02	5.43168889E-02	1.95243635E-02	5.44424617E-02	-1.89244049E-02	5.46292480E-02	-2.12867506E-02
19	5.79342031E-02	1.96628840E-02	5.78454023E-02	1.88978027E-02	5.76530897E-02	-1.65356306E-02	5.69733825E-02	-1.69204435E-02
20	5.79342031E-02	2.71512149E-02	5.71352697E-02	2.71388378E-02	5.68186521E-02	-2.65568648E-02	5.62169272E-02	-2.88472801E-02
21	6.04595933E-02	2.70321600E-02	6.02605942E-02	2.62106415E-02	5.97877831E-02	-2.45199788E-02	5.84133933E-02	-2.65929915E-02
22	5.83434725E-02	3.42381761E-02	5.82811655E-02	3.39618414E-02	5.77905313E-02	-3.30531375E-02	5.65157681E-02	-3.52168190E-02
23	6.11982717E-02	3.39216002E-02	6.09598075E-02	3.30474671E-02	6.03843800E-02	-3.12862399E-02	5.84167806E-02	-3.33037775E-02
24	5.82471648E-02	4.04904736E-02	5.78703147E-02	4.01406143E-02	5.71449832E-02	-3.90800117E-02	5.53246249E-02	-4.10897404E-02
25	6.03806479E-02	4.00119142E-02	6.01789947E-02	3.92066531E-02	5.93362001E-02	-3.75892667E-02	5.89321679E-02	-3.95670355E-02
26	5.62229702E-02	4.59595304E-02	5.60746605E-02	4.56029135E-02	5.52331355E-02	-4.44969255E-02	5.30926973E-02	-4.64212555E-02
27	5.83342587E-02	4.54480721E-02	5.81445607E-02	4.47221836E-02	5.73011295E-02	-4.31999606E-02	5.48365432E-02	-4.51101685E-02
28	5.37129437E-02	5.07685932E-02	5.32926932E-02	5.03777030E-02	5.25698835E-02	-4.92341988E-02	5.02389327E-02	-5.10742794E-02
29	5.55032801E-02	5.02873669E-02	5.51823218E-02	4.96013106E-02	5.44977244E-02	-4.81416657E-02	5.17346319E-02	-4.99633399E-02

ROW	CHORD 5		CHORD 6		CHORD 7		CHORD 8	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
1	6.63324958E-02	1.10000000E-03	6.63324958E-02	1.10000000E-03	6.63324958E-02	1.10000000E-03	6.63324958E-02	1.10000000E-03
2	6.63324958E-02	3.30000000E-03	6.63324958E-02	3.30000000E-03	6.63324958E-02	3.30000000E-03	6.63324958E-02	3.30000000E-03
3	6.63324958E-02	5.50000000E-03	6.63324958E-02	5.50000000E-03	6.63324958E-02	5.50000000E-03	6.63324958E-02	5.50000000E-03
4	6.63324958E-02	7.70000000E-03	6.63324958E-02	7.70000000E-03	6.63324958E-02	7.70000000E-03	6.63324958E-02	7.70000000E-03

A 21

SAMPLE CASE --- TWO AR-2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

WING UPPER SURFACE NORMAL WASH

(MACH 1.200 RED. FREQ. 2 .50000)

MODE SHAPE 2

PAGE CONTINUED

ROW	CHORD 5		CHORD 6		CHORD 7		CHORD 8	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
16	5.30174346E-02	-9.91932790E-04	5.50613145E-02	1.04153983E-02	5.69535875E-02	1.23276743E-02	5.68563377E-02	1.49503673E-02
17	5.43792077E-02	-5.14658627E-03	5.64535688E-02	6.87301296E-03	5.88168279E-02	1.24604221E-02	6.05004025E-02	1.52164635E-02
18	5.59270913E-02	-1.44806783E-02	5.75290193E-02	-1.04166120E-03	5.99711433E-02	7.29512421E-03	6.20593209E-02	1.32708146E-02
19	5.67210304E-02	-1.47246991E-02	5.82865656E-02	-3.40350016E-03	6.00786565E-02	3.32387786E-03	6.20850019E-02	1.00460534E-02
20	5.66753455E-02	-2.26377513E-02	5.86437178E-02	-9.53426289E-03	6.01737174E-02	-1.99920037E-03	6.14715913E-02	4.29761481E-03
21	5.77773579E-02	-2.21106111E-02	5.90086164E-02	-1.10086512E-02	5.99448603E-02	-4.79230398E-03	6.10546534E-02	1.06655474E-03
22	5.65311153E-02	-2.93613307E-02	5.63963492E-02	-1.68161852E-02	5.93276447E-02	-9.65812058E-03	6.05020749E-02	-3.49853931E-03
23	5.73780943E-02	-2.67875354E-02	5.86483818E-02	-1.78121969E-02	5.93340972E-02	-1.17494530E-02	6.01318867E-02	-6.13287008E-03
24	5.52344872E-02	-3.54524593E-02	5.74486641E-02	-2.32323246E-02	5.85094276E-02	-1.62468699E-02	5.94703856E-02	-1.01681791E-02
25	5.60419323E-02	-3.48533143E-02	5.77589677E-02	-2.38314847E-02	5.84255983E-02	-1.79039209E-02	5.91504935E-02	-1.23946730E-02
26	5.31226009E-02	-4.09267661E-02	5.59590667E-02	-2.89583669E-02	5.73138244E-02	-2.20966808E-02	5.82993476E-02	-1.60938672E-02
27	5.39440323E-02	-4.03355086E-02	5.62390469E-02	-2.94680882E-02	5.71515111E-02	-2.34842761E-02	5.80062583E-02	-1.80323749E-02
28	5.04753262E-02	-4.57299653E-02	5.39317467E-02	-3.41047684E-02	5.55869650E-02	-2.74147080E-02	5.68689858E-02	-2.14771115E-02
29	5.13339490E-02	-4.51762087E-02	5.42664754E-02	-3.44715660E-02	5.54375842E-02	-2.85786802E-02	5.64938414E-02	-2.32117324E-02
ROW	CHORD 9		CHORD 10		CHORD 11		CHORD 12	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
1	6.63324958E-02	1.10000000E-03	6.63324958E-02	1.10000000E-03	0.	0.	0.	0.
2	6.63324958E-02	3.30000000E-03	6.63324958E-02	3.30000000E-03	-4.03191249E-02	2.90959820E-03	0.	0.
3	6.63324958E-02	5.50000000E-03	6.63324958E-02	5.50000000E-03	-6.37658304E-02	6.71026764E-03	-3.28298178E-03	1.16513312E-03
4	6.63324958E-02	7.70000000E-03	6.63324958E-02	7.70000000E-03	-8.77160363E-02	1.13970809E-02	-1.27975218E-02	3.64123429E-03
5	6.63324958E-02	9.90000000E-03	6.63324958E-02	9.90000000E-03	-1.03366280E-01	1.37205476E-02	-1.65889770E-02	5.33256845E-03
6	6.63324958E-02	1.21000000E-02	6.63324958E-02	1.21000000E-02	-1.20361175E-01	2.09651249E-02	-2.42536793E-02	8.65275973E-03
7	6.63324958E-02	1.43000000E-02	6.63324958E-02	1.43000000E-02	-1.31860755E-01	2.52582849E-02	-2.72308740E-02	1.06907871E-02
8	6.63324958E-02	1.65000000E-02	6.63324958E-02	1.65000000E-02	-1.44199558E-01	3.02989260E-02	-3.31316394E-02	1.39589871E-02
9	6.63324958E-02	1.87000000E-02	6.63324958E-02	1.87000000E-02	-1.52601945E-01	3.40551597E-02	-3.53058470E-02	1.58599667E-02
10	6.63324958E-02	2.09000000E-02	6.63324958E-02	2.09000000E-02	-1.61611196E-01	3.83100086E-02	-3.97414186E-02	1.89511006E-02
11	6.63324958E-02	2.31000000E-02	6.63324958E-02	2.31000000E-02	-1.67686671E-01	4.10811266E-02	-4.12400857E-02	2.05096020E-02
12	6.63324958E-02	2.53000000E-02	6.63324958E-02	2.53000000E-02	-1.74174867E-01	4.41086530E-02	-4.45073661E-02	2.31806971E-02
13	6.63324958E-02	2.75000000E-02	6.63324958E-02	2.75000000E-02	-1.78535748E-01	4.55466811E-02	-4.59494213E-02	2.42402706E-02
14	6.63324958E-02	2.97000000E-02	6.63324958E-02	2.97000000E-02	-1.83251245E-01	4.70548004E-02	-4.76900367E-02	2.63266740E-02
15	6.63324958E-02	3.19000000E-02	6.63324958E-02	3.19000000E-02	-1.86493032E-01	4.869535457E-02	-4.85573076E-02	2.87945501E-02
16	6.09175906E-02	1.84679076E-02	6.31184786E-02	2.39634324E-02	-1.90119958E-01	4.68089760E-02	-5.03699424E-02	2.82128903E-02
17	6.28204477E-02	1.85005614E-02	6.33756342E-02	2.26240652E-02	-1.90328165E-01	5.10933837E-02	-5.09286531E-02	2.80540306E-02
18	6.30009000E-02	1.64886655E-02	6.33470944E-02	2.00808602E-02	-1.90092157E-01	5.68633914E-02	-5.20635922E-02	2.92283809E-02
19	6.34150301E-02	1.47885497E-02	6.37584871E-02	1.80803960E-02	-1.86250356E-01	6.30070751E-02	-5.14282272E-02	3.03414323E-02

20	6.30432416E-02	1.03561103E-02	6.39999152E-02	1.56041868E-02	-1.86882715E-01	6.90765780E-02	-5.14090667E-02	3.22391763E-02
21	6.21532667E-02	6.29833288E-03	6.36310905E-02	1.21088638E-02	-1.84303575E-01	7.50734456E-02	-5.02526410E-02	3.34852678E-02
22	6.1492204E-02	2.07468036E-03	6.24588453E-02	7.77605235E-03	-1.83175413E-01	7.98877033E-02	-4.97309900E-02	3.53360366E-02
23	6.09647776E-02	-9.99128031E-04	6.19944156E-02	4.46565443E-03	-1.81492997E-01	8.41865910E-02	-4.92946749E-02	3.55649390E-02
24	6.02810218E-02	-4.71283641E-03	6.11197429E-02	9.56613631E-04	-1.79884154E-01	8.90572969E-02	-4.98551290E-02	3.61484949E-02
25	5.98618112E-02	-7.32674558E-03	6.07745348E-02	-1.94799443E-03	-1.77350702E-01	9.38154949E-02	-4.9326213E-02	3.64861702E-02
26	5.91174285E-02	-1.06877451E-02	5.99687379E-02	-5.04681649E-03	-1.75067883E-01	9.88635838E-02	-4.92164213E-02	3.75344509E-02
27	5.87565932E-02	-1.30221590E-02	5.96653320E-02	-7.68608120E-03	-1.71938660E-01	1.03716705E-01	-4.83770963E-02	3.80963244E-02
28	5.78690989E-02	-1.81133529E-02	5.88382305E-02	-1.05127273E-02	-1.69066770E-01	1.08791922E-01	-4.78377835E-02	3.93204031E-02
29	5.74475102E-02	-1.82508756E-02	5.85207798E-02	-1.29808019E-02	-1.65445011E-01	1.13599786E-01	-4.67697835E-02	4.00044691E-02

SAMPLE CASE --- TWO AR-2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
 WING UPPER SURFACE NORMAL WASH
 (MACH 1.200 RED. FREQ.= .50000)
 MODE SHAPE 2

PAGE CONTINUED

ROW	CHORD 13		CHORD 14		CHORD 15		CHORD 16	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
4	-3.44373466E-03	1.09582337E-03	0.	0.	0.	0.	0.	0.
5	-7.24405673E-03	2.76973062E-03	-1.94322794E-03	9.14503279E-04	0.	0.	0.	0.
6	-1.14663490E-02	4.75141796E-03	-4.66356703E-03	2.37317247E-03	-1.33621092E-03	0.03363569E-04	-9.40314423E-04	7.17394071E-04
7	-1.44401394E-02	6.72365208E-03	-6.94868535E-03	3.93003987E-03	-3.12042302E-03	2.05351456E-03	-2.20337043E-03	1.81024070E-03
8	-1.81664905E-02	9.05484645E-03	-9.47261486E-03	5.78202322E-03	-4.80292119E-03	3.43996838E-03	-3.37792331E-03	3.03931286E-03
9	-2.02131926E-02	1.09509815E-02	-1.12706190E-02	7.47864451E-03	-6.39605141E-03	4.98731558E-03	-4.53058056E-03	4.41260790E-03
10	-2.32317936E-02	1.33351588E-02	-1.33288884E-02	9.41336548E-03	-7.82872197E-03	6.54581970E-03	-5.48306080E-03	5.75757710E-03
11	-2.43529959E-02	1.49535110E-02	-1.45765125E-02	1.10071844E-02	-9.00767772E-03	8.09634408E-03	-6.35119173E-03	7.14683044E-03
12	-2.69835747E-02	1.71581298E-02	-1.61628975E-02	1.28415526E-02	-1.01029280E-02	9.62019278E-03	-7.01319218E-03	9.68264618E-03
13	-2.76730137E-02	1.83620450E-02	-1.69400700E-02	1.41529826E-02	-1.08007889E-02	1.05959437E-02	-7.60542406E-03	1.07675525E-02
14	-2.94323537E-02	2.02243594E-02	-1.81283837E-02	1.57496171E-02	-1.16726875E-02	1.23521301E-02	-8.01363646E-03	1.18087750E-02
15	-2.98841640E-02	2.09353539E-02	-1.85712314E-02	1.66833163E-02	-1.21382507E-02	1.34415927E-02	-8.39943021E-03	1.26291324E-02
16	-3.12326537E-02	2.23541072E-02	-1.94720907E-02	1.79358435E-02	-1.27051411E-02	1.45444449E-02	-8.64480352E-03	1.34087037E-02
17	-3.13438959E-02	2.25455495E-02	-1.97418835E-02	1.84647869E-02	-1.29896767E-02	1.52873042E-02	-8.89608598E-03	1.39392364E-02
18	-3.26370046E-02	2.34617603E-02	-2.04616947E-02	1.93748727E-02	-1.34213285E-02	1.60961994E-02	-9.07545007E-03	1.44530753E-02
19	-3.26359469E-02	2.36484018E-02	-2.07060065E-02	1.94860691E-02	-1.36409376E-02	1.64825459E-02	-9.30568956E-03	1.47044550E-02
20	-3.28643249E-02	2.50550314E-02	-2.11489786E-02	2.02693319E-02	-1.40345680E-02	1.69978187E-02	-9.50493914E-03	1.50860083E-02
21	-3.22014537E-02	2.55990631E-02	-2.08702972E-02	2.05794628E-02	-1.41270885E-02	1.72172288E-02	-9.52991494E-03	1.53509598E-02
22	-3.19347106E-02	2.70052876E-02	-2.07607923E-02	2.15422036E-02	-1.41230280E-02	1.74460558E-02	-9.64058979E-03	1.56800083E-02
23	-3.14155437E-02	2.75224811E-02	-2.01632499E-02	2.18333212E-02	-1.38141988E-02	1.81072346E-02	-9.72991494E-03	1.57512844E-02
24	-3.10061574E-02	2.80695849E-02	-1.98079084E-02	2.27253158E-02	-1.35438422E-02	1.87063391E-02	-9.73848176E-03	1.59323004E-02
25	-3.17087586E-02	2.76185349E-02	-2.00256666E-02	2.23575239E-02	-1.30835002E-02	1.89705713E-02	-9.07008221E-03	1.62611932E-02
26	-3.19325492E-02	2.82438447E-02	-2.07941214E-02	2.24824962E-02	-1.35689011E-02	1.89659124E-02	-8.76404768E-03	0.
27	-3.17210688E-02	2.81994459E-02	-2.09429577E-02	2.22170399E-02	-1.42786537E-02	1.83661756E-02	0.	0.
28	-3.15969918E-02	2.89866994E-02	-2.11433359E-02	2.26414863E-02	0.	0.	0.	0.

ROW	CHORD 17		CHORD 18		CHORD 19		CHORD 20	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
8	-6.80367896E-04	6.46577563E-04	0.	0.	0.	0.	0.	0.
9	-1.56809326E-03	1.63111099E-03	-5.00370367E-04	5.90366475E-04	0.	0.	0.	0.
10	-2.43016460E-03	2.72071381E-03	-1.50251448E-03	1.47036362E-03	-3.58996464E-04	5.31856260E-04	-2.56037860E-04	4.85044808E-04
11	-3.28673723E-03	3.93950114E-03	-1.75198522E-03	2.45489521E-03	-8.22442031E-04	1.32669583E-03	-5.74993281E-04	1.20117793E-03
12	-3.90668812E-03	5.13412743E-03	-2.31126880E-03	3.53736006E-03	-1.25591964E-03	2.20743583E-03	-8.47535689E-04	1.39452066E-03
13	-4.47822893E-03	6.34627040E-03	-2.75948579E-03	4.60117377E-03	-1.60958492E-03	3.17036585E-03	-1.07577483E-03	2.89952186E-03
14	-4.92396938E-03	7.49156522E-03	-3.12763024E-03	5.67425163E-03	-1.88873135E-03	4.12041401E-03	-1.22803278E-03	3.69967184E-03
15	-5.27247418E-03	8.55905455E-03	-3.39420425E-03	6.66791795E-03	-2.10191067E-03	5.06897526E-03	-1.22803278E-03	3.69967184E-03

16	-5.5382195E-03	9.52779218E-03	-3.58969376E-03	7.61387747E-03	-2.23425180E-03	5.94106762E-03	-1.32038094E-03	4.33653747E-03
17	-5.73293618E-03	1.03907450E-02	-3.72376222E-03	8.44787514E-03	-2.32040123E-03	6.76624043E-03	-1.36217350E-03	5.30244189E-03
18	-5.86998431E-03	1.11394753E-02	-3.78811879E-03	9.20303966E-03	-2.32893800E-03	7.48972831E-03	-1.32992370E-03	4.02023724E-03
19	-5.99349060E-03	1.17565379E-02	-3.85606277E-03	9.83573598E-03	-2.34520198E-03	8.14194768E-03	-1.30371396E-03	6.65024197E-03
20	-6.10941861E-03	1.22699081E-02	-3.88300303E-03	1.03769604E-02	-2.31181928E-03	8.68254021E-03	-1.22175705E-03	7.20615633E-03
21	-6.23629381E-03	1.26428171E-02	-3.94833883E-03	1.07999108E-02	-2.31551503E-03	9.14655337E-03	-1.17658478E-03	7.67300306E-03
22	-6.38949685E-03	1.29393270E-02	-4.01009098E-03	1.11378211E-02	-2.30565947E-03	9.50465000E-03	-1.10604250E-03	8.06202447E-03
23	-6.48741106E-03	1.31748323E-02	-4.13444493E-03	1.13768915E-02	-2.35466728E-03	9.79888686E-03	-1.09350927E-03	8.37857866E-03
24	-6.58760843E-03	1.34638435E-02	-4.17073026E-03	1.15987044E-02	-2.42197068E-03	1.00047719E-02	0.	0.
25	-6.83331690E-03	1.36528040E-02	0.	0.	0.	0.	0.	0.

SAMPLE CASE --- TWO AREA SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

WING UPPER SURFACE NORMAL WASH

(MACH 1.200 RED. FREQ. 2 .50000)

MODE SHAPE 2

PAGE CONTINUED

ROW	CHORD 21		CHORD 22		CHORD 23		CHORD 24	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
12	-1.7471704E-04	4.41134579E-04	0.	0.	0.	0.	0.	0.
13	-3.6233701E-04	1.08787407E-03	-1.10360272E-04	4.00858990E-04	0.	0.	0.	0.
14	-3.42889730E-04	1.80196221E-03	-2.29793119E-04	9.83831432E-04	-5.87005649E-05	3.63373162E-04	0.	0.
15	-6.59348084E-04	2.57422456E-03	-3.02209597E-04	1.62445034E-03	-1.07912734E-04	8.87690371E-04	0.	0.
16	-7.10895802E-04	3.32035161E-03	-3.30188758E-04	2.31228805E-03	-1.07120E-04	1.46606029E-03	-2.04470693E-05	3.28139270E-04
17	-7.21670135E-04	4.03775393E-03	-3.13242567E-04	2.97201933E-03	-7.64634981E-05	2.07068664E-03	-1.287171C2E-03	7.97961800E-04
18	-6.56274602E-04	4.72743578E-03	-2.24045485E-04	3.61911566E-03	3.07807933E-05	2.65147306E-03	2.68933821E-03	1.30838854E-03
19	-5.95249172E-04	5.35402309E-03	-1.38806338E-04	4.20600277E-03	1.36125841E-04	3.2202310E-03	1.43172505E-04	1.8463031E-03
20	-4.77520700E-04	5.89223467E-03	1.08203342E-05	4.74381945E-03	3.05080343E-04	3.72424214E-03	2.56141846E-04	2.35903830E-03
21	-3.48237274E-04	6.37059690E-03	1.30843501E-04	5.21282666E-03	0.	0.	0.	0.
22	-2.72889739E-04	6.76716540E-03	0.	0.	0.	0.	0.	0.

ROW	CHORD 25		CHORD 26	
	REAL	IMAGINARY	REAL	IMAGINARY
16	2.11835982E-05	2.95121951E-04	0.	0.
17	6.25336506E-05	7.14094872E-04	3.34829454E-05	2.63851131E-04
18	1.82643847E-04	1.16460802E-03	0.	0.

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
 TAIL UPPER SURFACE NORMAL WASH
 (MACH 1.200 REF. FREQ.= .50000)
 MODE SHAPE 2

ROW	CHORD 1		CHORD 2		CHORD 3		CHORD 4	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
19	-3.9117621E-02	-2.47284530E-02	-5.64920791E-02	-2.46367453E-02	-5.72554692E-02	-2.4409793E-02	-5.51334895E-02	-2.42748900E-02
20	-2.48461090E-02	7.62604131E-03	-2.46804922E-02	7.66129327E-03	-2.53310010E-02	5.95447091E-03	-2.63939790E-02	2.21401063E-03
21	-3.06898118E-02	6.83011666E-03	-3.13032001E-02	4.88595339E-03	-3.08401114E-02	3.69553450E-03	-3.00202668E-02	2.03175787E-03
22	-2.96068597E-02	1.20080762E-02	-2.86532315E-02	1.28432359E-02	-2.85890768E-02	1.16241815E-02	-2.81264626E-02	9.73185104E-03
23	-3.39291760E-02	1.22402512E-02	-3.40362409E-02	1.10930708E-02	-3.32165604E-02	9.95689165E-03	-3.21064546E-02	8.10891403E-03
24	-3.11302725E-02	1.77250011E-02	-3.05313275E-02	1.76149357E-02	-3.01158143E-02	1.66364085E-02	-2.93183322E-02	1.47214772E-02
25	-3.53730719E-02	1.76619096E-02	-3.53764304E-02	1.66124153E-02	-3.45626225E-02	1.51083640E-02	-3.33954026E-02	1.29380666E-02
26	-3.16113445E-02	2.23868962E-02	-3.12358647E-02	2.19958548E-02	-3.03487004E-02	2.09051624E-02	-2.93686729E-02	1.88433564E-02
27	-3.56790652E-02	2.81230891E-02	-3.53209339E-02	2.11576834E-02	-3.42314106E-02	1.96584595E-02	-3.28932760E-02	1.73115522E-02
28	-3.11340922E-02	2.60897632E-02	-3.04566532E-02	2.56364876E-02	-2.94349385E-02	2.45211349E-02	-2.84291956E-02	2.24151503E-02
29	-3.45387728E-02	2.58209947E-02	-3.42202784E-02	2.49951715E-02	-3.31852087E-02	2.35320405E-02	-3.18147717E-02	2.13563401E-02
30	-2.93787437E-02	2.91033778E-02	-2.88783485E-02	2.85903589E-02	-2.79622851E-02	2.74651740E-02	-2.70761231E-02	2.53743520E-02
31	-3.26955241E-02	2.89805176E-02	-3.23820660E-02	2.81842294E-02	-3.15038605E-02	2.67441671E-02	-3.03207664E-02	2.45837878E-02
32	-2.72462480E-02	3.15949597E-02	-2.68237195E-02	3.10318318E-02	-2.60198742E-02	2.98457003E-02	-2.53350361E-02	2.77613574E-02
33	-3.05258394E-02	3.16924432E-02	-3.02489669E-02	3.09226013E-02	-2.94557784E-02	2.94746310E-02	-2.84195966E-02	2.72679793E-02

ROW	CHORD 5		CHORD 6		CHORD 7		CHORD 8	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
19	-3.21233318E-02	-2.39568663E-02	-4.79293684E-02	-2.34355827E-02	-4.23363163E-02	-2.26869769E-02	-3.49877890E-02	-2.12205154E-02
20	-2.65862529E-02	-1.01561359E-03	-2.58952780E-02	-4.15937027E-03	-2.45644304E-02	-7.96759246E-03	-2.16729183E-02	-1.07745919E-02
21	-2.92213121E-02	-5.47973131E-04	-2.75445926E-02	-2.77955362E-03	-2.48483338E-02	-5.32290548E-03	-2.16808426E-02	-8.90376015E-03
22	-2.73428720E-02	7.10127731E-03	-2.67000506E-02	2.88845795E-03	-2.49032017E-02	-1.75247646E-03	-2.05845508E-02	-5.28808725E-03
23	-3.04962510E-02	5.80160599E-03	-2.80354560E-02	3.08336269E-03	-2.45566608E-02	-3.51993370E-06	-2.00419223E-02	-3.50318960E-03
24	-2.83361607E-02	1.16844286E-02	-2.64446779E-02	7.89827475E-03	-2.36806037E-02	3.49289337E-03	-1.92031360E-02	-6.75563461E-04
25	-3.10470633E-02	1.05926293E-02	-2.79477669E-02	7.71316302E-03	-2.38625376E-02	4.38741259E-03	-1.87358939E-02	6.65827359E-04
26	-2.81167079E-02	1.57413325E-02	-2.61050249E-02	1.18386578E-02	-2.59706839E-02	7.38917661E-03	-1.80625267E-02	2.98958883E-03
27	-3.05765608E-02	1.48822223E-02	-2.75236357E-02	1.16549883E-02	-2.34166344E-02	7.92846070E-03	-1.81323607E-02	3.84393348E-03
28	-2.71303877E-02	1.93245717E-02	-2.51602003E-02	1.53230418E-02	-2.21926048E-02	1.06830386E-02	-1.73781630E-02	5.90809682E-03
29	-2.95788211E-02	1.85767642E-02	-2.66038760E-02	1.51246384E-02	-2.26697374E-02	1.10511189E-02	-1.75375588E-02	6.57607053E-03
30	-2.58359739E-02	2.22959332E-02	-2.39751045E-02	1.82512376E-02	-2.11095975E-02	1.34970067E-02	-1.65410037E-02	8.40865099E-03
31	-2.82808373E-02	2.17110618E-02	-2.54841924E-02	1.80687860E-02	-2.17556449E-02	1.37261493E-02	-1.67579133E-02	8.91770117E-03
32	-2.42778166E-02	2.47252123E-02	-2.26228002E-02	2.06877060E-02	-1.9585437E-02	1.58357227E-02	-1.56430476E-02	1.04866373E-02
33	-2.86663884E-02	2.45311480E-02	-2.41465345E-02	2.03405601E-02	-2.07147208E-02	1.67003373E-02	-1.59561423E-02	1.08896999E-02

ROW	CHORD 9		CHORD 10		CHORD 11		CHORD 12	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY

19	-2.55259243E-02	-1.97606723E-02	-1.05726275E-02	-1.89298755E-02	0.	1.10794454E-02	1.05953449E-02	0.	0.
20	-1.61412604E-02	-1.24913695E-02	-9.80053148E-03	-1.36382870E-02	-1.36382870E-02	1.78471129E-02	1.25946556E-02	1.16311944E-03	6.26503715E-04
21	-1.74744325E-02	-1.72599814E-02	-1.02335305E-02	-1.41845819E-02	-1.41845819E-02	2.81171846E-02	1.46408357E-02	4.61711867E-03	2.28438453E-03
22	-1.37567273E-02	-9.3159556E-03	-9.84003631E-03	-1.35098295E-02	-1.35098295E-02	3.60612125E-02	1.27059992E-02	6.58320169E-03	1.58140658E-03
23	-1.49396138E-02	-8.09035379E-03	-8.38649828E-03	-1.12506059E-02	-1.12506059E-02	4.17276699E-02	7.21638329E-03	1.04482331E-02	1.59428211E-03
24	-1.39520516E-02	-9.78332840E-03	-7.17765922E-03	-9.56439179E-03	-9.56439179E-03	4.41397703E-02	-7.34915422E-04	1.15768873E-02	-6.45210097E-04
25	-1.35225999E-02	-7.42223203E-03	-6.76934638E-03	-8.26180481E-03	-8.26180481E-03	4.69572737E-02	-8.37461989E-03	1.33856852E-02	-2.76231588E-03
26	-1.26624036E-02	-2.53040020E-03	-5.87722163E-03	-6.93507078E-03	-6.93507078E-03	4.63529623E-02	-1.83111331E-02	1.53836910E-02	-5.82981572E-03
27	-1.25910156E-02	-1.54149850E-03	-5.62035202E-03	-5.90032363E-03	-5.90032363E-03	4.51872967E-02	-2.74816020E-02	1.39189235E-02	-8.64355010E-03
28	-1.20677130E-02	-2.56646955E-05	-5.06859491E-03	-4.90644161E-03	-4.90644161E-03	4.27370943E-02	-3.64239780E-02	1.24981795E-02	-1.18814964E-02
29	-1.21529390E-02	7.74236549E-04	-5.20730714E-03	-4.10311505E-03	-4.10311505E-03				

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
 TAIL UPPER SURFACE NORMAL WASH

(MACH 1.200 RED. FREQ.= .50000)

MODE SHAPE 2

PAGE CONTINUED

ROW	CHORD 9		CHORD 10		CHORD 11		CHORD 12	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
30	-1.15607068E-02	2.09637463E-03	-4.75938727E-03	-3.29669267E-03	4.09917563E-02	-4.49303773E-02	1.20979134E-02	-1.46844504E-02
31	-1.18434588E-02	2.75538040E-03	-4.91897138E-03	-2.60756560E-03	3.69329560E-02	-5.27032714E-02	1.04671532E-02	-1.75665544E-02
32	-1.08709377E-02	3.67724276E-03	-4.46767084E-03	-1.96746854E-03	3.39315537E-02	-5.99058115E-02	0.	0.
33	-1.11438713E-02	4.41252218E-03	-4.63870147E-03	-1.39306731E-03	0.	0.	0.	0.
ROW	CHORD 13		CHORD 14		CHORD 15		CHORD 16	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
22	1.18454473E-03	6.89342520E-04	0.	0.	0.	0.	0.	0.
23	2.75826323E-03	9.38935130E-04	7.51679787E-03	3.13609382E-04	0.	0.	0.	0.
24	4.80393318E-03	6.93452950E-04	1.94020972E-03	4.52327671E-04	5.66505863E-04	1.70718127E-04	4.41015186E-04	6.17972186E-05
25	6.59107323E-03	1.00952742E-04	3.1988254E-03	1.39905863E-04	1.41981786E-03	1.59598195E-04	1.09673289E-03	2.23414305E-06
26	8.03644877E-03	-1.11506897E-03	4.65586578E-03	-4.09409200E-04	2.36709161E-03	-1.50783785E-04	1.82619153E-03	-3.29137268E-04
27	6.40911009E-03	-3.04584950E-03	5.33423333E-03	-1.53631813E-03	3.36667695E-03	-7.25777348E-04	0.	0.
28	9.03934072E-03	-4.89832448E-03	5.84244144E-03	-2.82022265E-03	3.92398829E-03	-1.60534938E-03	0.	0.
29	8.44787231E-03	-7.14273830E-03	5.81398097E-03	-4.34614532E-03	0.	0.	0.	0.
30	8.02748330E-03	-9.13926582E-03	0.	0.	0.	0.	0.	0.

ROW	CHORD 17	
	REAL	IMAGINARY
28	3.54947168E-04	2.67341335E-05

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
 WING VELOCITY POTENTIALS
 (MACH 1.200 REF. FREQ. = .50000)
 MODE SHAPE 2

ROW	CHORD 1		CHORD 2		CHORD 3		CHORD 4	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
1	-6.63184769E-02	6.99825447E-04	-6.63184769E-02	6.99825447E-04	-6.63184769E-02	6.99825447E-04	-6.63184769E-02	6.99825447E-04
2	-1.97801526E-01	1.06032274E-02	-1.97801526E-01	1.06032274E-02	-1.97801526E-01	1.06032274E-02	-1.97801526E-01	1.06032274E-02
3	-3.27215061E-01	3.02371871E-02	-3.27215061E-01	3.02371871E-02	-3.27215061E-01	3.02371871E-02	-3.27215061E-01	3.02371871E-02
4	-4.52690913E-01	5.90062499E-02	-4.52690913E-01	5.90062499E-02	-4.52690913E-01	5.90062499E-02	-4.52690913E-01	5.90062499E-02
5	-5.72820386E-01	9.61220392E-02	-5.72820386E-01	9.61220392E-02	-5.72820386E-01	9.61220392E-02	-5.72820386E-01	9.61220392E-02
6	-6.86349473E-01	1.40572481E-01	-6.86349473E-01	1.40572481E-01	-6.86349473E-01	1.40572481E-01	-6.86349473E-01	1.40572481E-01
7	-7.92220506E-01	1.91153573E-01	-7.92220506E-01	1.91153573E-01	-7.92220506E-01	1.91153573E-01	-7.92220506E-01	1.91153573E-01
8	-8.89601234E-01	2.46507223E-01	-8.89601234E-01	2.46507223E-01	-8.89601234E-01	2.46507223E-01	-8.89601234E-01	2.46507223E-01
9	-9.77902331E-01	3.05162316E-01	-9.77902331E-01	3.05162316E-01	-9.77902331E-01	3.05162316E-01	-9.77902331E-01	3.05162316E-01
10	-1.05679945E+00	3.65581916E-01	-1.04945283E+00	3.44639787E-01	-1.02901545E+00	3.29063251E-01	-9.97376643E-01	2.96151347E-01
11	-1.12041939E+00	4.15341562E-01	-1.10393729E+00	3.89161759E-01	-1.07795893E+00	3.55321572E-01	-1.04147231E+00	3.16894320E-01
12	-1.16482969E+00	4.34737154E-01	-1.14839280E+00	4.14327181E-01	-1.11856163E+00	3.75043558E-01	-1.07837117E+00	3.31908032E-01
13	-1.19683799E+00	4.34075395E-01	-1.18244982E+00	4.20748755E-01	-1.15273527E+00	3.87880203E-01	-1.10961148E+00	3.41279931E-01
14	-1.22347731E+00	4.14853563E-01	-1.20910451E+00	4.04809431E-01	-1.18025890E+00	3.83665452E-01	-1.13651259E+00	3.45050803E-01
15	-1.24801429E+00	3.83210227E-01	-1.23336063E+00	3.75190134E-01	-1.20395529E+00	3.58850050E-01	-1.15961275E+00	3.32953755E-01
ROW	CHORD 5		CHORD 6		CHORD 7		CHORD 8	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
1	-6.63184769E-02	6.99825447E-04	-6.63184769E-02	6.99825447E-04	-6.63184769E-02	6.99825447E-04	-6.63184769E-02	6.99825447E-04
2	-1.97801526E-01	1.06032274E-02	-1.97801526E-01	1.06032274E-02	-1.97801526E-01	1.06032274E-02	-1.97801526E-01	1.06032274E-02
3	-3.27215061E-01	3.02371871E-02	-3.27215061E-01	3.02371871E-02	-3.27215061E-01	3.02371871E-02	-3.27215061E-01	3.02371871E-02
4	-4.52690913E-01	5.90062499E-02	-4.52690913E-01	5.90062499E-02	-4.52690913E-01	5.90062499E-02	-4.52690913E-01	5.90062499E-02
5	-5.72820386E-01	9.61220392E-02	-5.72820386E-01	9.61220392E-02	-5.72820386E-01	9.61220392E-02	-5.72820386E-01	9.61220392E-02
6	-6.86349473E-01	1.40572481E-01	-6.86349473E-01	1.40572481E-01	-6.86349473E-01	1.40572481E-01	-6.86349473E-01	1.40572481E-01
7	-7.92220506E-01	1.91153573E-01	-7.92220506E-01	1.91153573E-01	-7.92220506E-01	1.91153573E-01	-7.92220506E-01	1.91153573E-01
8	-8.89601234E-01	2.46507223E-01	-8.89601234E-01	2.46507223E-01	-8.89601234E-01	2.46507223E-01	-8.89601234E-01	2.46507223E-01
9	-9.77902331E-01	3.05162316E-01	-9.77902331E-01	3.05162316E-01	-9.77902331E-01	3.05162316E-01	-9.77902331E-01	3.05162316E-01
10	-1.05679945E+00	3.65581916E-01	-1.04945283E+00	3.44639787E-01	-1.02901545E+00	3.29063251E-01	-9.97376643E-01	2.96151347E-01
11	-1.12041939E+00	4.15341562E-01	-1.10393729E+00	3.89161759E-01	-1.07795893E+00	3.55321572E-01	-1.04147231E+00	3.16894320E-01
12	-1.16482969E+00	4.34737154E-01	-1.14839280E+00	4.14327181E-01	-1.11856163E+00	3.75043558E-01	-1.07837117E+00	3.31908032E-01
13	-1.19683799E+00	4.34075395E-01	-1.18244982E+00	4.20748755E-01	-1.15273527E+00	3.87880203E-01	-1.10961148E+00	3.41279931E-01
14	-1.22347731E+00	4.14853563E-01	-1.20910451E+00	4.04809431E-01	-1.18025890E+00	3.83665452E-01	-1.13651259E+00	3.45050803E-01
15	-1.24801429E+00	3.83210227E-01	-1.23336063E+00	3.75190134E-01	-1.20395529E+00	3.58850050E-01	-1.15961275E+00	3.32953755E-01
ROW	CHORD 9		CHORD 10					
	REAL	IMAGINARY	REAL	IMAGINARY				
1	-6.63184769E-02	6.99825447E-04	-6.63184769E-02	6.99825447E-04				
2	-1.97801526E-01	1.06032274E-02	-1.97801526E-01	1.06032274E-02				
3	-3.27215061E-01	3.02371871E-02	-3.27215061E-01	3.02371871E-02				
4	-4.52690913E-01	5.90062499E-02	-4.52690913E-01	5.90062499E-02				
5	-5.72820386E-01	9.61220392E-02	-5.72820386E-01	9.61220392E-02				
6	-6.86349473E-01	1.40572481E-01	-6.86349473E-01	1.40572481E-01				
7	-7.92220506E-01	1.91153573E-01	-7.92220506E-01	1.91153573E-01				
8	-8.89601234E-01	2.46507223E-01	-8.89601234E-01	2.46507223E-01				
9	-9.77902331E-01	3.05162316E-01	-9.77902331E-01	3.05162316E-01				
10	-1.05679945E+00	3.65581916E-01	-1.04945283E+00	3.44639787E-01				
11	-1.12041939E+00	4.15341562E-01	-1.10393729E+00	3.89161759E-01				
12	-1.16482969E+00	4.34737154E-01	-1.14839280E+00	4.14327181E-01				
13	-1.19683799E+00	4.34075395E-01	-1.18244982E+00	4.20748755E-01				
14	-1.22347731E+00	4.14853563E-01	-1.20910451E+00	4.04809431E-01				
15	-1.24801429E+00	3.83210227E-01	-1.23336063E+00	3.75190134E-01				

1	-C.C3164769E-02	6.99623447E-04	-6.63184769E-02	6.39623447E-04
2	-1.97801926E-01	1.06032274E-02	-1.57587972E-01	6.6026028E-03
3	-2.9977444E-01	2.4386611E-02	-2.08731114E-01	1.19663897E-02
4	-3.47333527E-01	3.57494077E-02	-2.48334137E-01	1.76262658E-02
5	-4.21240420E-01	4.67293668E-02	-2.60706427E-01	2.34318670E-02
6	-4.63629337E-01	5.73101329E-02	-3.08069376E-01	2.92289346E-02
7	-5.03437442E-01	6.72248226E-02	-3.30969647E-01	3.46723412E-02
8	-5.33232699E-01	7.61603875E-02	-3.50743423E-01	3.96982473E-02
9	-5.63332130E-01	8.38794754E-02	-3.67349192E-01	4.39258200E-02
10	-5.83231425E-01	9.00364358E-02	-3.81661317E-01	4.73447109E-02
11	-6.04717836E-01	9.44563009E-02	-3.93838286E-01	4.96062124E-02

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

WING VELOCITY POTENTIALS
(MACH 1.200 REP. FREQ. 2 .50000)
MODE SHAPE 2

PAGE CONTINUED

ROW	CHORD 9		CHORD 10	
	REAL	IMAGINARY	REAL	IMAGINARY
12	-6.2120969E-01	9.6661733E-02	-4.0394936E-01	5.07496370E-02
13	-6.35321693E-01	9.71596346E-02	-4.12613392E-01	5.04889462E-02
14	-6.47431767E-01	9.51715199E-02	-4.20176166E-01	4.89172886E-02
15	-6.56142073E-01	9.09024242E-02	-4.26709994E-01	4.58274281E-02

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
 TAIL VELOCITY POTENTIALS
 (MACH 1.200 RED. FREQ.= .50000)
 MODE SHAPE 2

ROW	CHORD 1		CHORD 2		CHORD 3		CHORD 4	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
19	5.97433780E-02	2.31088310E-02	5.91216173E-02	2.30340104E-02	5.78806629E-02	2.28403074E-02	5.57754206E-02	2.27629510E-02
20	1.46470080E-01	2.78757585E-02	1.45037517E-01	2.77702938E-02	1.43125207E-01	2.94306397E-02	1.40045624E-01	3.31633313E-02
21	2.01552933E-01	-5.37579813E-03	2.01274448E-01	-2.32686529E-03	1.99487290E-01	2.62429453E-03	1.96206356E-01	1.04770871E-02
22	2.56580431E-01	-4.70753809E-02	2.55802374E-01	-4.30367946E-02	2.54021831E-01	-3.50384370E-02	2.49149730E-01	-2.41029791E-02
23	3.08448622E-01	-9.74763524E-02	3.06874214E-01	-9.26085988E-02	3.03440111E-01	-8.27094581E-02	2.97336699E-01	-6.75252971E-02
24	3.54545105E-01	-1.59559509E-01	3.52153682E-01	-1.49457387E-01	3.47013340E-01	-1.37037780E-01	3.38513662E-01	-1.17763929E-01
25	3.93230914E-01	-2.19259555E-01	3.90080554E-01	-2.11507473E-01	3.83344452E-01	-1.95719526E-01	3.72801632E-01	-1.71411789E-01
26	4.23409796E-01	-2.65771694E-01	4.19514982E-01	-2.57976075E-01	4.11430948E-01	-2.56308588E-01	3.94204272E-01	-2.28690102E-01
27	4.48295615E-01	-3.52018374E-01	4.39733714E-01	-3.40039537E-01	4.25972110E-01	-3.17893534E-01	4.02342843E-01	-2.87754882E-01
28	4.55780187E-01	-4.14891134E-01	4.46458305E-01	-4.01960084E-01	4.27108548E-01	-3.78338436E-01	4.01364477E-01	-3.45836762E-01
29	4.53332338E-01	-4.72205637E-01	4.40185045E-01	-4.59374498E-01	4.19285624E-01	-4.35215577E-01	3.91594525E-01	-4.00829088E-01
30	4.37143359E-01	-5.22131407E-01	4.25676392E-01	-5.10412302E-01	4.03243902E-01	-4.86273229E-01	3.74763314E-01	-4.50826031E-01
31	4.09093927E-01	-5.64011702E-01	4.00473676E-01	-5.53005807E-01	3.81129364E-01	-5.29899932E-01	3.53722985E-01	-4.94311337E-01
32	3.74888221E-01	-5.97697386E-01	3.66908708E-01	-5.86474744E-01	3.52562795E-01	-5.64599112E-01	3.29866176E-01	-5.30358871E-01
33	3.36840700E-01	-6.22788219E-01	3.31036645E-01	-6.12047491E-01	3.19023321E-01	-5.90246446E-01	3.01479551E-01	-5.57447765E-01
ROW	CHORD 5		CHORD 6		CHORD 7		CHORD 8	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
19	5.27392407E-02	2.25273225E-02	4.85333313E-02	2.21201277E-02	4.29256515E-02	2.15236602E-02	3.55402098E-02	2.02975324E-02
20	1.34119693E-01	3.64134763E-02	1.24965086E-01	3.94374982E-02	1.12265330E-01	4.27409130E-02	9.44370744E-02	4.47694904E-02
21	1.90130695E-01	2.03093159E-02	1.78897556E-01	3.00371568E-02	1.62090265E-01	4.03214375E-02	1.39089447E-01	5.11807962E-02
22	2.41044741E-01	-9.58081912E-03	2.28247085E-01	8.29327055E-03	2.08086990E-01	2.76731701E-02	1.75224443E-01	4.26959113E-02
23	2.85599371E-01	-4.77504760E-02	2.70733635E-01	-2.26849413E-02	2.43073813E-01	2.25471358E-03	2.00846176E-01	2.11704993E-02
24	3.25826922E-01	-9.13774307E-02	3.01548347E-01	-6.24702939E-02	2.65034587E-01	-3.42866084E-02	2.19522689E-01	-8.53602719E-03
25	3.52350933E-01	-1.41518821E-01	3.20329212E-01	-1.09050184E-01	2.79463213E-01	-7.58958706E-02	2.29167812E-01	-4.44157259E-02
26	3.68271073E-01	-1.95777859E-01	3.30605361E-01	-1.58984353E-01	2.86127406E-01	-1.20475048E-01	2.32058896E-01	-8.28144432E-02
27	3.71596353E-01	-2.51240008E-01	3.32634184E-01	-2.10039671E-01	2.89300485E-01	-1.666149224E-01	2.31124392E-01	-1.21215542E-01
28	3.6808991E-01	-3.05952223E-01	3.27263805E-01	-2.60338892E-01	2.80059837E-01	-2.10677344E-01	2.25905927E-01	-1.58884346E-01
29	3.57083773E-01	-3.57939345E-01	3.16924269E-01	-3.08107460E-01	2.70517188E-01	-2.53121799E-01	2.17564360E-01	-1.94826837E-01
30	3.41233763E-01	-4.05593333E-01	3.02268739E-01	-3.52236864E-01	2.57429565E-01	-2.92580919E-01	2.06580637E-01	-2.28348195E-01
31	3.21339231E-01	-4.47771330E-01	2.84285562E-01	-3.91774332E-01	2.41727585E-01	-3.28257770E-01	1.93798630E-01	-2.58994292E-01
32	2.98267486E-01	-4.83436870E-01	2.64207216E-01	-4.25935886E-01	2.24422800E-01	-3.59638340E-01	1.79326366E-01	-2.86202224E-01
33	2.78979176E-01	-5.12058128E-01	2.43281831E-01	-4.54115447E-01	2.06002668E-01	-3.86107239E-01	1.63776446E-01	-3.09499650E-01
ROW	CHORD 9		CHORD 10					
	REAL	IMAGINARY	REAL	IMAGINARY				

19	2.604500E-02	1.90352832E-02	1.70750952E-02	1.8467070E-02
20	7.15521769E-02	4.57770154E-02	4.11306969E-02	3.68189604E-02
21	1.04603119E-01	5.50223982E-02	5.71184535E-02	4.02919310E-02
22	1.50132856E-01	4.93833270E-02	7.50743746E-02	4.05107805E-02
23	1.50055431E-01	3.39158235E-02	8.70721134E-02	3.16234622E-02
24	1.63251455E-01	1.09206750E-02	9.34423749E-02	1.87325030E-02
25	1.69862355E-01	-1.69498835E-02	9.78418707E-02	2.79335586E-03
26	1.71529728E-01	-4.68245435E-02	9.94733429E-02	-1.80265241E-02
27	1.69835881E-01	7.88091329E-02	9.75119620E-02	-3.68526362E-02
28	1.68825395E-01	-1.06053035E-01	9.42648411E-02	-3.49432734E-02
29	1.58868996E-01	-1.33989080E-01	9.07389581E-02	-7.21829427E-02

SAMPLE CASE ---- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

TAIL VELOCITY POTENTIALS
 (MACH 1.200 RED. FREQ.= .50000)
 MODE SHAPE 2

PAGE CONTINUED

ROW	CHORD 9		CHORD 10	
	REAL	IMAGINARY	REAL	IMAGINARY
30	1.50820379E-01	-1.60200176E-01	8.55962221E-02	-8.84275988E-02
31	1.40998919E-01	-1.84285674E-01	7.94168208E-02	-1.03457324E-01
32	1.29783770E-01	-2.05856326E-01	7.26917087E-02	-1.16966695E-01
33	1.17831936E-01	-2.24565247E-01	6.52551056E-02	-1.28768131E-01

ENTERING PROGRAM CHORDS CURRENT ELAPSED TIME IS CP = 122.634, PP = 56.058

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

SMOOTHED WING VELOCITY POTENTIALS
(MACH 1.200 RED. FREQ. = .50000)

MODE SHAPE 2

ROW	CHORD 1			CHORD 2			CHORD 3			CHORD 4		
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY	
1	-6.63184769E-02	6.99825447E-04		-6.63184769E-02	6.99825447E-04		-6.63184769E-02	6.99825447E-04		-6.63184769E-02	6.99825447E-04	
2	-1.97118157E-01	1.27157599E-02		-1.97118157E-01	1.27157599E-02		-1.97118157E-01	1.27157599E-02		-1.97118157E-01	1.27157599E-02	
3	-3.24536639E-01	2.78274530E-02		-3.24536639E-01	2.78274530E-02		-3.24536639E-01	2.78274530E-02		-3.24536639E-01	2.78274530E-02	
4	-4.48830232E-01	5.23722576E-02		-4.48830232E-01	5.23722576E-02		-4.48830232E-01	5.23722576E-02		-4.48830232E-01	5.23722576E-02	
5	-5.69164048E-01	8.88699981E-02		-5.69164048E-01	8.88699981E-02		-5.69164048E-01	8.88699981E-02		-5.69164048E-01	8.88699981E-02	
6	-6.84089498E-01	1.36746250E-01		-6.84089498E-01	1.36746250E-01		-6.84089498E-01	1.36746250E-01		-6.84089498E-01	1.36746250E-01	
7	-7.91721792E-01	1.93055618E-01		-7.91721792E-01	1.93055618E-01		-7.91721792E-01	1.93055618E-01		-7.91721792E-01	1.93055618E-01	
8	-8.90037439E-01	2.53205022E-01		-8.90037439E-01	2.53205022E-01		-8.90037439E-01	2.53205022E-01		-8.90037439E-01	2.53205022E-01	
9	-9.77151748E-01	3.11676968E-01		-9.77151748E-01	3.11676968E-01		-9.77151748E-01	3.11676968E-01		-9.77151748E-01	3.11676968E-01	
10	-1.05159632E+00	3.62752835E-01		-1.05159632E+00	3.62752835E-01		-1.05159632E+00	3.62752835E-01		-1.05159632E+00	3.62752835E-01	
11	-1.11239656E+00	4.01236152E-01		-1.11239656E+00	4.01236152E-01		-1.11239656E+00	4.01236152E-01		-1.11239656E+00	4.01236152E-01	
12	-1.16034915E+00	4.23175879E-01		-1.16034915E+00	4.23175879E-01		-1.16034915E+00	4.23175879E-01		-1.16034915E+00	4.23175879E-01	
13	-1.19629959E+00	4.26589686E-01		-1.19629959E+00	4.26589686E-01		-1.19629959E+00	4.26589686E-01		-1.19629959E+00	4.26589686E-01	
14	-1.22341965E+00	4.12187232E-01		-1.22341965E+00	4.12187232E-01		-1.22341965E+00	4.12187232E-01		-1.22341965E+00	4.12187232E-01	
15	-1.24648491E+00	3.84093449E-01		-1.24648491E+00	3.84093449E-01		-1.24648491E+00	3.84093449E-01		-1.24648491E+00	3.84093449E-01	

ROW	CHORD 5			CHORD 6			CHORD 7			CHORD 8		
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY	
1	-6.63184769E-02	6.99825447E-04		-6.63184769E-02	6.99825447E-04		-6.63184769E-02	6.99825447E-04		-6.63184769E-02	6.99825447E-04	
2	-1.97118157E-01	1.27157599E-02		-1.97118157E-01	1.27157599E-02		-1.97118157E-01	1.27157599E-02		-1.97118157E-01	1.27157599E-02	
3	-3.24536639E-01	2.78274530E-02		-3.24536639E-01	2.78274530E-02		-3.24536639E-01	2.78274530E-02		-3.24536639E-01	2.78274530E-02	
4	-4.48830232E-01	5.23722576E-02		-4.48830232E-01	5.23722576E-02		-4.48830232E-01	5.23722576E-02		-4.48830232E-01	5.23722576E-02	
5	-5.69164048E-01	8.88699981E-02		-5.69164048E-01	8.88699981E-02		-5.69164048E-01	8.88699981E-02		-5.69164048E-01	8.88699981E-02	
6	-6.84089498E-01	1.36746250E-01		-6.84089498E-01	1.36746250E-01		-6.84089498E-01	1.36746250E-01		-6.84089498E-01	1.36746250E-01	
7	-7.91721792E-01	1.93055618E-01		-7.91721792E-01	1.93055618E-01		-7.91721792E-01	1.93055618E-01		-7.91721792E-01	1.93055618E-01	
8	-8.90037439E-01	2.53205022E-01		-8.90037439E-01	2.53205022E-01		-8.90037439E-01	2.53205022E-01		-8.90037439E-01	2.53205022E-01	
9	-9.77151748E-01	3.11676968E-01		-9.77151748E-01	3.11676968E-01		-9.77151748E-01	3.11676968E-01		-9.77151748E-01	3.11676968E-01	
10	-1.05159632E+00	3.62752835E-01		-1.05159632E+00	3.62752835E-01		-1.05159632E+00	3.62752835E-01		-1.05159632E+00	3.62752835E-01	
11	-1.11239656E+00	4.01236152E-01		-1.11239656E+00	4.01236152E-01		-1.11239656E+00	4.01236152E-01		-1.11239656E+00	4.01236152E-01	
12	-1.16034915E+00	4.23175879E-01		-1.16034915E+00	4.23175879E-01		-1.16034915E+00	4.23175879E-01		-1.16034915E+00	4.23175879E-01	
13	-1.19629959E+00	4.26589686E-01		-1.19629959E+00	4.26589686E-01		-1.19629959E+00	4.26589686E-01		-1.19629959E+00	4.26589686E-01	
14	-1.22341965E+00	4.12187232E-01		-1.22341965E+00	4.12187232E-01		-1.22341965E+00	4.12187232E-01		-1.22341965E+00	4.12187232E-01	
15	-1.24648491E+00	3.84093449E-01		-1.24648491E+00	3.84093449E-01		-1.24648491E+00	3.84093449E-01		-1.24648491E+00	3.84093449E-01	

CHORD 9 CHORD 10
REAL IMAGINARY REAL IMAGINARY

1	-6.63184769E-02	6.99825447E-04	-6.63184769E-02	6.99825447E-04
2	-1.87618875E-01	1.15968823E-02	-1.43343422E-01	6.43275503E-03
3	-2.82095229E-01	2.30677982E-02	-2.03047262E-01	1.20415075E-02
4	-3.57193181E-01	3.47372587E-02	-2.45865286E-01	1.76860369E-02
5	-4.13783190E-01	4.62349894E-02	-2.78647672E-01	2.33853878E-02
6	-4.62303815E-01	5.72043569E-02	-3.04869815E-01	2.90472125E-02
7	-4.99755000E-01	6.73110300E-02	-3.28854662E-01	3.44972878E-02
8	-5.30491397E-01	7.62316302E-02	-3.49993044E-01	3.95090314E-02
9	-5.56265455E-01	8.37623820E-02	-3.62964003E-01	4.38330190E-02
10	-5.78331082E-01	8.96277688E-02	-3.77955130E-01	4.72265012E-02
11	-5.97388888E-01	9.38891743E-02	-3.90882882E-01	4.94829200E-02

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
 SMOOTHED WING VELOCITY POTENTIALS
 (MACH 1.200 RED. FREQ. = .50000)
 MODE SHAPE 2

PAGE CONTINUED

ROW	REAL	CHORD 9	IMAGINARY	REAL	CHORD 10	IMAGINARY
12	-6.1441810E-01	9.58535425E-02	-4.01612967E-01	5.04614257E-02		
13	-6.29292640E-01	9.61020248E-02	-4.10160572E-01	5.01163937E-02		
14	-6.42352395E-01	9.44986311E-02	-4.11010800E-01	4.89269414E-02		
15	-6.53756026E-01	9.11986615E-02	-4.23138945E-01	4.59264449E-02		

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
SMOOTHED TAIL VELOCITY POTENTIALS
(MACH 1.200 RED. FREQ. = .50000)
MODE SHAPE 2

ROW	CHORD 1		CHORD 2		CHORD 3		CHORD 4	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
19	5.97433760E-02	2.31066510E-02	5.91216173E-02	2.30340104E-02	5.78806829E-02	2.26040507E-02	5.57754206E-02	2.27629510E-02
20	1.3719179E-01	1.70346754E-02	1.36236081E-01	1.76367243E-02	1.34671664E-01	1.96073137E-02	1.31929613E-01	2.32984408E-02
21	2.01305110E-01	-7.60417390E-03	2.01017365E-01	-5.85533367E-03	2.00069169E-01	-1.51733043E-03	1.97212464E-01	5.96041923E-03
22	2.37299536E-01	-4.73861693E-02	2.57146395E-01	-4.41446737E-02	2.56433384E-01	-3.71176455E-02	2.52884636E-01	-2.57274212E-02
23	3.06828959E-01	-9.68213134E-02	3.06545565E-01	-9.38770040E-02	3.04984542E-01	-8.38683105E-02	2.99613792E-01	-6.84432937E-02
24	3.51082130E-01	-1.58445810E-01	3.49737711E-01	-1.51735395E-01	3.46053979E-01	-1.38584073E-01	3.37625229E-01	-1.18090083E-01
25	3.89164169E-01	-2.22915939E-01	3.86217079E-01	-2.14519415E-01	3.79333215E-01	-1.98266359E-01	3.66852495E-01	-1.74817963E-01
26	4.19830679E-01	-2.69101622E-01	4.14816298E-01	-2.79230274E-01	4.04135036E-01	-2.60131082E-01	3.87088015E-01	-2.33047351E-01
27	4.41221663E-01	-3.54181322E-01	4.34085331E-01	-3.43154956E-01	4.19624370E-01	-3.21756255E-01	3.98133717E-01	-2.91491434E-01
28	4.51396648E-01	-4.13733910E-01	4.42640543E-01	-4.03950425E-01	4.25090365E-01	-3.80929597E-01	3.99951651E-01	-3.48179012E-01
29	4.49686781E-01	-4.71834549E-01	4.39553477E-01	-4.59727650E-01	4.20185475E-01	-4.35896146E-01	3.92814621E-01	-4.01477207E-01
30	4.35050969E-01	-5.21147517E-01	4.24710021E-01	-5.09135864E-01	4.05180532E-01	-4.85305868E-01	3.77456800E-01	-4.50114203E-01
31	4.08700880E-01	-5.63020373E-01	3.99181280E-01	-5.51446654E-01	3.81214832E-01	-5.28281265E-01	3.55224365E-01	-4.93201884E-01
32	3.73375767E-01	-5.97577737E-01	3.63992568E-01	-5.86638113E-01	3.50547409E-01	-5.64464988E-01	3.28226131E-01	-5.30259066E-01
33	3.33985581E-01	-6.25815209E-01	3.28492378E-01	-6.15476988E-01	3.16808117E-01	-5.94067443E-01	2.99484055E-01	-5.61233238E-01

ROW	CHORD 5		CHORD 6		CHORD 7		CHORD 8	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
19	5.27392407E-02	2.25273225E-02	4.85333313E-02	2.21201277E-02	4.29256515E-02	2.15236602E-02	3.55402098E-02	2.02375324E-02
20	1.26981689E-01	2.73878696E-02	1.190551130E-01	3.17193941E-02	1.07386336E-01	3.62411181E-02	9.05652650E-02	3.94935158E-02
21	1.90709403E-01	1.52077630E-02	1.79393232E-01	2.55213333E-02	1.61632194E-01	3.59942943E-02	1.35933077E-01	4.41803953E-02
22	2.44310135E-01	-1.10304228E-02	2.29147888E-01	5.65873036E-03	2.05466085E-01	2.23719269E-02	1.71718108E-01	3.58108518E-02
23	2.88048902E-01	-4.83032874E-02	2.69525844E-01	-2.5484251E-02	2.38994491E-01	-2.59391333E-03	1.96340599E-01	1.66888496E-02
24	3.22126451E-01	-9.39973708E-02	2.97771442E-01	-6.53737189E-02	2.62783306E-01	-3.65656042E-02	2.16498047E-01	-1.12045819E-02
25	3.46749677E-01	-1.45159921E-01	3.17407723E-01	-1.11427642E-01	2.77551627E-01	-7.70377523E-02	2.27096678E-01	-4.53452198E-02
26	3.62172030E-01	-1.99349659E-01	3.28153529E-01	-1.61117522E-01	2.84390547E-01	-1.21474974E-01	2.31182923E-01	-8.32997231E-02
27	3.68773935E-01	-2.54287549E-01	3.50894603E-01	-2.12070002E-01	2.84421943E-01	-1.67449677E-01	2.29874903E-01	-1.22759633E-01
28	3.67113213E-01	-3.07987563E-01	3.26654694E-01	-2.62168500E-01	2.78862268E-01	-2.12779839E-01	2.24593890E-01	-1.61675472E-01
29	3.57887495E-01	-3.58607447E-01	3.16566656E-01	-3.09654641E-01	2.68923444E-01	-2.55666791E-01	2.15495789E-01	-1.98390583E-01
30	3.4294641E-01	-4.05499490E-01	3.01843594E-01	-3.53230887E-01	2.55746150E-01	-2.94832997E-01	2.04402619E-01	-2.31373359E-01
31	3.2283156E-01	-4.47261290E-01	2.83749763E-01	-3.92160164E-01	2.40319615E-01	-3.29598833E-01	1.91733984E-01	-2.61361087E-01
32	2.98682610E-01	-4.83786521E-01	2.63572089E-01	-4.26369497E-01	2.23419408E-01	-3.60325373E-01	1.77938549E-01	-2.87474081E-01
33	2.74584054E-01	-5.15315699E-01	2.42590774E-01	-4.56550636E-01	2.05353172E-01	-3.87942162E-01	1.63125515E-01	-3.11369708E-01

19	2.6045000E-02	1.90552432E-02	1.70750952E-02	1.84678707E-02
20	6.80787276E-02	4.04667885E-02	3.94010565E-02	3.28520168E-02
21	1.02114556E-01	4.79949451E-02	5.84053396E-02	3.84332000E-02
22	1.28315740E-01	4.36974129E-02	7.37037680E-02	3.63091120E-02
23	1.47825765E-01	2.97647822E-02	8.51876260E-02	2.84585964E-02
24	1.60727357E-01	8.43868369E-03	9.29748121E-02	1.56893188E-02
25	1.68001398E-01	-1.80701021E-02	9.73609899E-02	-4.11863234E-04
26	1.70486964E-01	-4.76635347E-02	9.87707399E-02	-1.85296096E-02
27	1.69038385E-01	-7.84382042E-02	9.77087114E-02	-3.74706277E-02
28	1.64486310E-01	-1.08767221E-01	9.47107730E-02	-5.62142021E-02
29	1.57995767E-01	-1.37382106E-01	9.02931885E-02	-7.39827241E-02

SAMPLE CASE ---- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
 SMOOTHED TAIL VELOCITY POTENTIALS
 (MACH 1.200 REP. FREQ. = .50000)

PAGE CONTINUED

MODE SHAPE 2

ROW	CHORD 9		CHORD 10	
	REAL	IMAGINARY	REAL	IMAGINARY
30	1.49023425E-01	-1.63454602E-01	6.49136603E-02	-5.01922222E-02
31	1.39206632E-01	-1.06674961E-01	7.89026902E-02	-1.04702891E-01
32	1.28702970E-01	-2.07333039E-01	7.24345246E-02	-1.17669622E-01
33	1.17367190E-01	-2.26460801E-01	6.54684096E-02	-1.23692331E-01

ENTERING PROGRAM FORCES CURRENT ELAPSED TIME IS CP = 123.887, PP = 60.520

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

WING BOX LIFTS
(MACH 1.200 ACC.FREQ. = .50000)
MODE SHAPE 2

ROW	CHORD 1		CHORD 2		CHORD 3		CHORD 4	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
1	-3.97215631E-01	1.35929062E-02	-3.97164069E-01	1.36026822E-02	-3.97253729E-01	1.37324396E-02	-3.97765888E-01	1.41211589E-02
2	-3.90360234E-01	2.11846215E-02	-3.92081378E-01	2.42544355E-02	-3.94846411E-01	2.90484196E-02	-3.98289485E-01	3.29325302E-02
3	-3.82252977E-01	2.73287559E-02	-3.85583731E-01	3.31932352E-02	-3.91012651E-01	4.19421057E-02	-3.96123532E-01	4.79731000E-02
4	-3.73996770E-01	4.71420761E-02	-3.76597369E-01	5.07711631E-02	-3.80166406E-01	5.36841431E-02	-3.81626565E-01	5.76313589E-02
5	-3.63533687E-01	7.02821740E-02	-3.63844440E-01	6.91906240E-02	-3.62883753E-01	6.64105472E-02	-3.57828774E-01	6.10880666E-02
6	-3.49193047E-01	8.86567527E-02	-3.46460515E-01	8.23822268E-02	-3.39888974E-01	7.11910807E-02	-3.27494699E-01	5.79080607E-02
7	-3.29783766E-01	9.63960259E-02	-3.24041214E-01	8.59835319E-02	-3.12686371E-01	6.80203355E-02	-2.93139145E-01	4.80380167E-02
8	-3.04666675E-01	6.98246685E-02	-2.96699338E-01	7.73177985E-02	-2.80592080E-01	5.58088150E-02	-2.57043102E-01	3.18162087E-02
9	-2.74726874E-01	6.74343688E-02	-2.65121991E-01	5.53727320E-02	-2.46765890E-01	3.43740157E-02	-2.21263653E-01	9.98026936E-03
10	-2.40466015E-01	6.98359180E-02	-2.30626214E-01	2.07792311E-02	-2.12543033E-01	4.43150997E-03	-1.87679892E-01	-1.63250497E-02
11	-2.04074674E-01	-2.01684892E-02	-1.95215208E-01	-2.42098643E-02	-1.78966103E-01	-3.24139725E-02	-1.57948940E-01	-4.55403180E-02
12	-1.68304655E-01	-7.78130246E-02	-1.61634466E-01	-7.57410263E-02	-1.49216679E-01	-7.36774625E-02	-1.33581355E-01	-7.56826654E-02
13	-1.37741327E-01	-1.36195967E-01	-1.33427898E-01	-1.26381291E-01	-1.25647318E-01	-1.16002669E-01	-1.15928051E-01	-1.04340022E-01
14	-3.16879946E-01	-1.86407452E-01	-1.14993963E-01	-1.75139513E-01	-1.11313294E-01	-1.53170897E-01	-1.06201211E-01	-1.28661360E-01
15	-1.07933723E-01	-8.09354434E-01	-1.07326390E-01	-1.97213450E-01	-1.05712941E-01	-1.74534127E-01	-1.02928376E-01	-1.40506033E-01

ROW	CHORD 5		CHORD 6		CHORD 7		CHORD 8	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
1	-3.98876214E-01	1.49240795E-02	-4.00321197E-01	1.59026779E-02	-4.00919262E-01	1.63261614E-02	-3.97579662E-01	1.53034158E-02
2	-4.01339633E-01	3.51986354E-02	-4.01626647E-01	3.56575326E-02	-3.94714548E-01	3.26421818E-02	-3.73799187E-01	2.55540753E-02
3	-3.97824419E-01	4.91914791E-02	-3.91347569E-01	4.56756415E-02	-3.70198477E-01	3.62148631E-02	-3.27639754E-01	2.20321061E-02
4	-3.77241148E-01	5.42285899E-02	-3.62257916E-01	4.53579546E-02	-3.31054863E-01	3.13672036E-02	-2.78907724E-01	1.45671281E-02
5	-3.45317007E-01	5.15562486E-02	-3.21977014E-01	3.74992493E-02	-2.84643330E-01	2.09465936E-02	-2.31410071E-01	4.65717477E-03
6	-3.06964229E-01	4.23827223E-02	-2.76882413E-01	2.44348834E-02	-2.36985476E-01	7.23640692E-03	-1.88117491E-01	-6.52558877E-03
7	-2.66305193E-01	2.79055432E-02	-2.32102668E-01	8.08438473E-03	-1.92753378E-01	-7.99700767E-03	-1.51157465E-01	-1.81036155E-02
8	-2.28677329E-01	9.53878777E-03	-1.91510125E-01	-1.00049594E-02	-1.55266544E-01	-2.34574256E-02	-1.21807340E-01	-2.94747270E-02
9	-1.90637999E-01	-1.20596437E-02	-1.57713704E-01	-2.86145123E-02	-1.26446754E-01	-3.82718095E-02	-1.00487389E-01	-4.02628633E-02
10	-1.59869413E-01	-3.48607483E-02	-1.32051684E-01	-4.68108982E-02	-1.06871052E-01	-5.19433781E-02	-8.67538918E-02	-5.03023332E-02
11	-1.39663519E-01	-5.79371412E-02	-1.14584485E-01	-6.39014123E-02	-9.56986178E-02	-6.43046753E-02	-7.92921967E-02	-5.96310642E-02
12	-1.16028903E-01	-7.94357765E-02	-1.04087455E-01	-7.93924315E-02	-9.0600945E-02	-7.54706392E-02	-7.59097972E-02	-6.83983531E-02
13	-1.06485694E-01	-9.77506676E-02	-9.80436516E-02	-9.29438243E-02	-8.81905000E-02	-8.57916708E-02	-7.35294000E-02	-7.69006154E-02
14	-9.97904451E-02	-1.10995609E-01	-9.28366271E-02	-1.04326361E-01	-8.31181458E-02	-9.58067036E-02	-6.81819861E-02	-8.55198361E-02
15	-9.72982854E-02	-1.17206063E-01	-8.93510378E-02	-1.10466222E-01	-7.90100018E-02	-1.01711003E-01	-6.39986477E-02	-9.06459757E-02

1	-3.85894944E-01	1.19061519E-02	-3.15163317E-01	4.12032196E-03
2	-3.27339916E-01	1.45591002E-02	-2.06769075E-01	2.56366645E-03
3	-2.57949663E-01	6.61589194E-03	-1.52743544E-01	-3.33976552E-03
4	-2.04126452E-01	-7.93463293E-04	-1.15740509E-01	-7.46498569E-03
5	-1.63083742E-01	-7.70869395E-03	-9.12912121E-02	-1.07371504E-02
6	-1.32297626E-01	-1.44570547E-02	-7.55794936E-02	-1.37351619E-02
7	-1.09527690E-01	-2.12606552E-02	-6.54453264E-02	-1.69136809E-02
8	-9.28176767E-02	-2.82477879E-02	-5.83877825E-02	-2.05251600E-02
9	-8.04973521E-02	-3.54612575E-02	-5.25679600E-02	-2.46618767E-02
10	-7.11813710E-02	-4.28679082E-02	-4.68119595E-02	-2.92779666E-02
11	-6.37721413E-02	-5.03679330E-02	-4.06138234E-02	-3.42114571E-02

SAMPLE CASE ---- TWO ARE=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

WING BOX LIFTS
 (MACH 1.200 RED.FREQ. = .50000)
 MODE SHAPE 2

PAGE CONTINUED

ROW	CHORD 9		CHORD 10	
	REAL	IMAGINARY	REAL	IMAGINARY
12	-5.74596892E-02	-5.78045016E-02	-3.41384900E-02	-3.92062991E-02
13	-5.17227248E-02	-6.49718898E-02	-2.82247455E-02	-4.39344020E-02
14	-4.63295066E-02	-7.16270061E-02	-2.43881755E-02	-4.80176656E-02
15	-4.35031367E-02	-7.33247223E-02	-2.30696988E-02	-5.01516867E-02

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

TAIL BOX LIFTS

(MACH 1.200 RED.FREQ. = .50000)

MODE SHAPE 2

ROW	CHORD 1		CHORD 2		CHORD 3		CHORD 4	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
19	2.9433089E-01	6.4932624E-02	2.9221242E-01	6.7225601E-02	2.8799950E-01	6.9760564E-02	2.8069969E-01	7.5017702E-02
20	2.1201036E-01	-3.7585709E-02	2.1152216E-01	-2.9930027E-02	2.1239690E-01	-2.3253937E-02	2.1089452E-01	-1.2137608E-02
21	1.8107662E-01	-7.6967892E-02	1.8286177E-01	-7.3037225E-02	1.8371443E-01	-6.3509176E-02	1.8173051E-01	-5.4188021E-02
22	1.6367105E-01	-1.1176503E-01	1.6390421E-01	-1.0698151E-01	1.6107764E-01	-9.8505133E-02	1.5694895E-01	-8.6879357E-02
23	1.5126471E-01	-1.3673578E-01	1.4897436E-01	-1.3154455E-01	1.4349496E-01	-1.2246794E-01	1.3459558E-01	-1.1078809E-01
24	1.3984897E-01	-1.5197147E-01	1.3528286E-01	-1.4690150E-01	1.2594626E-01	-1.3785614E-01	1.1327516E-01	-1.1266033E-01
25	1.2933932E-01	-1.5805293E-01	1.1956481E-01	-1.5358405E-01	1.0738712E-01	-1.4533564E-01	9.2049752E-02	-1.3511182E-01
26	1.0735962E-01	-1.5590694E-01	1.0006716E-01	-1.5244356E-01	8.6735949E-02	-1.4575451E-01	7.0462950E-02	-1.3718307E-01
27	8.3307132E-02	-1.4678287E-01	7.8259108E-02	-1.4461416E-01	6.3766971E-02	-1.4011800E-01	4.8541804E-02	-1.3379413E-01
28	5.4304568E-02	-1.3220923E-01	4.8638529E-02	-1.3147587E-01	3.8938556E-02	-1.2895633E-01	2.6799062E-02	-1.2581620E-01
29	2.2299040E-02	-1.1395031E-01	1.8941489E-02	-1.1461766E-01	1.3974412E-02	-1.1535460E-01	6.2394368E-03	-1.1439162E-01
30	-8.6316567E-03	-9.3962718E-02	-9.9497905E-03	-9.5805990E-02	-1.0218867E-02	-9.8757733E-02	-1.1658026E-02	-1.0033462E-01
31	-3.6678627E-02	-7.4351994E-02	-3.3978247E-02	-7.6920909E-02	-2.9534405E-02	-8.1215380E-02	-2.4897873E-02	-8.3004907E-02
32	-5.2878913E-02	-5.7329200E-02	-4.7903718E-02	-5.9973119E-02	-4.0650277E-02	-6.4121644E-02	-3.1003768E-02	-6.9730354E-02
33	-5.6154208E-02	-5.1739621E-02	-5.0313363E-02	-5.4109272E-02	-4.2321048E-02	-5.7573950E-02	-3.0537204E-02	-6.3442236E-02

ROW	CHORD 5		CHORD 6		CHORD 7		CHORD 8	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
19	2.6858675E-01	8.0523907E-02	2.5043710E-01	8.6019466E-02	2.2445148E-01	9.1376245E-02	1.8808540E-01	9.3632115E-02
20	2.0525906E-01	1.6835170E-03	1.9402537E-01	1.7035023E-02	1.7536288E-01	3.2553935E-02	1.4739859E-01	4.5121584E-02
21	1.7535652E-01	-3.8846814E-02	1.6341894E-01	-2.1354005E-02	1.4426135E-01	-4.7433729E-03	1.1792447E-01	8.1922535E-03
22	1.4784785E-01	-7.1436102E-02	1.3388805E-01	-5.3979028E-02	1.1436070E-01	-3.7627303E-02	9.0491793E-02	-2.4273238E-02
23	1.2215385E-01	-9.6272490E-02	1.0600232E-01	-8.0232662E-02	8.6027648E-02	-6.4952222E-02	6.5839413E-02	-5.1193143E-02
24	9.7884486E-02	-1.1368341E-01	8.0229578E-02	-9.9787387E-02	6.1783630E-02	-8.5952679E-02	4.4471877E-02	-7.1870074E-02
25	7.4883958E-02	-1.2414958E-01	5.6945483E-02	-1.1259844E-01	4.0318620E-02	-1.0025032E-01	2.6672608E-02	-8.5977850E-02
26	5.3137764E-02	-1.2829888E-01	3.6443988E-02	-1.1890871E-01	2.2504887E-02	-1.0786205E-01	1.2518306E-02	-9.3588356E-02
27	3.2877869E-02	-1.2690040E-01	1.8947420E-02	-1.1925062E-01	8.4107735E-03	-1.0925068E-01	1.8903538E-03	-9.5168393E-02
28	1.4537586E-02	-1.2088535E-01	4.8166573E-03	-1.1449211E-01	-2.0852511E-03	-1.0510607E-01	-5.5097853E-03	-9.1588527E-02
29	-1.2331489E-03	-1.1120601E-01	-6.4387160E-04	-1.0562504E-01	-9.2821795E-03	-9.6607355E-02	-1.0148160E-02	-8.4129947E-02
30	-1.3582804E-02	-9.9097224E-02	-1.4150240E-02	-9.4197096E-02	-1.3639951E-02	-8.5974093E-02	-1.2644734E-02	-7.4491314E-02
31	-2.1333144E-02	-8.5812624E-02	-1.8480419E-02	-8.1885627E-02	-1.5768414E-02	-7.4701510E-02	-1.3759980E-02	-6.4795613E-02
32	-2.3274169E-02	-7.2729320E-02	-1.9412597E-02	-7.0715082E-02	-1.6412177E-02	-6.5521976E-02	-1.4381482E-02	-5.7597008E-02
33	-2.1128310E-02	-6.7607642E-02	-1.7605918E-02	-6.6740481E-02	-1.5139169E-02	-6.2714580E-02	-1.3526006E-02	-5.9735475E-02

19	1.39991339E-01	9.23373399E-02	8.32942147E-02	7.90731510E-02
20	1.10632483E-01	5.04360347E-02	5.90224862E-02	3.40387006E-02
21	8.63127384E-02	1.50818031E-02	4.78699816E-02	1.13538123E-02
22	6.43424992E-02	-1.46314300E-02	3.67215047E-02	-7.66663250E-03
23	4.35843523E-02	-3.83719377E-02	2.62063325E-02	-2.28682566E-02
24	2.93723444E-02	-5.60410657E-02	1.87831047E-02	-3.42263744E-02
25	1.63201716E-02	-6.77771167E-02	8.77887623E-03	-4.18508766E-02
26	8.32936778E-03	-7.39596418E-02	2.37717043E-03	-4.59911155E-02
27	-1.20250608E-03	-7.52134345E-02	-2.37356650E-03	-4.70407895E-02
28	-6.37367283E-03	-7.24126447E-02	-5.55491626E-03	-4.54282795E-02
29	-9.56995023E-03	-6.86848728E-02	-7.37343156E-03	-4.21942758E-02

SAMPLE CASE --- TWO AREA SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
 TAIL BOX LIFTS

(MACH 1.200 RED.FREQ. = .50000)

PAGE CONTINUED

MODE SHAPE 2 -----

ROW	CHORD 9		CHORD 10	
	REAL	IMAGINARY	REAL	IMAGINARY
30	-1.1236920E-02	-5.94152631E-02	-8.15558508E-03	-3.78511791E-02
31	-1.19699486E-02	-3.22505983E-02	-8.34271615E-03	-3.35334692E-02
32	-1.23095785E-02	-4.71033924E-02	-8.48397774E-03	-3.04298476E-02
33	-1.15314834E-02	-4.58798961E-02	-8.03437375E-03	-2.97035954E-02

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

WING PRESS. DIFFERENCE
(MACH 1.200 REF.FREQ. = .50000)
MODE SHAPE 2

ROW	CHORD 1		CHORD 2		CHORD 3		CHORD 4	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
1	-5.98825125E+00	2.04920801E-01	-5.98747362E+00	2.05071164E-01	-5.98882530E+00	2.07082431E-01	-5.9956146E+00	2.12084481E-01
2	-1.88821687E+00	3.19370186E-01	-5.91034918E+00	3.65649373E-01	-5.95233362E+00	4.37921402E-01	-6.00443990E+00	4.96476374E-01
3	-5.76586043E+00	4.11996497E-01	-5.81292361E+00	5.00406849E-01	-5.89473751E+00	6.32501034E-01	-5.97176693E+00	7.23291843E-01
4	-5.63821345E+00	7.10693537E-01	-5.67741893E+00	7.69404032E-01	-5.73122721E+00	8.39470043E-01	-5.75323694E+00	8.6823426E-01
5	-5.46077804E+00	1.05954364E+00	-5.48516131E+00	1.04308790E+00	-5.47067841E+00	1.00117667E+00	-5.39447174E+00	9.20337256E-01
6	-5.26426326E+00	1.33655129E+00	-5.22309029E+00	1.24195800E+00	-5.12401908E+00	1.07324592E+00	-4.93711835E+00	8.72996862E-01
7	-4.97167734E+00	1.43322477E+00	-4.68510511E+00	1.29622505E+00	-4.70487906E+00	1.02544514E+00	-4.41923889E+00	7.24200350E-01
8	-4.59603812E+00	1.55415783E+00	-4.47291082E+00	1.16560967E+00	-4.23008477E+00	8.41349542E-01	-3.87507056E+00	4.79841793E-01
9	-4.14166346E+00	1.01661133E+00	-3.99686440E+00	6.34775343E-01	-3.72013576E+00	5.18207785E-01	-3.33576553E+00	1.50436222E-01
10	-3.62316158E+00	4.50094899E-01	-3.47682099E+00	3.13238695E-01	-3.19968441E+00	6.68075263E-02	-2.82938083E+00	-2.46109389E-01
11	-3.07654146E+00	-3.04051415E-01	-2.94237999E+00	-3.64977437E-01	-2.69801555E+00	-4.88659022E-01	-2.38116835E+00	-6.86343110E-01
12	-2.54030326E+00	-1.17307548E+00	-2.43373126E+00	-1.14183893E+00	-2.24952607E+00	-1.11072954E+00	-2.01381470E+00	-1.14095911E+00
13	-2.07652835E+00	-2.05325145E+00	-2.01150125E+00	-1.93542079E+00	-1.89420458E+00	-1.74880603E+00	-1.74768113E+00	-1.57298502E+00
14	-1.76197116E+00	-2.61019807E+00	-1.73359922E+00	-2.64032750E+00	-1.67811105E+00	-2.33928929E+00	-1.60104350E+00	-1.93964298E+00
15	-1.62747361E+00	-3.15613685E+00	-1.61800620E+00	-2.97310463E+00	-1.59368255E+00	-2.63120097E+00	-1.55170365E+00	-2.11820814E+00

ROW	CHORD 5		CHORD 6		CHORD 7		CHORD 8	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
1	-6.01326517E+00	2.24988964E-01	-6.03506913E+00	2.33741890E-01	-6.04408528E+00	2.46126143E-01	-5.99373893E+00	2.30707674E-01
2	-6.05072714E+00	5.30639394E-01	-6.05474952E+00	5.37557330E-01	-5.9505411E+00	4.92059406E-01	-5.6323478E+00	3.65242100E-01
3	-5.99742878E+00	7.41589451E-01	-5.89978659E+00	6.88586205E-01	-5.58093202E+00	5.45959603E-01	-4.93935513E+00	3.32146497E-01
4	-5.68712429E+00	8.17526753E-01	-5.46124357E+00	6.83796893E-01	-4.99083984E+00	4.72878477E-01	-4.20469214E+00	2.19607720E-01
5	-5.20584975E+00	7.77239692E-01	-4.853598613E+00	5.63322453E-01	-4.29115966E+00	3.15781780E-01	-3.48863809E+00	7.02095513E-02
6	-4.62765987E+00	6.36943580E-01	-4.17415944E+00	3.68369728E-01	-3.57269048E+00	1.09092939E-01	-2.83597788E+00	-9.83769523E-02
7	-4.01470187E+00	4.20691892E-01	-3.49907938E+00	1.21876666E-01	-2.90587265E+00	-1.20559427E-01	-2.27878453E+00	-2.72952424E-01
8	-3.41728932E+00	1.40787525E-01	-2.88712377E+00	-1.50830438E-01	-2.34064076E+00	-3.53633997E-01	-1.83651474E+00	-4.44346229E-01
9	-2.87397397E+00	-1.81805969E-01	-2.37762355E+00	-4.3138004E-01	-1.90625654E+00	-5.76969237E-01	-1.51490439E+00	-6.06385503E-01
10	-2.41162964E+00	-5.27053112E-01	-1.99075403E+00	-7.03697827E-01	-1.61114173E+00	-7.83075889E-01	-1.30786413E+00	-7.58434197E-01
11	-2.04556602E+00	-8.73435268E-01	-1.72742610E+00	-9.63350037E-01	-1.44271095E+00	-9.69429456E-01	-1.19537484E+00	-6.98972118E-01
12	-1.77932254E+00	-1.19753539E+00	-1.56917745E+00	-1.19688594E+00	-1.36720462E+00	-1.13776270E+00	-1.14438325E+00	-1.03114397E+00
13	-1.60333224E+00	-1.47364676E+00	-1.47806366E+00	-1.40118087E+00	-1.32952181E+00	-1.29335610E+00	-1.10849741E+00	-1.15332040E+00
14	-1.90439736E+00	-1.47332177E+00	-1.39634970E+00	-1.57277907E+00	-1.25303316E+00	-1.44434040E+00	-1.02788227E+00	-1.28923702E+00
15	-1.48879669E+00	-1.76694788E+00	-1.34701788E+00	-1.66534095E+00	-1.19112059E+00	-1.53335226E+00	-9.64815917E-01	-1.36653947E+00

ROW CHORD 9 CHORD 10
REAL IMAGINARY REAL IMAGINARY

1	-3.7723547C+00	1.79491993C-01	-4.01159019E+00	6.21252350E-02
2	-4.93513642C+00	2.25516921E-01	-3.11717014E+00	3.86490333C-02
3	-3.60073742C+00	9.07303213E-02	-2.30269562C+00	-3.03460596E-02
4	-3.97732205C+00	-1.19619092E-02	-1.74405303E+00	-1.12640405E-01
5	-2.43037992C+00	-1.16215949E-01	-1.37626601E+00	-5.1668633E-01
6	-1.99446176C+00	-2.17940301E-01	-1.73940374E+00	-2.07065357E-01
7	-1.09119206E+00	-3.20316436E-01	-9.8625455E-01	-2.54903337E-01
8	-1.30020214E+00	-4.25051426E-01	-0.60228941E-01	-3.09420429E-01
9	-1.21354324C+00	-3.34590571E-01	-7.92491016E-01	-3.71791701E-01
10	-1.07309934E+00	-6.46250032E-01	-7.05716643E-01	-4.41381956E-01
11	-9.01401205E-01	-7.99325461E-01	-0.12276425E-01	-9.19757121E-01

SAMPLE CASE --- TWO AR=Z SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

WING PRESS. DIFFERENCE
 (MACH 1.200 RED.FREQ. = .50000)
 MODE SHAPE 2

PAGE CONTINUES

ROW	REAL	IMAGINARY	CHORD 9	REAL	IMAGINARY	CHORD 10
12	-6.6623740E-01	-8.7143263E-01		-3.1463710E-01	-5.9105719E-01	
13	-7.7974941E-01	-9.7946803E-01		-4.2550404E-01	-6.6233603E-01	
14	-8.9844359E-01	-1.0796177E+00		-5.6766557E-01	-7.2369354E-01	
15	-8.5583446E-01	-1.1385828E+00		-5.4778876E-01	-7.5611032E-01	

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

TAIL PRESS. DIFFERENCE

(MACH 1.200 RED.FREQ. = .50000)

MODE SHAPE 2

ROW	CHORD 1		CHORD 2		CHORD 3		CHORD 4	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
19	4.44022021E+00	1.00242395E+00	4.40326807E+00	1.01346407E+00	4.34175589E+00	1.05198159E+00	4.23170705E+00	1.13093442E+00
20	3.19617650E+00	-4.912448051E-01	3.19831500E+00	-1.51212140E-01	3.2019702E+00	-3.50366593E-01	3.17935455E+00	-1.82988348E-01
21	2.74192038E+00	-1.16033472E+00	2.75674498E+00	-1.10107766E+00	2.76959595E+00	-9.87587996E-01	2.73999212E+00	-8.16915159E-01
22	2.46743394E+00	-1.68522316E+00	2.46491879E+00	-1.61240696E+00	2.44039725E+00	-1.48502076E+00	2.36608845E+00	-1.30975359E+00
23	2.28040135E+00	-2.06136948E+00	2.24587300E+00	-1.98310885E+00	2.16326792E+00	-1.84627367E+00	2.02910474E+00	-1.67019344E+00
24	2.10830263E+00	-2.29105617E+00	2.03946594E+00	-2.21462355E+00	1.89871138E+00	-2.07825964E+00	1.70768737E+00	-1.97086168E+00
25	1.89652545E+00	-2.56273760E+00	1.80250444E+00	-2.31536672E+00	1.61892182E+00	-2.19101722E+00	1.38770222E+00	-2.03688740E+00
26	1.61851232E+00	-2.35038561E+00	1.50887073E+00	-2.29817317E+00	1.30785965E+00	-2.19733205E+00	1.06226895E+00	-2.06811267E+00
27	1.25390227E+00	-2.21263302E+00	1.14964932E+00	-2.18014061E+00	9.61323271E-01	-2.11235840E+00	7.31795233E-01	-2.01641951E+00
28	8.18672359E-01	-1.99312921E+00	7.33233774E-01	-1.98207341E+00	5.87020620E-01	-1.95324060E+00	4.04011068E-01	-1.89672684E+00
29	3.35279169E-01	-1.71766560E+00	2.85533642E-01	-1.72792635E+00	2.04641872E-01	-1.73873454E+00	9.40030483E-02	-1.72451856E+00
30	-1.45198674E-01	-1.41654128E+00	-1.49998736E-01	-1.44424837E+00	-1.54070293E-01	-1.48882916E+00	-1.73751366E-01	-1.51561848E+00
31	-3.52927975E-01	-1.12088805E+00	-5.12241361E-01	-1.13962634E+00	-4.45247917E-01	-1.22436792E+00	-3.75349571E-01	-1.28595328E+00
32	-7.97148465E-01	-8.64270209E-01	-7.22175725E-01	-9.04128795E-01	-6.12823989E-01	-9.66670311E-01	-4.67429541E-01	-1.05134540E+00
33	-8.46556544E-01	-7.80004135E-01	-7.58502795E-01	-8.15727969E-01	-6.38013806E-01	-8.67959956E-01	-4.60365673E-01	-9.56427701E-01

ROW	CHORD 5		CHORD 6		CHORD 7		CHORD 8	
	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
19	4.05050517E+00	1.21394358E+00	3.77548133E+00	1.29879224E+00	3.38373340E+00	1.37754873E+00	2.83549414E+00	1.41155725E+00
20	3.09439679E+00	2.50764628E-02	2.92504260E+00	2.56782492E-01	2.64369497E+00	4.90769040E-01	2.22211736E+00	6.80233476E-01
21	2.64362925E+00	-5.856397763E-01	2.46363327E+00	-3.21923745E-01	2.17482177E+00	-7.15090375E-02	1.77777836E+00	1.23502809E-01
22	2.22889027E+00	-1.07693976E+00	2.01843759E+00	-8.13764488E-01	1.72405253E+00	-5.67232940E-01	1.36421512E+00	-3.63932836E-01
23	1.84153661E+00	-1.45135240E+00	1.59804519E+00	-1.20955591E+00	1.30596094E+00	-9.79191592E-01	9.92366498E-01	-7.71795834E-01
24	1.47581491E+00	-1.71364202E+00	1.20950540E+00	-1.50435145E+00	9.31423270E-01	-1.29578539E+00	6.70436782E-01	-1.08348213E+00
25	1.12891814E+00	-1.87162538E+00	8.58485470E-01	-1.69748653E+00	6.07826076E-01	-1.51133359E+00	4.02107721E-01	-1.29616486E+00
26	8.01082236E-01	-1.93417850E+00	5.49413797E-01	-1.79261630E+00	3.59273944E-01	-1.62608159E+00	1.88720577E-01	-1.41089756E+00
27	4.95652531E-01	-1.91309564E+00	2.85643317E-01	-1.7977077E+00	1.26797179E-01	-1.64632835E+00	2.84981562E-02	-1.3471751E+00
28	2.19182361E-01	-1.82200023E+00	6.95987278E-02	-1.72358677E+00	-3.14404740E-02	-1.58453372E+00	-8.30651390E-02	-1.38074900E+00
29	-1.85904345E-02	-1.67649371E+00	-9.70672961E-02	-1.59235751E+00	-1.359534122E-01	-1.45942579E+00	-1.52989275E-01	-1.26830668E+00
30	-2.0468864E-01	-1.49398453E+00	-2.13322894E-01	-1.42007469E+00	-2.03630007E-01	-1.29610823E-00	-1.90626541E-01	-1.12299882E+00
31	-5.21910760E-01	-1.29367701E+00	-2.78602803E-01	-1.23447228E+00	-2.37717800E-01	-1.12616764E+00	-2.07439512E-01	-9.76830634E-01
32	-3.50871239E-01	-1.09644026E+00	-2.92655348E-01	-1.06606999E+00	-2.47422880E-01	-9.87780957E-01	-2.16808001E-01	-8.68307579E-01
33	-3.18321258E-01	-1.01922356E+00	-2.65419206E-01	-1.00615062E+00	-2.28231566E-01	-9.45457877E-01	-2.03912223E-01	-8.40243912E-01

ROW	CHORD 9		CHORD 10	
	REAL	IMAGINARY	REAL	IMAGINARY

19	2.11044665E+00	1.39203866E+00	1.23370733E+00	1.19210276E+00
20	1.66764743E+00	7.60352133E-01	6.89797459E-01	5.13152719E-01
21	1.30121349E+00	2.27366766E-01	7.21667127E-01	1.71163161E-01
22	9.73014786E-01	-2.20377107E-01	5.53648770E-01	-1.15578834E-01
23	6.47209967E-01	-5.76479026E-01	3.95075326E-01	-3.44751940E-01
24	4.43819867E-01	-6.44850854E-01	2.33014823E-01	-5.15982008E-01
25	2.49050938E-01	-1.02177846E+00	1.32346539E-01	-6.30925704E-01
26	9.54168034E-02	-1.11498355E+00	3.58371987E-02	-6.93342154E-01
27	-1.81284612E-02	-1.13388519E+00	-3.57828613E-02	-7.09166565E-01
28	-9.60867332E-02	-1.09166169E+00	-8.37435136E-02	-6.86983964E-01
29	-1.44272428E-01	-1.00531228E+00	-1.11158663E-01	-6.36102842E-01

SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

TAIL PRESS. DIFFERENCE
(MACH 1.200 RED.FREQ. = .50000)

MODE SHAPE 2

PAGE CONTINUED

ROW	CHORD 9	CHORD 10
	REAL IMAGINARY	REAL IMAGINARY
30	-1.68698057E-01 -0.95718793E-01	-1.22950071E-01 -5.70627996E-01
31	-1.80433764E-01 -7.87707407E-01	-1.25771178E-01 -5.05536069E-01
32	-1.69373670E-01 -7.10110365E-01	-1.27930928E-01 -4.58747213E-01
33	-1.73843693E-01 -6.81009194E-01	-1.21122742E-01 -4.47798549E-01

SECTION LIFTS

(MACH 1.200 RED. FREQ. = .50000)

MODE SHAPE 2
WING

CHORD

	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
1	-4.14178604E+00 -7.62410686E-02	-4.09082028E+00 -8.78372846E-02	-3.98839565E+00 -1.11155716E-01	-3.83484957E+00 -1.49384478E-01				
5	-3.62843363E+00 -2.29619564E-01	-3.36649421E+00 -3.23848097E-01	-3.03659545E+00 -4.00020976E-01	-2.61837201E+00 -4.43659370E-01				
9	-2.08432383E+00 -4.37411750E-01	-1.33493772E+00 -3.35515650E-01						

TOTAL LIFT - WING

-3.21255904E+01 -2.59287395E+00

SECTION LIFTS
TAIL

CHORD

	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
1	1.38101091E+00 -1.42783647E+00	1.35633365E+00 -1.40470845E+00	1.30312439E+00 -1.35558662E+00	1.22415418E+00 -1.28064991E+00				
5	1.11208616E+00 -1.17471445E+00	9.68867023E-01 -1.03667112E+00	8.05754573E-01 -8.77489637E-01	6.25322874E-01 -7.03464991E-01				
9	4.33271994E-01 -5.11882095E-01	2.32736505E-01 -2.94431576E-01						

TOTAL LIFT - TAIL

9.44488166E+00 -1.00676363E+01

TOTAL LIFT

-2.26009266E+01 -1.26603102E+01

GENERALIZED FORCES
 (MACH 1.200 RED. FREQ. = .50000)

WT.
 FUNCT VELOCITY POTENTIAL MODE 1 VELOCITY POTENTIAL MODE 2
 REAL IMAGINARY REAL IMAGINARY
 1 -3.65294231E+00 -1.49246649E+01 -3.21255904E+01 -2.59267395E+00
 2 -1.31902372E+00 -4.98711150E+00 -1.14917114E+01 -3.45896395E+00

GENERALIZED AERODYNAMIC COEFFICIENTS
 (MACH 1.200 RED.FREQ. = .50000)
 REAL PART

WT. VELOCITY POTENTIAL MODES
 FUNCT

	1	2
1	3.65294231E-01	3.21255904E+00
2	1.31502372E-01	1.14917114E+00

GENERALIZED AERODYNAMIC COEFFICIENTS
 (MACH 1.200 RED.FREQ. 3 .50000)
 IMAGINARY PART

PAGE CONTINUED

WT.
 FUNCY

VELOCITY POTENTIAL MODES

	1	2
1	2.9849697E+00	5.16574791E-01
2	9.97422299E-01	6.91792789E-01

ENTERING PROGRAM FORCES CURRENT ELAPSED TIME IS CP = 125.982, FP = 66.297

PROGRAM FORCES IS BEING RECALLED TO COMPUTE AIR FORCES WITHOUT SHOOTING.

APPENDIX B
PROGRAM LISTINGS

	OVERLAY (AFMBOX,0,0)	DRIVER	00002
	PROGRAM DRIVER (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1=1000,	DRIVER	00003
1	TAPE2=1000,TAPE3=1000,MODESC =110,IVPSC=110,IGEOSC=110,	DRIVER	00004
2	IWTFSC=110,IAICSC=110)	DRIVER	00005
C		DRIVER	00006
C	THIS IS A DUMMY (0,0) OVERLAY DRIVING PROGRAM	DRIVER	00007
C		DRIVER	00008
	COMMON PKERNL (1640)	DRIVER	00009
	COMPLEX PKERNL	DRIVER	00010
C		DRIVER	00011
	COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUTP,	FILES	00002
1	IQUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC	FILES	00003
	COMMON /ARRAYS/ KBXCDW,LBXCDW,LBOXC,KBXCDT,LBXCDT,KJALPH,LJALPH,	ARRAYS	00002
1	KALPHA,KKERNL,LKERNL,KPNTRM,LPNTRM,KDEFSL,KELPHI,	ARRAYS	00003
2	LMODES,KPNTSD,LPNTSD,KSDW,LSDW,KPNTDW,LPNTDW,	ARRAYS	00004
3	KDW,LDW,KTVP,LTVP	ARRAYS	00005
	DATA ITPEI,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC/	DRIVER	00014
1	5LTAPE1,6LMODESC,5LIVPSC,6LIGEOSC,6LIWTFSC,6LIAICSC /	DRIVER	00015
	NT5 = 5	DRIVER	00016
	NT6 = 6	DRIVER	00017
	READ(5,5) LINK,L1,L2	DRIVER	00018
5	FORMAT(A6,4X,2I10)	DRIVER	00019
	WRITE(6,6) LINK,L1,L2	DRIVER	00020
6	FORMAT(*1 PROGRAM BEGINS *,A6,2I5)	DRIVER	00021
	CALL OVERLAY (LINK,L1,L2,0)	DRIVER	00022
	WRITE(6,7)	DRIVER	00023
7	FORMAT(*0 PROGRAM TERMINATES*)	DRIVER	00024
	CALL EXIT	DRIVER	00025
	END	DRIVER	00026

	OVERLAY (AFMBOX,1,0)	CONTROL 00002
	PROGRAM CONTROL	CONTROL 00003
C		CONTROL 00004
C	SUPERSONIC UNSTEADY AERODYNAMICS,	CONTROL 00005
C	WING / HORIZONTAL TAIL, VERTICAL SEPARATION, DIHEDRAL	CONTROL 00006
C		CONTROL 00007
C	THIS PRIMARY OVERLAY CONTROLS THE CALLING OF THE COMPUTATIONAL	CONTROL 00008
C	SECTIONS OF THE PROGRAM	CONTROL 00009
C		CONTROL 00010
C	THIS IS THE TOTAL COMMON FOR ALL THE OVERLAY STRUCTURE	CONTROL 00014
	COMMON /CONTRL/ PREVEX,OMACH, TITLE(8), PRVGEOM,PRVMODE,DIHW,DIHT,	CONTRL 00002
1	DEFAULT	CONTRL 00003
	LOGICAL PRVGEOM,PRVMODE,DIHW,DIHT,DEFAULT	CONTRL 00004
	COMMON /PROBLM/ XMACH,NMODES,NTSLOP,NKVALS,SMOOTH,NDEG,CRDFIT,	PROBLM 00002
1	EXAIC,SUBDV,PLYWOOD	PROBLM 00003
	LOGICAL SMOOTH,CRDFIT,EXAIC,SUBDV,PLYWOOD	PROBLM 00004
	COMMON /GEOMTY/ COPLAN,NSUBDV,XSUBDV,NSUBD2,NSUBCN,NSURF,	GEOMTY 00002
1	B1,B1BETA,B13,B1BTAS,WLAX,WLAZ,PSIW,	GEOMTY 00003
2	MXBW,MXBBW,MYBW,MYBBW,MXBSW,MYBSW,MYBBSW,	GEOMTY 00004
3	IXBW,XCENTR	GEOMTY 00005
	LOGICAL COPLAN	GEOMTY 00006
	COMMON /GEOM2/ TLAX,TLAZ,PSIT,MXBT,MYBT,MYBBT,MXBST,MYBST,	GEOM2 00002
1	MYBBST,IXBT,IXBST,CAPL	GEOM2 00003
	COMMON /KERN/ ERR,MXSKRN,IPKERN,NFLKRN,NSFATK,NRONEA	KERN 00002
	COMMON /KVAL/ IKVAL,XKVAL(20),XKS(20)	KVAL 00002
	COMMON /FILES/ NT5,NT6,INTAPE,INFSP,NPLAIC,NSFAIC,NOUTP,	FILES 00002
1	IQUFSP,MODESC,IVPSC,IGEOSC,IWYFSC,IAICSC	FILES 00003
	COMMON /IOCONT/ OPLAIC,OSFAIC,WTGEOM,WTGNAF,WTSI,WTEL,PRBOX,	IOCONT 00002
1	PRPAIC,PRSAIC,PRMODS,PRCOEF,PRDW,PRSW,PRVP,	IOCONT 00003
2	PRBL,PRDCP,PRGNAC,PRGNAC,PRSL,PRLW,PRNW,PRCM	BCSFRB 00001
	EQUIVALENCE (PRUM,PRDW)	IOCONT 00005
	LOGICAL OPLAIC,OSFAIC,WTGEOM,WTGNAF,WTSI,WTEL,PRBOX,PRPAIC,	IOCONT 00006
1	PRSAIC,PRMODS,PRCOEF,PRDW,PRSW,PRVP,PRBL,PRSL,PRGNAF,	IOCONT 00007
2	PRDCP,PRGNAC,PRUM,PRLW,PRNW,PRCM	BCSFRB 00002
	COMMON /TAPEIO/ NFS,NMS,LS,NMR,ID(20),NID,ITYPE,LRS,LWS,M,N,	TAPEIO 00002
1	PARM(10),IRR	TAPEIO 00003
	DIMENSION IPARM(10)	TAPEIO 00004
	EQUIVALENCE (PARM,IPARM)	TAPEIO 00005
	COMMON /MODES/ SYM,SYMT,MTYFEW,MTYPET	MODECM 00002
	COMMON /ARRAYS/ KBXCOW,LBXCOW,LBXCX,KBXCDT,LBXCDT,KJALPH,LJALPH,	ARRAYS 00002
1	KALPHA,KKERNL,LKERNL,KPNTRM,LPNTRM,KDEFSL,KELPHI,	ARRAYS 00003
2	LNODES,KPNTSD,LPNTSD,KSCW,LSCW,KPNTDW,LPNTDW,	ARRAYS 00004
3	KDW,LDW,KTVP,LTVP	ARRAYS 00005
	COMMON /SAMPLW/ ISMPLW,ICHORD(10),IBOXF(10),IBOXL(10),ZLOC(10)	SAMPLW 00002
	COMMON /PLANXY/ NMLE,NWTE,NTLE,NTTE, XMLE(10),YMLE(10),	PLANXY 00002
1	XWTE(10),YWTE(10), XTLE(10),YTLE(10),	PLANXY 00003
2	XTTE(10),YTTE(10)	PLANXY 00004
	COMMON /CHECKPR/ DPPCPR,GEOPR,MODCPR,AICCPR,NMSCPR,SMCPR,GAFCPR	CHECKPR 00002
	LOGICAL DPPCPR, GEOPR, MODCPR, AICCPR,NMSCPR, SMCPR, GAFCPR	CHECKPR 00003
	COMMON /RWBUFF/ BFCODE,IBFCNT, BUFF(3280)	RWBUFF 00002
	DATA BFCODE,IBFCNT / 8HBUFSIZE,3280 /	BCSCNA 00001
	DATA TEVI47 /8HAFMBOX /	FTNX1 00002
	DATA PREVEX /10HNEVER EXEC /	FTNX1 00003
	DATA EXEC /10HAFMBOX EXC /	FTNX1 00004
	CALL RDINIT	CONTROL 00030
1	CONTINUE	CONTROL 00031
	PROG = GHATAPP	CONTROL 00032

CALL DTIME(CPTIME,PPTIME)	CONTROL 00033
WRITE (NT6,6005) PROG,CPTIME,PPTIME	CONTROL 00034
6005 FORMAT(1H0,10X,ENTERING PROGRAM *,A6,* CURRENT ELAPSED TIME IS *	CONTROL 00035
1 * CP =*,F8.3,*, PP =*,F8.3)	CONTROL 00036
CALL OVERLAY(TEV147,1,1,0)	CONTROL 00037
PARM(3) = XMACH	CONTROL 00038
IPARM(5) = NKVALS	CONTROL 00039
IF (OMACH .EQ. XMACH .AND. PRVGEOM) GO TO 100	CONTROL 00040
C	CONTROL 00041
PROG = GHGEOM	CONTROL 00042
CALL DTIME(CPTIME,PPTIME)	CONTROL 00043
WRITE (NT6,6005) PROG,CPTIME,PPTIME	CONTROL 00044
C COMPUTE GEOMETRY SECTION	CONTROL 00045
CALL OVERLAY(TEV147,1,2,0)	CONTROL 00046
C	CONTROL 00047
C READ MODE SHAPES, PLACE IN INTERNAL STORAGE CONVENTION,	CONTROL 00048
C STORE ON SCRATCH FILE. COMPUTE AND STORE OPTIONAL	CONTROL 00049
C THICKNESS SLOPE FUNCTIONS	CONTROL 00050
C	CONTROL 00051
100 CONTINUE	CONTROL 00052
IF (NKVALS .LE. 0) GO TO 810	CONTROL 00053
PROG = GHMODES	CONTROL 00054
CALL DTIME(CPTIME,PPTIME)	CONTROL 00055
WRITE (NT6,6005) PROG,CPTIME,PPTIME	CONTROL 00056
CALL OVERLAY(TEV147,1,3,0)	CONTROL 00057
C	CONTROL 00058
C SPACE OUTPUT TAPE IF DESIRED	CONTROL 00059
IF (NOUTP .LE. 0) GO TO 200	CONTROL 00060
IF (PREVEX .NE. EXEC) REWIND NOUTP	CONTROL 00061
C FILE SPACING A FUNCTION OF INSTALLATION CAPABILITIES	CONTROL 00062
200 CONTINUE	CONTROL 00063
C	CONTROL 00064
C LOOP ON NUMBER OF K1 VALUES THRU KERNELS, DOWNWASHES AND	CONTROL 00065
C AIR FORCES	CONTROL 00066
DO 800 IKVAL = 1,NKVALS	CONTROL 00067
C	CONTROL 00068
C CALL KERNEL ROUTINES	CONTROL 00069
PROG = GHKIC	CONTROL 00070
CALL DTIME(CPTIME,PPTIME)	CONTROL 00071
WRITE (NT6,6005) PROG,CPTIME,PPTIME	CONTROL 00072
CALL OVERLAY(TEV147,1,4,0)	CONTROL 00073
C	CONTROL 00074
PROG = GHVELPOT	CONTROL 00075
CALL DTIME(CPTIME,PPTIME)	CONTROL 00076
WRITE (NT6,6005) PROG,CPTIME,PPTIME	CONTROL 00077
C CALL DOWNWASH AND VELOCITY POTENTIAL ROUTINES.	CONTROL 00078
CALL OVERLAY(TEV147,1,5,0)	CONTROL 00079
C	CONTROL 00080
IF(.NOT.SMOOTH) GO TO 600	CONTROL 00081
IF(CRDFIT) GO TO 600	CONTROL 00082
C	CONTROL 00083
PROG = GHSMOOTH	CONTROL 00084
CALL DTIME(CPTIME,PPTIME)	CONTROL 00085
WRITE (NT6,6005) PROG,CPTIME,PPTIME	CONTROL 00086
CALL OVERLAY(TEV147,1,6,0)	CONTROL 00087
GO TO 700	CONTROL 00088
C	CONTROL 00089

600 CONTINUE	CONTROL 00090
IF(.NOT.CRDFIT) GO TO 700	CONTROL 00091
C	CONTROL 00092
PROG = GHCHORDF	CONTROL 00093
CALL DTIME(CPTIME,PPTIME)	CONTROL 00094
WRITE (NT6,6005) PROG,CPTIME,PPTIME	CONTROL 00095
CALL OVERLAY (TEV147,1,7,0)	CONTROL 00096
C	CONTROL 00097
700 CONTINUE	CONTROL 00098
PROG = GHFORCES	CONTROL 00099
CALL DTIME(CPTIME,PPTIME)	CONTROL 00100
WRITE (NT6,6005) PROG,CPTIME,PPTIME	CONTROL 00101
CALL OVERLAY (TEV147,1,8,0)	CONTROL 00102
IF(.NOT.(SMOOTH.OR.CRDFIT)) GO TO 800	CONTROL 00103
NIVPSC = IAICSC	CONTROL 00104
IAICSC = IVPSC	CONTROL 00105
IVPSC = NIVPSC	CONTROL 00106
CALL DTIME(CPTIME,PPTIME)	CONTROL 00107
WRITE (NT6,6005) PROG,CPTIME,PPTIME	CONTROL 00108
WRITE (NT6,6010)	CONTROL 00109
8010 FORMAT(1HD,5X,90(1H*),///6X,*PROGRAM FORCES IS BEING RECALLED TO C	CONTROL 00110
10MPUTE AIR FORCES WITHOUT SMOOTHING,*/6X,90(1H*))	CONTROL 00111
CALL OVERLAY (TEV147,1,8,GHRECALL)	CONTROL 00112
C	CONTROL 00113
800 CONTINUE	CONTROL 00114
C	CONTROL 00115
END OF LOOP ON REDUCED FREQUENCIES	CONTROL 00116
C	CONTROL 00117
810 CONTINUE	CONTROL 00118
IF (NOUTP .GT. 0) REWIND NOUTP	CONTROL 00119
IF (NPLAIC .GT. 0) REWIND NPLAIC	CONTROL 00120
IF (NSPAIC .GT. 0) REWIND NSPAIC	CONTROL 00121
CALL DTIME(CPTIME,PPTIME)	CONTROL 00122
WRITE (NT6,6006) CPTIME,PPTIME	CONTROL 00123
8006 FORMAT(1HD,10X*PROGRAM COMPLETED *,6X,* CURRENT ELAPSED TIME IS *	CONTROL 00124
1 * CP =*,F8.3,*, PP =*,F8.3)	CONTROL 00125
READ(5,8005) LINK,L1,L2	CONTROL 00126
8005 FORMAT(A6,4X,2I10)	CONTROL 00127
C	CONTROL 00128
DETERMINE IF ANOTHER CYCLE IS WANTED.	CONTROL 00129
C	CONTROL 00130
IF L1 = -1, RECYCLE	CONTROL 00131
IF(L1.EQ.-1) GO TO 1	CONTROL 00132
C	CONTROL 00133
IF L1 = -2, RETURN TO CALLING PROGRAM	CONTROL 00134
IF(L1.EQ.-2) RETURN	CONTROL 00135
C	CONTROL 00136
IF L1 = 0, CALL EXIT	CONTROL 00137
IF(L1.EQ.0) CALL EXIT	
C	
IF L1 = POS. CALL OVERLAY	
IF(L1.GT.0) CALL OVERLAY(LINK,L1,L2,0)	
END	

C SUBROUTINE FLUSH(I) ROUTINE TO FORCE AN ERROR EXIT DIMENSION MESSAGE(4) DATA MESSAGE /10H PROGRAM F,10H LUSHED VIA,10H MODE 1 , 0 / DATA NT5 /6L OUTPUT/ WRITE (NT5,8000) (MESSAGE(I),I=1,3) ENDFILE NT5 CALL REMARK(MESSAGE) CALL FLSHXXX 8000 FORMAT(5H0*** , 3A10, 4H ***) END	FLUSH 00002 FLUSH 00003 FLUSH 00004 FLUSH 00005 FLUSH 00006 FLUSH 00007 FLUSH 00008 FLUSH 00009 FLUSH 00010 FLUSH 00011 FLUSH 00012

```

SUBROUTINE RDINIT
COMMON /TAPEIO/ NFS,NMS,LS,NMR,ID(20),NID,ITYPE,LRS,LWS,M,N,
1          PARM(10), IRR
DIMENSION IA(1)
EQUIVALENCE (IA,NFS)
DO 10 I= 1,41
IA(I) = 0
10 CONTINUE
NID = 20
RETURN
END

```

```

RDINIT 00002
RDINIT 00003
RDINIT 00004
RDINIT 00005
RDINIT 00006
RDINIT 00007
RDINIT 00008
RDINIT 00009
RDINIT 00010
RDINIT 00011
RDINIT 00012

```

C SUBROUTINE DTIME(CPTIME,PPTIME)
 ROUTINE TO INTERROGATE THE SYSTEM CLOCKS
 PPTIME = 0
 CALL SECOND(CPTIME)
 RETURN
 END

DTIME 00002
DTIME 00003
DTIME 00004
DTIME 00005
DTIME 00006
DTIME 00007

SUBROUTINE READMX(INFILE,MXREAD,RANDIN, NFS, NMS, LS, NMR, K, NID,	READMX 00002
1 ID, ITYPE, LRS, A, M, N, PARM, IRR)	READMX 00003
C	READMX 00004
C ROUTINE TO READ A MATRIX ON TAPE OR DISK FILE.	READMX 00005
C THIS VERSION WILL WORK WITH SEQUENTIAL FILES ONLY.	READMX 00006
C SOME VARIABLES ARE PASSED FOR RANDOM OPERATION BUT	READMX 00007
C ARE NOT CURRENTLY USED.	READMX 00008
C	READMX 00009
C INPUT -	READMX 00010
C INFILE - TAPE NUMBER OR LEFT ADJUSTED FILE NAME	READMX 00011
C MXREAD - .T. SMART FORMAT (NOT USED)	READMX 00012
C .F. TEL001 FORMAT	READMX 00013
C RANDIN - .T. RANDOM FILE (NOT USED)	READMX 00014
C .F. SEQUENTIAL FILE	READMX 00015
C NFS - NUMBER OF FILES TO SPACE	READMX 00016
C NMS - NUMBER OF MATRICES TO SPACE	READMX 00017
C LS - LEVEL NUMBER TO SPACE (NOT USED)	READMX 00018
C NMR - IDENTIFIER (NAME OR NUMBER) (NOT USED)	READMX 00019
C K - ROW DIMENSION OF ARRAY A	READMX 00020
C (IF K=0, MATRIX WILL BE LEFT IN /RWBUFF/. IT WILL	READMX 00021
C BE STORED AS A ROW-WISE MATRIX, NOT AS A FORTRAN	READMX 00022
C COLUMN-WISE MATRIX. M-ROWS AND N-COLUMNS)	READMX 00023
C NID - NUMBER OF WORDS AVAILABLE IN ID ARRAY	READMX 00024
C IN/OUT	READMX 00025
C ID - IDENTIFICATION ARRAY	READMX 00026
C ITYPE - REAL,DIAGONAL,NULL,MIXED,COMPLEX	READMX 00027
C OUTPUT -	READMX 00028
C LRS - LEVEL NUMBER OF MATRIX READ (NOT USED)	READMX 00029
C A - ARRAY CONTAINING MATRIX	READMX 00030
C M - ROW DIMENSION OF MATRIX	READMX 00031
C N - COLUMN DIMENSION OF MATRIX	READMX 00032
C PARM - ARRAY OF NUMERICAL PARAMETERS STORED WITH THE MATRIX	READMX 00033
C IRR -	READMX 00034
C 0, NO ERROR	READMX 00035
C 1, MATRIX SPACING IS NEGATIVE	READMX 00036
C 2, FILE SPACING IS NEGATIVE	READMX 00037
C 4, MATRIX DIMENSIONS ILLEGAL	READMX 00038
C 5, M .GT. K	READMX 00039
C 1500 + I, ENCOUNTERED AFTER MATRIX I WHILE	READMX 00040
C SKIPPING MATRICES.	READMX 00041
C	READMX 00042
C DIMENSION ID(1), A(K,1), PARM(10), B(16)	READMX 00043
C	READMX 00044
C COMMON /RWBUFF/ BFCODE,IBFCNT, BUFF(3280)	READMX 00045
C	READMX 00047
C DIMENSION IBUFF(2500), I?PARM(10),IB(16)	READMX 00048
C EQUIVALENCE (BUFF,IBUFF),(I,IB)	READMX 00049
C	READMX 00050
C LOGICAL MXREAD,RANDIN	READMX 00051
C IRR = 0	READMX 00052
C	READMX 00053
C DO FILE SPACING	READMX 00054
C	READMX 00055
C IF(NFS) 215,230,220	READMX 00056
215 CONTINUE	READMX 00057
C IRR = 2	READMX 00058
C GO TO 1000	READMX 00059


```

220 CONTINUE
  DO 225 I=1,NF3
222 CONTINUE
  BUFFER IN (INFILE,1) (BUFF(1),BUFF(1BFCNT))
221 CONTINUE
  IF(UNIT,INFILE) 221,222,225
225 CONTINUE
230 CONTINUE
C
C      DO MATRIX SPACING
C
  IF(NM6) 235,250,240
235 CONTINUE
  IRR = 1
  GO TO 1000
240 CONTINUE
  NM2 = NM5 + NM6
  DO 245 I=1,NM2
  BUFFER IN (INFILE,1) (BUFF(1),BUFF(1BFCNT))
241 CONTINUE
  IF(UNIT,INFILE) 241,242,245
242 CONTINUE
  GO TO 245
243 CONTINUE
  IRR = 1500 + (I+1)/2
  GO TO 1000
245 CONTINUE
250 CONTINUE
C
C      READ B HEADER CARD
C
  BUFFER IN (INFILE,1) (B(1),B(16))
300 CONTINUE
  IF (UNIT,INFILE) 300,310,305
305 CONTINUE
  IRR = 1500 + NM5 + 1
  GO TO 1000
310 CONTINUE
C
C      SET PARAMETERS AND SIZES
C
  ID(2) =IB(1)
  M      =IB(2)
  N      =IB(3)
  MTN    =IB(6)
  DO 325 I=7,16
  PARM(I-6) = B(I)
325 CONTINUE
C
C      TEST FOR PROPER SIZES
C
  IF(M.GT.0.AND.N.GT.0.AND.MTN.LE.IBFCNT) GO TO 350
  IRR = 4
  GO TO 1000
350 CONTINUE
C
C      READ THE ARRAY

```

```

READMX 00060
READMX 00061
READMX 00062
READMX 00063
READMX 00064
READMX 00065
READMX 00066
READMX 00067
READMX 00068
READMX 00069
READMX 00070
READMX 00071
READMX 00072
READMX 00073
READMX 00074
READMX 00075
READMX 00076
READMX 00077
READMX 00078
READMX 00079
READMX 00080
READMX 00081
READMX 00082
READMX 00083
READMX 00084
READMX 00085
READMX 00086
READMX 00087
READMX 00088
READMX 00089
READMX 00090
READMX 00091
READMX 00092
READMX 00093
READMX 00094
READMX 00095
READMX 00096
READMX 00097
READMX 00098
READMX 00099
READMX 00100
READMX 00101
READMX 00102
READMX 00103
READMX 00104
READMX 00105
READMX 00106
READMX 00107
READMX 00108
READMX 00109
READMX 00110
READMX 00111
READMX 00112
READMX 00113
READMX 00114
READMX 00115
READMX 00116

```

C		READMX	00117
400	CONTINUE	READMX	00118
	BUFFER IN (INFILE,1) (BUFF(1),BUFF(MTN))	READMX	00119
410	CONTINUE	READMX	00120
	IF (UNIT,INFILE) 410,420,415	READMX	00121
415	CONTINUE	READMX	00122
	IRR = 1500 +NMS +1	READMX	00123
	GO TO 1000	READMX	00124
420	CONTINUE	READMX	00125
C		READMX	00126
C	IF K=0 LEAVE THE MATRIX IN THE BUFF AREA AND EXIT	READMX	00127
C	IF K.GT.0 TRANSFER BUFF TO ARRAY A	READMX	00128
C		READMX	00129
	IF(K.LE.0) GO TO 1000	READMX	00130
C		READMX	00131
C	TRANSFORM BUFF TO ARRAY A	READMX	00132
C		READMX	00133
	IF(ITYPE.EQ.7)COMPLEX) GO TO 475	READMX	00134
	IX = 0	READMX	00135
	DO 450 I=1,M	READMX	00136
	DO 450 J=1,N	READMX	00137
	IX = IX +1	READMX	00138
	A(I,J) = BUFF(IX)	READMX	00139
450	CONTINUE	READMX	00140
	GO TO 500	READMX	00141
475	CONTINUE	READMX	00142
	K2 = K+K	READMX	00143
	CALL CBUFR(A,K2,M,N,BUFF)	READMX	00144
C		READMX	00145
500	CONTINUE	READMX	00146
C		READMX	00147
1000	CONTINUE	READMX	00148
	RETURN	READMX	00149
	END	READMX	00150

```

SUBROUTINE CBUFR(A,K2,M,N,BUFF)
DIMENSION A(K2,1),BUFF(1)
C
C      PUTS A COMPLEX ARRAY STORED IN BUFF INTO FORTRAN ARRAY A
C
  IX = 0
  IX2 = MM
  MM = M+M-1
  DO 100 I=1,MM,2
  DO 100 J=1,N
    IX = IX + 1
    IX2 = IX2 + 1
    A(I,J) = BUFF(IX)
    A(I+1,J) = BUFF(IX2)
  100 CONTINUE
  RETURN
  END

```

```

READMX 00151
READMX 00152
READMX 00153
READMX 00154
READMX 00155
READMX 00156
READMX 00157
READMX 00158
READMX 00159
READMX 00160
READMX 00161
READMX 00162
READMX 00163
READMX 00164
READMX 00165
READMX 00166
READMX 00167

```

SUBROUTINE WRTEMX(IOUTFL, MXWRIT, RANDOU, NFS, NMS, LS, NMR, LWS,	WRTEMX 00002
1 K, ID, A, ITYPE, M, N, PARM, IRR)	WRTEMX 00003
C	WRTEMX 00004
C ROUTINE TO WRITE A MATRIX ON TAPE OR DISK FILE.	WRTEMX 00005
C THIS VERSION WILL ONLY WORK WITH SEQUENTIAL FILES.	WRTEMX 00006
C SOME VARIABLES ARE PASSED FOR RANDOM OPERATION BUT	WRTEMX 00007
C ARE NOT CURRENTLY USED.	WRTEMX 00008
C	WRTEMX 00009
C IOUTFL - TAPE NUMBER OR LEFT-JUSTIFIED FILE NAME	WRTEMX 00010
C MXWRIT - .T. SNARK FORMAT (NOT USED)	WRTEMX 00011
C .F. TEL001 FORMAT (NOT USED).	WRTEMX 00012
C RANDOU - .T. RANDOM FILE (NOT USED)	WRTEMX 00013
C .F. SEQUENTIAL FILE (NOT USED)	WRTEMX 00014
C NFS - NUMBER OF FILES TO SPACE (SEQ. ONLY)	WRTEMX 00015
C NMS - NUMBER OF MATRICES TO SPACE	WRTEMX 00016
C LS - LEVEL NUMBER TO SPACE (NOT USED)	WRTEMX 00017
C NMR - IDENTIFIER (NAME OR NUMBER) (NOT USED)	WRTEMX 00018
C LWS - LEVEL NUMBER OF THIS MATRIX (NOT USED)	WRTEMX 00019
C K - ROW DIMENSION OF A	WRTEMX 00020
C - (IF 0, MATRIX IS ALREADY IN /RWBUFF/)	WRTEMX 00021
C ID - ARRAY CONTAINING MATRIX NAME	WRTEMX 00022
C A - ARRAY CONTAINING MATRIX	WRTEMX 00023
C ITYPE - REAL, DIAGONAL, NULL, MIXED, COMPLEX	WRTEMX 00024
C M - ROW DIMENSION OF MATRIX	WRTEMX 00025
C N - COLUMN DIMENSION OF MATRIX	WRTEMX 00026
C PARM - 10 WORD PARAMETER ARRAY	WRTEMX 00027
C IRR - ERROR RETURN	WRTEMX 00028
C = 0, NO ERROR	WRTEMX 00029
C 1, MATRIX SPACING IS NEGATIVE	WRTEMX 00030
C 2, FILE SPACING IS NEGATIVE	WRTEMX 00031
C 4, M,N DIMENSIONS ARE .GT. IBGCNT	WRTEMX 00032
C 1500 + 1, ENCOUNTERED EOF AFTER MATRIX I/2 WHILE	WRTEMX 00033
C SKIPPING MATRICES.	WRTEMX 00034
C	WRTEMX 00035
C	WRTEMX 00036
C DIMENSION ID(1), A(K,1), PARM(10), B(16)	WRTEMX 00037
C DIMENSION IB(16)	WRTEMX 00038
C EQUIVALENCE (B,IB)	WRTEMX 00039
C LOGICAL MXWRIT,RANDOU	WRTEMX 00040
C	WRTEMX 00041
C COMMON /RWBUFF/ BFCODE,IBFCNT, BUFF(3280)	WRTEMX 00042
C DATA BFCODE,IBFCNT /8HBUFSIZE, 3280 /	WRTEMX 00043
C	WRTEMX 00044
C	WRTEMX 00045
C TEST FOR PROPER SIZE	WRTEMX 00046
C	WRTEMX 00047
C	WRTEMX 00048
C ISIZ = M*N	WRTEMX 00049
C IF(ITYPE.EQ.7HCOMPLEX) ISIZ = ISIZ+ISIZ	WRTEMX 00050
C IF(ISIZ.LE.IBFCNT) GO TO 205	WRTEMX 00051
C IRR = 4	WRTEMX 00052
C GO TO 1000	WRTEMX 00053
C	WRTEMX 00054
C 205 CONTINUE	WRTEMX 00055
C IF(K.LE.0) GO TO 300	WRTEMX 00056
C IF(K.GE.W) GO TO 210	WRTEMX 00057
C IRR = 5	WRTEMX 00058
C GO TO 1000	

210 CONTINUE	WRTMX 00059
C	WRTMX 00060
C DO FILE SPACING	WRTMX 00061
C	WRTMX 00062
IF(NF3) 215,230,220	WRTMX 00063
215 CONTINUE	WRTMX 00064
IRR = 2	WRTMX 00065
GO TO 1000	WRTMX 00066
220 CONTINUE	WRTMX 00067
DO 225 I=1,NF3	WRTMX 00068
222 CONTINUE	WRTMX 00069
BUFFER IN (IOUTFL,1) (BUFF(1),BUFF(1BFCNT))	WRTMX 00070
221 CONTINUE	WRTMX 00071
IF(UNIT,IOUTFL) 221,222,225	WRTMX 00072
225 CONTINUE	WRTMX 00073
230 CONTINUE	WRTMX 00074
C	WRTMX 00075
C DO MATRIX SPACING	WRTMX 00076
C	WRTMX 00077
IF(NM5) 235,250,240	WRTMX 00078
235 CONTINUE	WRTMX 00079
IRR = 1	WRTMX 00080
GO TO 1000	WRTMX 00081
240 CONTINUE	WRTMX 00082
NM2 = NM5 + NM5	WRTMX 00083
DO 245 I=1,NM2	WRTMX 00084
BUFFER IN (IOUTFL,1) (BUFF(1),BUFF(1BFCNT))	WRTMX 00085
241 CONTINUE	WRTMX 00086
IF (UNIT,IOUTFL) 241,242,243	WRTMX 00087
242 CONTINUE	WRTMX 00088
GO TO 245	WRTMX 00089
243 CONTINUE	WRTMX 00090
IRR = 1500 + (I+1)/2	WRTMX 00091
GO TO 1000	WRTMX 00092
245 CONTINUE	WRTMX 00093
250 CONTINUE	WRTMX 00094
C	WRTMX 00095
C CREATE B HEADER RECORD	WRTMX 00096
C	WRTMX 00097
300 CONTINUE	WRTMX 00098
IB(1) = ID(2)	WRTMX 00099
IB(2) = M	WRTMX 00100
IB(3) = N	WRTMX 00101
B(4) = 0	WRTMX 00102
IB(5) = 0	WRTMX 00103
IB(6) = ISIZ	WRTMX 00104
DO 325 I=7,16	WRTMX 00105
B(I) = PARM(I-6)	WRTMX 00106
325 CONTINUE	WRTMX 00107
C	WRTMX 00108
IF(K.LE.0) GO TO 400	WRTMX 00109
C	WRTMX 00110
C PUT ARRAY A INTO BUFFER	WRTMX 00111
C	WRTMX 00112
IF(ITYPE.EQ.7HCOMPLEX) GO TO 375	WRTMX 00113
C	WRTMX 00114
C NOT COMPLEX PUT INTO BUFFER.	WRTMX 00115

C	IX = 0	WRTMX 00116
	DO 350 I=1,M	WRTMX 00117
	DO 350 J=1,N	WRTMX 00118
	IX = IX + 1	WRTMX 00119
	BUFF(IX) = A(I,J)	WRTMX 00120
	350 CONTINUE	WRTMX 00121
	GO TO 400	WRTMX 00122
C		WRTMX 00123
C	COMPLEX, CALL ROUTINE TO STORE INTO BUFFER.	WRTMX 00124
C		WRTMX 00125
	375 CONTINUE	WRTMX 00126
	K2 = K+K	WRTMX 00127
	CALL COMBUF(A,K2,M,N,BUFF)	WRTMX 00128
	IX = 2*MXN	WRTMX 00129
C		WRTMX 00130
	400 CONTINUE	WRTMX 00131
C		WRTMX 00132
C	WRITE THE B HEADER RECORD AND THE BUFFER ARRAY RECORD	WRTMX 00133
C		WRTMX 00134
	BUFFER OUT (IOUTFL,1) (B(1),B(16))	WRTMX 00135
	500 CONTINUE	WRTMX 00136
	IF (UNIT,IOUTFL) 500,510,510	WRTMX 00137
	510 CONTINUE	WRTMX 00138
C		WRTMX 00139
	BUFFER OUT (IOUTFL,1) (BUFF(1),BUFF(IX))	WRTMX 00140
	520 CONTINUE	WRTMX 00141
	IF (UNIT,IOUTFL) 520,530,530	WRTMX 00142
	530 CONTINUE	WRTMX 00143
C		WRTMX 00144
	1000 CONTINUE	WRTMX 00145
	RETURN	WRTMX 00146
	END	WRTMX 00147
		WRTMX 00148

SUBROUTINE COMBUF(A,K2,M,N,BUFF)	WRTEMX 00149
DIMENSION A(K2,1),BUFF(1)	WRTEMX 00150
C	WRTEMX 00151
C PUTS COMPLEX ARRAY A INTO BUFFER BUFF	WRTEMX 00152
C	WRTEMX 00153
IX = 0	WRTEMX 00154
IX2 = MMN	WRTEMX 00155
MM = M+M-1	WRTEMX 00156
DO 100 I=1,MM,2	WRTEMX 00157
DO 100 J=1,N	WRTEMX 00158
IX = IX + 1	WRTEMX 00159
IX2 = IX2 + 1	WRTEMX 00160
BUFF(IX) = A(I,J)	WRTEMX 00161
BUFF(IX2) = A(I+1,J)	WRTEMX 00162
100 CONTINUE	WRTEMX 00163
RETURN	WRTEMX 00164
END	WRTEMX 00165

OVERLAY (AFMBOX,1,1)	DATAPP 00002
PROGRAM DATAPP	DATAPP 00003
COMMON /CONTRL/ PREVEX,OMACH, TITLE(8), PRVGEOM,PRVMODE,DIHW,DIHT,	CONTRL 00002
1 DEFAULT	CONTRL 00003
LOGICAL PRVGEOM,PRVMODE,DIHW,DIHT,DEFAULT	CONTRL 00004
COMMON /PROBLM/ XMACH,NMODES,NTSLOF,NKVALS,SMOOTH,NDEG,CRDFIT,	PROBLM 00002
1 EXAIC,SUBDV,PLYWOOD	PROBLM 00003
LOGICAL SMOOTH,CRDFIT,EXAIC,SUBDV,PLYWOOD	PROBLM 00004
COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUTP,	FILES 00002
1 IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC	FILES 00003
COMMON / KERN / ERR,MXSKRN,IPKERN,NPLKRN,NSPATK,NROEA	KERN 00002
COMMON /KVAL / IKVAL,XKVAL(20), XKS(20)	KVAL 00002
DIMENSION XK1(20)	DATAPP 00009
EQUIVALENCE (XK1,XKVAL)	DATAPP 00010
COMMON /IOCONT/ OPLAIC,OSPAIC,WTGEOM,WTGNMF,WTSL,WTBL,FRBOX,	IOCONT 00002
1 FRPAIC,FRSAIC,FRMODS,FRCOEF,FRDW,FRSW,FRVP,	IOCONT 00003
2 FRBL,FRDCP,FRGNMF,FRGNAC,FRSL,FRLW,FRNW,FRCH	BCSFRB 00001
EQUIVALENCE (FRUW,FRDW)	IOCONT 00005
LOGICAL OPLAIC,OSPAIC,WTGEOM,WTGNMF,WTSL,WTBL,FRBOX,FRPAIC,	IOCONT 00006
1 FRSAIC,FRMODS,FRCOEF,FRDW,FRSW,FRVP,FRBL,FRSL,FRGNMF,	IOCONT 00007
2 FRDCP,FRGNAC,FRUW,FRLW,FRNW,FRCH	BCSFRB 00002
COMMON / MODES/ SYM,SYMT,MTYPEW,MTYPEP	MODCOM 00002
COMMON /GEOMTY/ COPLAN,NSUBDV,XSUBDV,NSUBD2,NSUBCN,NSURF,	GEOMTY 00002
1 B1,B1BETA,B1S,B1BTAS,WLAX,WLAZ,PSIW,	GEOMTY 00003
2 MXBW,MXBBW,MYBW,MYBBW,MXBSW,MYBSW,MYBBSW,	GEOMTY 00004
3 IXBW,XCENTR	GEOMTY 00005
LOGICAL COPLAN	GEOMTY 00006
COMMON /SAMPLW/ ISMPLW,ICHORD(10),IBOXF(10),IBOXL(10),ZLOC(10)	SAMPLW 00002
COMMON /CHECKFR/ DPPCFR,GEOPFR,MODCFR,AICCFR,NMSCFR,SMCFR,GAFCFR	CHECKFR 00002
LOGICAL DPPCFR, GEOPFR, MODCFR, AICCFR, NMSCFR, SMCFR, GAFCFR	CHECKFR 00003
EQUIVALENCE (CHECKFR,DPPCFR)	DATAPP 00016
LOGICAL CHECKFR	DATAPP 00017
COMMON /ARRAYS/ KBXCDW,LBXCDW,LBOXC,KBXCDT,LBXCDT,KJALPH,LJALPH,	ARRAYS 00002
1 KALPHA,KKERNL,LKERNL,KPNTRM,LPNTRM,KDEFSL,KELPHI,	ARRAYS 00003
2 LMODES,KPNTSD,LPNTSD,KSDW,LSDW,KPNTDW,LPNTDW,	ARRAYS 00004
3 KDW,LDW,KTVP,LTVP	ARRAYS 00005
LOGICAL MXWRIT,RANDOU,MXREAD,RANDIN	FTNX1 00005
INTEGER Oaic, OSAIC	FTNX1 00006
EQUIVALENCE (MACH,XMACH)	DATAPP 00020
REAL MACH	DATAPP 00021
DATA EXEC /ICHAFMBOX EXC /	FTNX1 00007
NAMelist /CARDB / XMACH	DATAPP 00022
NAMelist /CARDC / DEFAULT,PRVGEOM,PRVMODE,SYM,MTYPEW,MTYPEP,	DATAPP 00023
1 NSURF,DIHW,DIHT,ISMPLW,WTGNMF,WTBL,	DATAPP 00024
2 FRGNMF,FRBL,FRSL,FRPAIC,FRSAIC,FRCOEF,FRMODS,	DATAPP 00025
X PRDCP,FRGNAC,FRLW,FRNW,FRUW,FRCH,	BCSFRB 00003
3 FRBOX,FRDW,FRSW,FRVP,SUBDV,EXAIC,SMOOTH,NDEG,	DATAPP 00027
4 DPPCFR,GEOPFR,MODCFR,AICCFR,NMSCFR,SMCFR,GAFCFR,	DATAPP 00028
5 NROEA,CRDFIT,PLYWOOD	DATAPP 00029
NAMelist /CARDD / Oaic,MAIC,OSAIC,NSAIC,INTAPE,NOUTP,INFSP,IOUFSP	DATAPP 00030
NAMelist /CARDE / XK1,XKS,XKVAL	DATAPP 00032
MXWRIT = .FALSE.	DATAPP 00033
RANDOU = .FALSE.	DATAPP 00034
MXREAD = .FALSE.	DATAPP 00035
RANDIN = .FALSE.	DATAPP 00036
IF (PREVEX.EQ.EXEC) GO TO 200	DATAPP 00037
OMACH = 0.0	DATAPP 00038

C
C SET CONTROL PARAMETERS TO DEFAULT OPTIONS

```

      DEFAULT = .FALSE.
100 CONTINUE
      PRVGEOM = .FALSE.
      PRVMODE = .FALSE.
      SYM      = 1.0
      NTYPEW   = 2
      NTYPET   = 2
      NSURF    = 1
      DIHW     = .TRUE.
      DIHT     = .TRUE.
      ISMPLW   = 0
      WTGNMF   = .TRUE.
      WTBL     = .FALSE.
      PRGNMF   = .TRUE.
      PRBL     = .FALSE.
      PRSL     = .FALSE.
      PRPAIC   = .FALSE.
      PRSAIC   = .FALSE.
      PRCOEF   = .FALSE.
      PRMODS   = .FALSE.
      PRBOX    = .FALSE.
      PRDW     = .FALSE.
      PRSW     = .FALSE.
      PRDCP    = .FALSE.
      PRGMAC   = .FALSE.
      PRLW     = .FALSE.
      PRNW     = .FALSE.
      PRUW     = .FALSE.
      PRVP     = .FALSE.
      PRCH     = .FALSE.
      SUBDV    = .FALSE.
      EXAIC    = .FALSE.
      SMOOTH   = .FALSE.
      ORDFIT   = .FALSE.
      NDEG     = 0
      NRONEA   = 0
      PLYWOOD  = .FALSE.
      DPPCPR   = .FALSE.
      GEOCPR   = .FALSE.
      MODCPR   = .FALSE.
      AICCPR   = .FALSE.
      NMSCPR   = .FALSE.
      SMCPR    = .FALSE.
      GAFCPR   = .FALSE.
      PREVEX   = EXEC
      IF(DEFAULT) 400,300
200 CONTINUE
      OMACH = XMACH
300 CONTINUE
      READ (NT5,9005) TITLE
9005 FORMAT(8A10)
      READ(NT5,CARD8)
      IF(XMACH.GT.1.0) GO TO 310
      WRITE (NT6,8005) XMACH
      CALL FLUSH(1)

```

```

DATAPP 00039
DATAPP 00040
DATAPP 00041
DATAPP 00042
DATAPP 00043
DATAPP 00044
DATAPP 00045
DATAPP 00046
DATAPP 00047
DATAPP 00048
DATAPP 00049
DATAPP 00050
DATAPP 00051
DATAPP 00052
DATAPP 00053
DATAPP 00054
DATAPP 00055
DATAPP 00056
DATAPP 00057
DATAPP 00058
DATAPP 00059
DATAPP 00060
DATAPP 00061
DATAPP 00062
DATAPP 00063
DATAPP 00064
DATAPP 00065
DATAPP 00066
DATAPP 00067
DATAPP 00068
DATAPP 00069
BCSFRB 00004
DATAPP 00070
DATAPP 00071
DATAPP 00072
DATAPP 00073
DATAPP 00074
DATAPP 00075
DATAPP 00076
DATAPP 00077
DATAPP 00078
DATAPP 00079
DATAPP 00080
DATAPP 00081
DATAPP 00082
DATAPP 00083
DATAPP 00084
DATAPP 00085
DATAPP 00086
DATAPP 00087
DATAPP 00088
DATAPP 00089
DATAPP 00090
DATAPP 00091
DATAPP 00092
DATAPP 00093
DATAPP 00094

```

310 CONTINUE	DATAPP 00095
IF(XMACH.GE.1.2) GO TO 320	DATAPP 00096
WRITE (NT6,8010)	DATAPP 00097
GO TO 350	DATAPP 00098
320 CONTINUE	DATAPP 00099
IF(XMACH.LE.3.0) GO TO 350	DATAPP 00100
C	DATAPP 00101
C MACH NO. GREATER THAN 3.0	DATAPP 00102
IF(XMACH.LT.5.0) GO TO 340	DATAPP 00103
WRITE (NT6,8015) XMACH	DATAPP 00104
CALL FLUSH(1)	DATAPP 00105
340 CONTINUE	DATAPP 00106
WRITE (NT6,8020)	DATAPP 00107
350 CONTINUE	DATAPP 00108
8005 FORMAT(52H0*** MACH NUMBER OF LESS THAN 1.0 CAN NOT BE USED.	DATAPP 00109
1 14HMACH NUMBER = E15.6, 6H ***)	DATAPP 00110
8010 FORMAT(62H0*** WARNING -- MACH NUMBER LESS THAN 1.2 IS BEING USED.	DATAPP 00111
1 ***)	DATAPP 00112
8015 FORMAT(43H0*** MACH NUMBER GREATER THAN 5.0. XMACH = E15.6,	DATAPP 00113
1 25H PROGRAM TERMINATED. ***)	DATAPP 00114
8020 FORMAT(64H0*** WARNING -- MACH NUMBER GREATER THAN 3.0 IS BEING US	DATAPP 00115
ED. ***)	DATAPP 00116
READ (NT5,CARDC)	DATAPP 00117
IF(SUBDV) 500,510	DATAPP 00118
500 NSUBDV =3	DATAPP 00119
GO TO 515	DATAPP 00120
510 NSUBDV =1	DATAPP 00121
515 CONTINUE	DATAPP 00122
C	DATAPP 00123
IF(DEFAULT) 100,400	DATAPP 00124
C	DATAPP 00125
C CARD D	DATAPP 00126
400 CONTINUE	DATAPP 00127
OAIC = 0	DATAPP 00128
MAIC = 0	DATAPP 00129
OBAIC = 0	DATAPP 00130
NBAIC = 0	DATAPP 00131
INTAPE = 0	DATAPP 00132
NOUFP = 1	DATAPP 00133
INFSP = 0	DATAPP 00134
IOUFSP = 0	DATAPP 00135
READ (NT5,CARDD)	DATAPP 00136
C	DATAPP 00137
IF(OAIC.EQ.0) GO TO 520	DATAPP 00138
NPLAIC = OAIC	DATAPP 00139
OPLAIC = .TRUE.	DATAPP 00140
GO TO 530	DATAPP 00141
520 CONTINUE	DATAPP 00142
NPLAIC = 0	DATAPP 00143
OPLAIC = .FALSE.	DATAPP 00144
530 CONTINUE	DATAPP 00145
IF(MAIC.EQ.0) GO TO 540	DATAPP 00146
NPLAIC = MAIC	DATAPP 00147
OPLAIC = .FALSE.	DATAPP 00148
540 CONTINUE	DATAPP 00149
C	DATAPP 00150
C DETERMINE OPTIONS OF SPATIAL KERNELS	DATAPP 00151

IF(OBAIC.EQ.0) GO TO 560	DATAPP 00152
NSPAIC = OBAIC	DATAPP 00153
OSPAIC = .TRUE.	DATAPP 00154
GO TO 570	DATAPP 00155
560 CONTINUE	DATAPP 00156
NSPAIC = 0	DATAPP 00157
OSPAIC = .FALSE.	DATAPP 00158
570 CONTINUE	DATAPP 00159
IF(NSAIC.EQ.0) GO TO 580	DATAPP 00160
NSPAIC = NSAIC	DATAPP 00161
OSPAIC = .FALSE.	DATAPP 00162
580 CONTINUE	DATAPP 00163
C	DATAPP 00164
IF (NOUTP.NE.0) GO TO 600	DATAPP 00165
IF(WTGNF) WRITE (NT6,9041)	DATAPP 00166
IF(WTBL) WRITE (NT6,9042)	DATAPP 00167
WTGNF = .FALSE.	DATAPP 00168
WTBL = .FALSE.	DATAPP 00169
600 CONTINUE	DATAPP 00170
C CARD E	DATAPP 00171
DO 610 I=1,20	DATAPP 00172
XK1(I) = -1.	DATAPP 00173
XK1(I) = -1.	DATAPP 00174
610 CONTINUE	DATAPP 00175
READ(NT5,CARDE)	DATAPP 00176
DO 620 I=1,20	DATAPP 00177
IF(XK1(I).NE.-1.0.OR .XK1(I).NE.-1.0) GO TO 620	DATAPP 00178
NKVALS = I-1	DATAPP 00179
GO TO 625	DATAPP 00180
620 CONTINUE	DATAPP 00181
NKVALS = 20	DATAPP 00182
625 CONTINUE	DATAPP 00183
C	DATAPP 00184
WRITE (NT6,9500)	DATAPP 00185
WRITE (NT6,9501)	DATAPP 00186
WRITE (NT6,9551) TITLE	DATAPP 00187
WRITE (NT6,9580) XMACH	DATAPP 00188
IF(DEFAULT) WRITE (NT6,9575)	DATAPP 00189
IF(SYM.EQ.1.0) WRITE (NT6,9552)	DATAPP 00190
IF(SYM.EQ.-1.) WRITE (NT6,9553)	DATAPP 00191
IF (PLYWOOD) WRITE (NT6,9554)	DATAPP 00192
IF (DPPCFR.AND.PLYWOOD) WRITE (NT6,9558)	DATAPP 00193
IF(.NOT.SUBDV) WRITE (NT6,9572)	DATAPP 00194
IF(SUBDV) WRITE (NT6,9573)	DATAPP 00195
IF (SUBDV .AND. NRCWEA .NE. 0) WRITE (NT6,9546) NRCWEA	DATAPP 00196
IF(NSURF.EQ.1) WRITE(NT6,9556)	DATAPP 00197
IF(NSURF.EQ.2) WRITE(NT6,9557)	DATAPP 00198
IF(.NOT.EXAIC) WRITE (NT6,9576)	DATAPP 00199
IF(EXAIC) WRITE (NT6,9577)	DATAPP 00200
IF(CRDFIT) SMOOTH = .FALSE.	DATAPP 00201
IF(SMOOTH) WRITE (NT6,9581) NDEG	DATAPP 00202
IF(CRDFIT) WRITE (NT6,9585) NDEG	DATAPP 00203
IF (.NOT. (SMOOTH .OR. CRDFIT) .OR. NDEG .LE. 10) GO TO 630	DATAPP 00204
NDEG = 10	DATAPP 00205
WRITE (NT6,9043) NDEG	DATAPP 00206
630 CONTINUE	DATAPP 00207
IF(PRBOX) WRITE (NT6,9569)	DATAPP 00208

IF (PRCOEF)	WRITE (NT6,9582)	DATAPP	00209
IF (PRMODS)	WRITE (NT6,9568)	DATAPP	00210
IF (PRFAIC)	WRITE (NT6,9583)	DATAPP	00211
IF (PRSAIC)	WRITE (NT6,9584)	DATAPP	00212
IF (PRCW)	WRITE (NT6,9570)	DATAPP	00213
IF (PRSW)	WRITE (NT6,9578)	DATAPP	00214
IF (PRLW)	WRITE (NT6,9544)	DATAPP	00215
IF (PRNW)	WRITE (NT6,9545)	DATAPP	00216
IF (PRVP)	WRITE (NT6,9571)	DATAPP	00217
IF (PRBL)	WRITE (NT6,9565)	DATAPP	00218
IF (PRSL)	WRITE (NT6,9566)	DATAPP	00219
IF (PRCM)	WRITE (NT6,9567)	BCSFRB	00005
IF (PRDCP)	WRITE (NT6,9542)	DATAPP	00220
IF (PRGNAC)	WRITE (NT6,9543)	DATAPP	00221
IF (PRGNAF)	WRITE (NT6,9564)	DATAPP	00222
IF (WTBL)	WRITE (NT6,9562)	DATAPP	00223
IF (WTGNAF)	WRITE (NT6,9561)	DATAPP	00224
IF (PRVGEOM)	WRITE (NT6,9531)	DATAPP	00225
IF (PRVNODE)	WRITE (NT6,9532)	DATAPP	00226
IF (MTYPEW.EQ.1)	WRITE (NT6,9533)	DATAPP	00227
IF (MTYPEW.EQ.2)	WRITE (NT6,9534)	DATAPP	00228
IF (MTYPEW.EQ.3)	WRITE (NT6,9535)	DATAPP	00229
IF (NSURF.EQ.1)	GO TO 650	DATAPP	00230
IF (MTYPEP.EQ.1)	WRITE (NT6,9536)	DATAPP	00231
IF (MTYPEP.EQ.2)	WRITE (NT6,9537)	DATAPP	00232
IF (MTYPEP.EQ.3)	WRITE (NT6,9538)	DATAPP	00233
650 CONTINUE		DATAPP	00234
IF (DIHW)	WRITE (NT6,9539)	DATAPP	00235
IF (NSURF.EQ.1)	GO TO 660	DATAPP	00236
IF (DIHT)	WRITE (NT6,9540)	DATAPP	00237
660 CONTINUE		DATAPP	00238
ERR = 0.01		DATAPP	00239
IF (EXAIC)	ERR = 0.0001	DATAPP	00240
C		DATAPP	00241
C	THIS SET OF VARIABLES ARE DIMENSION SIZES FOR ARRAYS.	DATAPP	00242
C	THE NUMBER IS THE DIMENSION OF THE ARRAY.	DATAPP	00243
C	FOR DOUBLE DIMENSIONED ARRAYS IT IS THE LARGEST NUMBER,	DATAPP	00244
C	NOT THE PRODUCT OF THE TWO DIMENSIONS.	DATAPP	00245
C		DATAPP	00246
	KKERNL = 1	DATAPP	00247
	LKERNL = 1640	DATAPP	00248
	LBXCDW = 150	DATAPP	00249
	LBXCDT = 90	DATAPP	00250
	LBXNC = 8	DATAPP	00251
	LJALPH = 200	DATAPP	00252
	LPNTRM = 100	DATAPP	00253
	LXODES = 1000	DATAPP	00254
	LPNTSD = 30	DATAPP	00255
	LSDW = 600	DATAPP	00256
	LPNTDW = 100	DATAPP	00257
	LDW = 1275	DATAPP	00258
	LYVP = 250	DATAPP	00259
C		DATAPP	00260
	WRITE (NT6,6001)	DATAPP	00261
	WRITE (NT6,6002) Oaic,Naic,Obaic,Nsaic,Intape,Infsp,Noutp,Ioufsp	DATAPP	00262
C		DATAPP	00263
	8001 FORMAT(1H0/45X, 39HTHE FOLLOWING TAPE SETUP IS REQUESTED - /)	DATAPP	00264

6002	FORMAT(51X,*OLD AIC TAPE ==*,I3,/51X,*NEW AIC TAPE ==*,I3,/	DATAPP	00265
1	51X,*OLD SPATIAL AIC TAPE ==*,I3,/51X,*NEW SPATIAL AIC TAPE ==*,I3,	DATAPP	00266
2	/ 51X,*INPUT DATA TAPE ==*,I3,* SPACED*,I3,* FILES,*	DATAPP	00267
3	/ 51X,*OUTPUT TAPE ==*,I3,* SPACED*,I3,* FILES,* //)	DATAPP	00268
C		DATAPP	00269
C	PRINT THE NKVAL OR XKS ARRAY.	DATAPP	00270
C		DATAPP	00271
	IF(XK1(1).EQ.-1.0) GO TO 700	DATAPP	00272
	WRITE(NT6,6003)	DATAPP	00273
	WRITE(NT6,6004) (XK1(I),I=1,NKVALS)	DATAPP	00274
	GO TO 900	DATAPP	00275
C		DATAPP	00276
700	CONTINUE	DATAPP	00277
	IF(XKS(1).EQ.-1.0) GO TO 800	DATAPP	00278
	WRITE(NT6,6005)	DATAPP	00279
	WRITE(NT6,6004) (XKS(I),I=1,NKVALS)	DATAPP	00280
	GO TO 900	DATAPP	00281
C		DATAPP	00282
800	CONTINUE	DATAPP	00283
	WRITE(NT6,6006)	DATAPP	00284
C		DATAPP	00285
900	CONTINUE	DATAPP	00286
C		DATAPP	00287
6003	FORMAT(1HD,29X, *THE FOLLOWING IS THE REDUCED FREQUENCY ARRAY BASE	DATAPP	00288
	1D ON BOX LENGTH* /)	DATAPP	00289
6005	FORMAT(1HD,29X, *THE FOLLOWING IS THE REDUCED FREQUENCY ARRAY BASE	DATAPP	00290
	1D ON WING SEMI-SPAN* /)	DATAPP	00291
6004	FORMAT(1H / (31X,6F11.5))	DATAPP	00292
6006	FORMAT(49HD*** WARNING -- NO REDUCED FREQUENCIES SPECIFIED.	DATAPP	00293
1	51H PROGRAM WILL TERMINATE AFTER GEOMETRY SECTION ***)	DATAPP	00294
C		DATAPP	00295
C		DATAPP	00296
1000	RETURN	DATAPP	00297
9500	FORMAT(1H1,29X,58(1H*),/30X,1H*,56X,1H*,/30X,58H* UNSTEADY AE	DATAPP	00299
	1RODYNAMICS OF WING-HORIZONTAL TAIL *,/30X,1H*,12X,*CONFIGURATI	DATAPP	00300
	2ONS IN SUPERSONIC FLOW*11X,1H*,/30X,1H*,56X,1H*,/30X,58H* PREP	DATAPP	00301
	3ARED UNDER CONTRACT NO. AF 33615-70-C-1126 *,/30X,1H*,20X,*PRO	DATAPP	00302
	4JECT NO. 1370*, 20X,1H*,/30X,1H*,56X,1H*)	DATAPP	00303
9501	FORMAT(30X,1H*,5X,*FOR DEPARTMENT OF THE AIR FORCE*,19X,1H*, /	DATAPP	00304
1	30X,1H*,10X,*AERONAUTICAL SYSTEMS DIVISION*, 17X,1H*, /	DATAPP	00305
2	30X,1H*,10X,*AIR FORCE FLIGHT DYNAMICS LABORATORY*,10X,1H*, /	DATAPP	00306
3	30X,1H*,10X,*WRIGHT-PATTERSON AIR FORCE BASE*,15X,1H*, /	DATAPP	00307
4	30X,1H*,56X,1H*, /	DATAPP	00308
5	30X,1H*,5X,*BY THE BOEING COMPANY*,28X,1H*, /	DATAPP	00309
6	30X,1H*,10X,*COMMERCIAL AIRPLANE DIVISION*,18X,1H*, /	DATAPP	00310
7	30X,1H*,10X,*SEATTLE, WASHINGTON*, 27X,1H*, /	DATAPP	00311
8	30X,1H*,56X,1H*,/ 30X,58(1H*),/)	DATAPP	00312
9041	FORMAT(72HD*** WARNING -- NO OUTPUT TAPE WAS REQUESTED FOR GENERAL	DATAPP	00313
	1IZED FORCES. ***)	DATAPP	00314
9042	FORMAT(63HD*** WARNING -- NO OUTPUT TAPE WAS REQUESTED FOR BOX LIF	DATAPP	00315
	1TS. ***)	DATAPP	00316
9043	FORMAT(54HD*** WARNING -- ORDER FOR VELOCITY POTENTIAL SMOOTHING	DATAPP	00317
1	36H TOO LARGE. IT HAS BEEN REDUCED TO ,I2, 4H ***)	DATAPP	00318
9551	FORMAT(1HD,5X,7HTITLE -,13X,8A10,13X,7H- TITLE /1HD/45X,	DATAPP	00319
1	37HTHE FOLLOWING OPTIONS ARE REQUESTED - /)	DATAPP	00320
9531	FORMAT(51X,*GEOMETRY FROM PREVIOUS CYCLE*)	DATAPP	00321
9532	FORMAT(51X,*MODE SHAPES FROM PREVIOUS CYCLE*)	DATAPP	00322

9533 FORMAT(51X, *MODAL INPUT FOR WING IS POLYNOMIAL COEFFICIENTS*)	DATAPP	00323
9534 FORMAT(51X, *MODAL INPUT FOR WING IS ARBITRARY LOCATIONS FOR SURFACE FITTING *)	DATAPP	00324
9535 FORMAT(51X, *MODAL INPUT FOR WING IS BOX CENTER VALUES*)	DATAPP	00325
9536 FORMAT(51X, *MODAL INPUT FOR TAIL IS POLYNOMIAL COEFFICIENTS*)	DATAPP	00326
9537 FORMAT(51X, *MODAL INPUT FOR TAIL IS ARBITRARY LOCATIONS FOR SURFACE FITTING *)	DATAPP	00327
9538 FORMAT(51X, *MODAL INPUT FOR TAIL IS BOX CENTER VALUES*)	DATAPP	00328
9539 FORMAT(51X, *DIHEDRAL WING INFLUENCE CALCULATED*)	DATAPP	00329
9540 FORMAT(51X, *DIHEDRAL TAIL INFLUENCE CALCULATED*)	DATAPP	00330
9542 FORMAT(51X, *PRINT PRESSURE DIFFERENCE COEFFICIENTS*)	DATAPP	00331
9543 FORMAT(51X, *PRINT GENERALIZED AERODYNAMIC COEFFICIENTS*)	DATAPP	00332
9544 FORMAT(51X, *PRINT LONGITUDINAL WASHES ALONG SAMPLING CHORDS*)	DATAPP	00333
9545 FORMAT(51X, *PRINT NORMAL WASHES*)	DATAPP	00334
9546 FORMAT(51X, *EFFECTIVE SUBDIVIDED AREA OF*, I3, * ROWS REQUESTED*)	DATAPP	00335
9552 FORMAT(51X, *SYMMETRIC ANALYSIS*)	DATAPP	00336
9553 FORMAT(51X, *ANTI-SYMMETRIC ANALYSIS*)	DATAPP	00337
9554 FORMAT(51X, *PLYWOOD OPTION IS USED. (PLANFORM BOUNDARY DETERMINED BY BOX PATTERN.) *)	DATAPP	00338
9556 FORMAT(51X, *SINGLE PLANFORM ANALYSIS*)	DATAPP	00339
9557 FORMAT(51X, *ANALYSIS FOR 2 PLANFORMS*)	DATAPP	00340
9558 FORMAT(1HD, 100(1H\$) /// * THE SPRUCE GOOSE IS LOOSE * // 1HD, 1 100(1H\$))	DATAPP	00341
9561 FORMAT(51X, *WRITE GENERALIZED AIR FORCES ON TAPE*)	DATAPP	00342
9562 FORMAT(51X, *WRITE BOX LIFTS ON TAPE*)	DATAPP	00343
9564 FORMAT(51X, *PRINT GENERALIZED AIR FORCES*)	DATAPP	00344
9565 FORMAT(51X, *PRINT THE BOX LIFTS*)	DATAPP	00345
9566 FORMAT(51X, *PRINT THE SECTION LIFTS*)	DATAPP	00346
9567 FORMAT(51X, *SECTION MOMENTS WILL BE COMPUTED WITH MODE SHAPE ONE* / 1 51X, * ASSUMED FOR THE PITCH MODE. *)	BCSFRB	00006
9568 FORMAT(51X, *PRINT MODE SHAPES USED*)	BCSFRB	00007
9569 FORMAT(51X, *PRINT THE BOX PATTERN*)	DATAPP	00351
9570 FORMAT(51X, *PRINT THE UPWASHES ALONG SAMPLING CHORDS*)	DATAPP	00352
9571 FORMAT(51X, *PRINT THE VELOCITY POTENTIALS*)	DATAPP	00353
9572 FORMAT(51X, *BASIC (UNSUBDIVIDED) ANALYSIS WILL BE USED*)	DATAPP	00354
9573 FORMAT(51X, *SUBDIVISION WILL BE APPLIED*)	DATAPP	00355
9575 FORMAT(51X, *ALL PARAMETERS SET TO "FAULT VALUES")	DATAPP	00356
9576 FORMAT(51X, *APPROXIMATE KERNELS WILL BE USED*)	DATAPP	00357
9577 FORMAT(51X, *EXACT KERNELS WILL BE USED*)	DATAPP	00358
9578 FORMAT(51X, *PRINT THE SIDEWASHES ALONG SAMPLING CHORDS*)	DATAPP	00359
9580 FORMAT(51X, *MACH NUMBER = *, F8.6)	DATAPP	00360
9581 FORMAT(51X, *VELOCITY POTENTIALS WILL BE SMOOTHED BY A LEAST-SQUARE, 1 *RES* / 61X, *POLYNOMIAL SURFACE FIT, OF ORDER*, I2, I1H. / 2 61X, * (O = PROGRAM DETERMINED.) *)	DATAPP	00361
9582 FORMAT(51X, *PRINT MODE SHAPE POLYNOMIAL COEFFICIENTS, IF AVAILABLE, 1 *LE *)	DATAPP	00362
9583 FORMAT(51X, *PRINT THE PLANAR AIC ARRAYS USED *)	DATAPP	00363
9584 FORMAT(51X, *PRINT THE SPATIAL AIC ARRAYS USED *)	DATAPP	00364
9585 FORMAT(51X, *VELOCITY POTENTIALS WILL BE SMOOTHED BY A LEAST SQUARE, 1 *RES* / 61X, *POLYNOMIAL CHORDWISE FIT, OF ORDER*, I2, I1H. / 2 61X, * (O = PROGRAM DETERMINED.) *)	DATAPP	00365
END	DATAPP	00366
	DATAPP	00367
	DATAPP	00368
	DATAPP	00369
	DATAPP	00370
	DATAPP	00371
	DATAPP	00372

OVERLAY (AFMBOX,1,2)	GEOMBX	00002
PROGRAM GEOMBX	GEOMBX	00003
C THIS OVERLAY READS ALL GEOMETRIC INFORMATION (CARDS G TO L, OR	GEOMBX	00004
C FROM TAPE) AND COMPUTES THE INTERNAL GEOMETRY NEEDED	GEOMBX	00005
C ERRORS IN GEOMETRIC DEFINITIONS ARE CAUGHT	GEOMBX	00006
C ALL GEOMETRY IS NON-DIMENSIONALIZED BY BOX WIDTH (LENGTH)	GEOMBX	00007
C BOX CODES ARE DEFINED -	GEOMBX	00008
C 0 = NOT USED	GEOMBX	00009
C 1 = ON-PLANFORM	GEOMBX	00010
C 2 = DIAPHRAGM	GEOMBX	00011
C 3 = WAKE	GEOMBX	00012
C ALPHA ARRAY, FRACTIONAL PART OF EDGE BOXES, IS COMPUTED	GEOMBX	00013
C MAXIMUM PLANAR AIC ARRAY SIZE IS DETERMINED	GEOMBX	00014
C FOR EACH CHORD REQUIRING A SPATIAL AIC ARRAY, DETERMINE	GEOMBX	00015
C WHICH AIC ARRAY TO USE (KPTWM, KPTTT, KPTWRT, DPTWLT)	GEOMBX	00016
C EL, THE VERTICAL DISTANCE SEPARATING THE SURFACES	GEOMBX	00017
C YBAR, THE HORIZONTAL OFFSET	GEOMBX	00018
C MUAIC ARRAY, A MAP OF NEEDED AIC VALUES	GEOMBX	00019
C	GEOMBX	00020
COMMON /CONTRL/ PREVEX, CMACH, TITLE(8), PRVGEOM, PRVMODE, DIHW, DIHT,	CONTRL	00002
1 DEFAULT	CONTRL	00003
LOGICAL PRVGEOM, PRVMODE, DIHW, DIHT, DEFAULT	CONTRL	00004
COMMON /PROBLM/ XMACH, NModes, NTSLOP, NKVALS, SMOOTH, NDEG, CRDFIT,	PROBLM	00002
1 EXAIC, SUBDV, PLYWOOD	PROBLM	00003
LOGICAL SMOOTH, CRDFIT, EXAIC, SUBDV, PLYWOOD	PROBLM	00004
COMMON /KVAL / IKVAL, XKVAL(20), XKS(20)	KVAL	00002
COMMON /GEOMTY/ COPLAN, NSUBDV, XSUBDV, NSUBD2, NSUBCN, NSURF,	GEOMTY	00002
1 B1, B1BETA, B1S, B1BTAS, WLAX, WLAZ, PSIW,	GEOMTY	00003
2 MYBW, MYBBW, MYBW, MYBBW, MYBSW, MYBSW, MYBBSW,	GEOMTY	00004
3 IXBW, XCENR	GEOMTY	00005
LOGICAL COPLAN	GEOMTY	00006
COMMON /GEOM2 / TLAX, TLAZ, PSIT, MXBT, MYBT, MYBBT, MXBST, MYBST,	GEOM2	00002
1 MYBBST, IXBT, IXBST, CARL	GEOM2	00003
COMMON / KERN / ERR, MXSKRN, IKERN, NPKRN, NSPATK, NROEA	KERN	00002
COMMON /FILES / NT5, NT6, INTAPE, INFSP, NPLAIC, NSPAIC, NOUTP,	FILES	00002
1 IOUFSP, MODESC, IVPSC, IGEOSC, IWTFC, IAICSC	FILES	00003
COMMON /IOCONT/ OPLAIC, OSPAIC, WTGEOM, WTGNAF, WTSI, WTL, PRBOX,	IOCONT	00002
1 PRPAIC, PRSAIC, PRMODS, PRCOEF, PRDW, PRSW, PRVP,	IOCONT	00003
2 PRBL, PRDCP, PRGNAF, PRGNAC, PRSL, FRLW, FRNW, FRCH	BCSFRB	00001
EQUIVALENCE (FRLW, FRDW)	IOCONT	00005
LOGICAL OPLAIC, OSPAIC, WTGEOM, WTGNAF, WTSI, WTL, PRBOX, PRPAIC,	IOCONT	00006
1 PRSAIC, PRMODS, PRCOEF, PRDW, PRSW, PRVP, PRBL, PRSL, PRGNAF,	IOCONT	00007
2 PRDCP, PRGNAC, FRLW, FRLW, FRNW, FRCH	BCSFRB	00002
COMMON /TAPEIO/ NFS, NMS, LS, NMR, ID(20), NID, ITYPE, LRS, LWS, M, N,	TAPEIO	00002
1 PARM(10), IRR	TAPEIO	00003
DIMENSION IARM(10)	TAPEIO	00004
EQUIVALENCE (PARM, IARM)	TAPEIO	00005
COMMON / MODES/ SYM, SYMT, MTYPEW, MTYPEL	MODCOM	00002
COMMON /ARRAYS/ KBXCDW, LBXCDW, LBOXC, LBXCDT, LBXCDT, KJALPH, LJALPH,	ARRAYS	00002
1 KALPHA, KKERNL, LKERL, KPNTRM, LPNTRM, KDEFSL, KELPHI,	ARRAYS	00003
2 LMODES, KPNTSD, LPNTSD, KSDW, LSDW, KPNTDW, LPNTDW,	ARRAYS	00004
3 KDW, LDW, KTVP, LTVP	ARRAYS	00005
COMMON /SAMPLW/ ISMPLW, ICHORD(10), IBOXF(10), IBOXL(10), ZLOC(10)	SAMPLW	00002
COMMON /MUAICS/ YBAR, EL, MUAIC(2,50), NROWS, SURF,	MUAICS	00002
1 YBARL, ELL, MUAICL(2,50), NROWSL, SURFL, PSIDIF	MUAICS	00003
LOGICAL SURF, SURFL	MUAICS	00004
COMMON /EDGES / FEXLOC(250), TEXLOC(250), JDIAG	EDGES	00002

COMMON /PLANXY/ NMLE,NMTE,NTLE,NTTE, XMLE(10),YMLE(10),	PLANXY 00002
1 XWTE(10),YWTE(10), XYLE(10),YCLE(10),	PLANXY 00003
2 XTTE(10),YTTE(10)	PLANXY 00004
LOGICAL MXWRIT,RANDOU, MXREAD,RANDIN	GEOMBX 00036
COMMON /CHECKPR/ DFCPR, GEOCFR, MODCFR, AICCFR, NMSCFR, SMCPR, GAFCFR	CHECKPR 00002
LOGICAL DFCPR, GEOCFR, MODCFR, AICCFR, NMSCFR, SMCPR, GAFCFR	CHECKPR 00003
EQUIVALENCE (CHECKPR, GEOCFR)	GEOMBX 00039
LOGICAL CHECKPR	GEOMBX 00040
C	GEOMBX 00042
C	GEOMBX 00051
DIMENSION IBOXW(150,8), IBOXT(90,8),	GEOMBX 00052
1 IWAKE(160), ICODE(160)	GEOMBX 00053
DIMENSION KPTW(50), KPTTT(50), KPTLWT(50), KPTRWT(50)	GEOMBX 00054
DIMENSION ALPHA(200), IJALPH(200)	GEOMBX 00055
DIMENSION KPT(4,50)	GEOMBX 00056
EQUIVALENCE (KPT, ALPHA)	GEOMBX 00057
DATA MXWRIT,RANDOU,MXREAD,RANDIN / 4*.F. /	FTNXI 00008
DATA NBWRD /20/	FTNXI 00009
DATA INIT,XINIT / 3776700. 3765432177777777777B/	GEOMBX 00058
DATA EPS / 1.0E-4 /	GEOMBX 00059
C	FTNXI 00010
C	FTNXI 00011
NAMLIST PARAMETERS FOR CARDS TO BE READ IN THIS SECTION	FTNXI 00012
NAMLIST /CARDF/ WLAX,WLAZ,PSIW, TLAX,TLAZ, PSIT, CHECKPR	FTNXI 00013
1 /CARDG/ NCHRDS, XCENR, XEDGE, ICHORD,IBOXF,IBOXL,ZLOC	FTNXI 00014
CARDH NMLE, NMTE, NTLE, NTTE (415)	FTNXI 00015
CARDS I TO L (6E10.0)	FTNXI 00016
C	FTNXI 00017
9001 FORMAT(415)	FTNXI 00018
9002 FORMAT(6E10.0)	FTNXI 00019
C	GEOMBX 00060
C	GEOMBX 00061
LSCHDS = LBOXC * 20	GEOMBX 00062
XSUBDV = NSUBDV	GEOMBX 00063
NSUBD2 = NSUBDV/2	GEOMBX 00064
NSUBCN = NSUBD2 + 1	GEOMBX 00065
HALFBX = XSUBDV/2.0	GEOMBX 00066
C	GEOMBX 00067
C	GEOMBX 00068
IS PREVIOUS GEOMETRY TO BE USED -	GEOMBX 00069
IF (.NOT. PRVGEOM) GO TO 15	GEOMBX 00070
C	GEOMBX 00071
YES. HAS THE MACH NUMBER CHANGED -	GEOMBX 00072
IF (XMACH.EQ. OMACH) GO TO 2000	GEOMBX 00073
C	GEOMBX 00074
YES. SKIP THE GEOMETRY READS, BUT READ THE REST FOR THE	GEOMBX 00075
C	GEOMBX 00076
NEW BOX PATTERN	GEOMBX 00077
B10 = 31	GEOMBX 00078
XCENR = XCENR	GEOMBX 00079
GO TO 272	GEOMBX 00080
C	GEOMBX 00081
C	GEOMBX 00082
READ CARDS F AND G	GEOMBX 00083
15 CONTINUE	GEOMBX 00084
WLAX = 0.	GEOMBX 00085
WLAZ = 0.	GEOMBX 00086
PSIW = 0.	
TLAX = 0.	
TLAZ = 0.	
PSIT = 0.	
READ (NT5,CARDF)	
WRITE (NT5,6010) WLAX,WLAZ,PSIW	

IF (NSURF .EQ. 2) WRITE (NT6,6012) TLAX,TLAZ,PSIT	GEOMBX 00087
DEGREE = .01745329251943	GEOMBX 00088
C CONVERT DEGREES TO RADIANS	GEOMBX 00089
IVAL = 4HPSIW	GEOMBX 00090
IF (PSIW .GT. 45. .OR. PSIW .LT. -45.) GO TO 8030	GEOMBX 00091
PSIW = PSIW * DEGREE	GEOMBX 00092
IF (NSURF .EQ. 1) GO TO 30	GEOMBX 00093
IVAL = 4HPSIT	GEOMBX 00094
C SPECIAL CHECK FOR VERTICAL TAIL	GEOMBX 00095
IF (PSIT .EQ. 90. .AND. SYM .LE. 0) GO TO 25	GEOMBX 00096
IF (PSIT .GT. 45. .OR. PSIT .LT. -45.) GO TO 8030	GEOMBX 00097
C SYMT = SYMMETRY INDICATOR FOR THE TAIL, IDENTICAL TO	GEOMBX 00098
C THE WING EXCEPT FOR A VERTICAL TAIL	GEOMBX 00099
SYMT = SYM	GEOMBX 00100
GO TO 28	GEOMBX 00101
25 CONTINUE	GEOMBX 00102
SYMT = 0.	GEOMBX 00103
28 CONTINUE	GEOMBX 00104
PSIT = PSIT * DEGREE	GEOMBX 00105
C	GEOMBX 00106
30 CONTINUE	GEOMBX 00107
NCHRS = INIT	GEOMBX 00108
XCENR = XINIT	GEOMBX 00109
XEDGE = XINIT	GEOMBX 00110
DO 50 I=1,10	GEOMBX 00111
ICHORD(I) = INIT	GEOMBX 00112
IBORF(I) = INIT	GEOMBX 00113
IBOXL(I) = INIT	GEOMBX 00114
50 ZLOC(I) = XINIT	GEOMBX 00115
READ(NT5,CARDG)	GEOMBX 00116
C	GEOMBX 00117
C CHECK AND PRINT PARAMETERS READ	GEOMBX 00118
WRITE (NT6,6015) NCHRS,XCENR, XEDGE	GEOMBX 00119
C	GEOMBX 00120
IVAL = 8HNCHRS	GEOMBX 00121
IF (NCHRS .EQ. INIT) GO TO 8010	GEOMBX 00122
IF (NCHRS .LE. 0 .OR. NCHRS .GE. LSCDWS/NSUBDV) GO TO 8015	GEOMBX 00123
MYBW = NCHRS	GEOMBX 00124
IVAL = 8HXCENR	GEOMBX 00125
IF (XCENR .EQ. XINIT) GO TO 120	GEOMBX 00126
C USE XCENR DIRECTLY, IGNORE XEDGE	GEOMBX 00127
IF (XEDGE .NE. XINIT) WRITE (NT6,9010)	GEOMBX 00128
GO TO 125	GEOMBX 00129
C GET XCENR FROM XEDGE	GEOMBX 00130
120 CONTINUE	GEOMBX 00131
IF (XEDGE .EQ. XINIT) GO TO 8020	GEOMBX 00132
125 CONTINUE	GEOMBX 00133
IF (ISMPLW .EQ. 0) GO TO 200	GEOMBX 00134
IF (NSURF .EQ. 2) GO TO 170	GEOMBX 00135
DO 150 I = 1,ISMPLW	GEOMBX 00136
IF (ICHORD(I) .GT. MYBW .OR. ICHORD(I) .LE. 0) GO TO 130	GEOMBX 00137
IF (IBORF(I) .EQ. INIT .OR. IBOXL(I) .EQ. INIT) GO TO 140	GEOMBX 00138
IF (IBORF(I) .LT. 1 .GT. IBOXL(I) .GT. LBXCDW/NSUBDV) GO TO 130	GEOMBX 00139
IF (IBORF(I) .LE. IBOXL(I)) GO TO 140	GEOMBX 00140
130 WRITE (NT6,9020) I	GEOMBX 00141
ISMPLW = I - 1	GEOMBX 00142
GO TO 160	GEOMBX 00143

140 CONTINUE	GEOMBX 00144
IF (ZLOC(I) .EQ. XINIT) ZLOC(I) = 0.	GEOMBX 00145
150 CONTINUE	GEOMBX 00146
160 CONTINUE	GEOMBX 00147
WRITE (NT6,6017) ISMPLW, (ICHORD(I), IBOXF(I), IBOXL(I), ZLOC(I),	GEOMBX 00148
1 I = 1,ISMPLW)	GEOMBX 00149
GO TO 200	GEOMBX 00150
C SAMPLING OF SONNASHES ILLEGAL IF TAIL DEFINED	GEOMBX 00151
170 CONTINUE	GEOMBX 00152
WRITE (NT6,6030) ISMPLW	GEOMBX 00153
ISMPLW = 0	GEOMBX 00154
200 CONTINUE	GEOMBX 00155
C	GEOMBX 00156
C OBTAIN THE LEADING AND TRAILING EDGE VALUES	GEOMBX 00157
C CARD INPUT OF PLANKFORMS IS REQUIRED	GEOMBX 00158
READ (NT5,5001) NMLE,NWTE,NTLE,NTTE	GEOMBX 00159
210 WRITE(NT6,6021) NMLE,NWTE	GEOMBX 00160
GO TO (214,212),NSURF	GEOMBX 00161
212 WRITE(NT6,6022) NTLE,NTTE	GEOMBX 00162
214 IVAL = 4HNMLE	GEOMBX 00163
IF (NMLE .LT. 2 .OR. NMLE .GT. 10) GO TO 8030	GEOMBX 00164
IVAL = 4HNWTE	GEOMBX 00165
IF (NWTE .LT. 2 .OR. NWTE .GT. 10) GO TO 8030	GEOMBX 00166
IF (NSURF .EQ. 1) GO TO 220	GEOMBX 00167
IVAL = 4HNTLE	GEOMBX 00168
IF (NTLE .LT. 2 .OR. NTLE .GT. 10) GO TO 8030	GEOMBX 00169
IVAL = 4HNTTE	GEOMBX 00170
IF (NTTE .LT. 2 .OR. NTTE .GT. 10) GO TO 8030	GEOMBX 00171
220 CONTINUE	GEOMBX 00172
C	GEOMBX 00173
C CARDS I AND J - WING DEFINITION POINTS	GEOMBX 00174
WRITE (NT6,6029)	GEOMBX 00175
IVAL = 9HWING L.E.	GEOMBX 00176
READ (NT5,5002) (XWLE(I),YWLE(I),I=1,NMLE)	GEOMBX 00177
WRITE (NT6,6030) IVAL, (XWLE(I),YWLE(I),I=1,NMLE)	GEOMBX 00178
CALL EDGCHK(XWLE,YWLE,NMLE,1,IRR)	GEOMBX 00179
IF (IRR .NE. 0) GO TO 8050	GEOMBX 00180
IVAL = 9HWING T.E.	GEOMBX 00181
READ (NT5,5002) (XWTE(I),YWTE(I),I=1,NWTE)	GEOMBX 00182
WRITE (NT6,6030) IVAL, (XWTE(I),YWTE(I),I=1,NWTE)	GEOMBX 00183
CALL EDGCHK(XWTE,YWTE,NWTE,2,IRR)	GEOMBX 00184
IF (IRR .NE. 0) GO TO 8050	GEOMBX 00185
IF (NSURF .EQ. 1) GO TO 270	GEOMBX 00186
C	GEOMBX 00187
C CARDS K AND L - TAIL DEFINITION POINTS	GEOMBX 00188
IVAL = 9HTAIL L.E.	GEOMBX 00189
READ (NT5,5002) (XTLE(I),YTLE(I),I=1,NTLE)	GEOMBX 00190
WRITE (NT6,6030) IVAL, (XTLE(I),YTLE(I),I=1,NTLE)	GEOMBX 00191
CALL EDGCHK(XTLE,YTLE,NTLE,1,IRR)	GEOMBX 00192
IF (IRR .NE. 0) GO TO 8050	GEOMBX 00193
IVAL = 9HTAIL T.E.	GEOMBX 00194
READ (NT5,5002) (XTTE(I),YTTE(I),I=1,NTTE)	GEOMBX 00195
WRITE (NT6,6030) IVAL, (XTTE(I),YTTE(I),I=1,NTTE)	GEOMBX 00196
CALL EDGCHK(XTTE,YTTE,NTTE,2,IRR)	GEOMBX 00197
IF (IRR .NE. 0) GO TO 8050	GEOMBX 00198
GO TO 270	GEOMBX 00199
C	GEOMBX 00200

C	PLANFORM DEFINITIONS TO BE READ FROM TAPE	GEOMBX	00201
	230 CONTINUE	GEOMBX	00202
C		GEOMBX	00203
	270 CONTINUE	GEOMBX	00204
	B1BETA = YWLE(NMLE)/MYBW	GEOMBX	00205
	272 CONTINUE	GEOMBX	00206
	B1 = B1BETA * SQRT(XMACH*XMACH-1.0)	GEOMBX	00207
	B1BTA2 = B1BETA* 0.5	GEOMBX	00208
	B12 = B1 * 0.5	GEOMBX	00209
	IF (NSUBDV .NE. 1) GO TO 275	GEOMBX	00210
	B1S = B1	GEOMBX	00211
	B1BTAS = B1BETA	GEOMBX	00212
	GO TO 280	GEOMBX	00213
	275 B1S = B1/XSUBDV	GEOMBX	00214
	B1BTAS = B1BETA/XSUBDV	GEOMBX	00215
	280 CONTINUE	GEOMBX	00216
	WRITE (NT6,6040) B1,B1BETA	GEOMBX	00217
C		GEOMBX	00218
C	SET THE XVAL ARRAY IF XKS WAS INPUT	GEOMBX	00219
C		GEOMBX	00220
	IF(XKS(1).EQ.-1.0) GO TO 295	GEOMBX	00221
	DO 290 I=1,NKVALS	GEOMBX	00222
	XVAL(I) = XKS(I) * (B1/YWLE(NMLE))	GEOMBX	00223
	290 CONTINUE	GEOMBX	00224
	295 CONTINUE	GEOMBX	00225
C		GEOMBX	00226
C	DETERMINE THE GLOBAL COORDINATE LOCATION OF THE FIRST UN-	GEOMBX	00227
C	SUBDIVIDED PLANFORM BOX CENTER, XCENTR	GEOMBX	00228
	IVAL = 6HXCENTR	GEOMBX	00229
	IF (XCENTR .EQ. XINIT) XCENTR = XEDGE + B12	GEOMBX	00230
	XEDGEW = XWLE(1) + (XWLE(2)-XWLE(1)) * B1BTA2 / YWLE(2)	GEOMBX	00231
	IF (PRVGCOM) XEDGEW = B10* XWLE(1) - B10 + XCNTRO +	GEOMBX	00232
	1 B10 *(XWLE(2) - XWLE(1)) * .5 / (YWLE(2) - .5)	GEOMBX	00233
	IF (XCENTR-XEDGEW) 310,330,320	GEOMBX	00234
	310 DO 315 I = 1,51	GEOMBX	00235
	XCENTR = XCENTR + B1	GEOMBX	00236
	IF (XCENTR .GE. XEDGEW) GO TO 330	GEOMBX	00237
	315 CONTINUE	GEOMBX	00238
	GO TO 8060	GEOMBX	00239
	320 DO 325 I = 1,51	GEOMBX	00240
	IF (XCENTR-B1 .LT. XEDGEW) GO TO 330	GEOMBX	00241
	XCENTR = XCENTR - B1	GEOMBX	00242
	325 CONTINUE	GEOMBX	00243
	GO TO 8060	GEOMBX	00244
	330 CONTINUE	GEOMBX	00245
C	IS PREVIOUS GEOMETRY BEGIN USED -	GEOMBX	00246
	IF (.NOT. PRVGCOM) GO TO 355	GEOMBX	00247
C		GEOMBX	00248
C	YES. CONVERT X-COORDINATE VALUES TO NEW BOX LENGTH	GEOMBX	00249
	PSIDIF = PSIT - PSIW	GEOMBX	00250
	SLIDE = -B10 + XCNTRO - XCENTR	GEOMBX	00251
	DO 335 I = 1,NMLE	GEOMBX	00252
	335 XWLE(I) = (B10*XWLE(I) + SLIDE)/B1 + 1.0	GEOMBX	00253
	DO 340 I = 1,NMTE	GEOMBX	00254
	340 XWTE(I) = (B10*XWTE(I) + SLIDE)/B1 + 1.0	GEOMBX	00255
	IF (NSURF .EQ. 1) GO TO 390	GEOMBX	00256
	DO 345 I = 1,NTLE	GEOMBX	00257

345	XTLE(I) = (B10*XTLE(I) + SLIDE)/B1 + 1.0	GEOMBX	00258
	DO 350 I = 1,NTLE	GEOMBX	00259
350	XTTE(I) = (B10*XTTE(I) + SLIDE)/B1 + 1.0	GEOMBX	00260
	GO TO 390	GEOMBX	00261
C		GEOMBX	00262
C	CONVERT GEOMETRIC INFORMATION TO THE NON-DIMENSIONAL	GEOMBX	00263
C	NC, MC, LC COORDINATE SYSTEM	GEOMBX	00264
355	CONTINUE	GEOMBX	00265
	DO 360 I = 1,NMLE	GEOMBX	00266
	XMLE(I) = (XMLE(I)-XCENR)/B1 + 1.0	GEOMBX	00267
360	YMLE(I) = YMLE(I)/B1BETA + 0.5	GEOMBX	00268
	DO 365 I = 1,NMTE	GEOMBX	00269
	XMTE(I) = (XMTE(I)-XCENR)/B1 + 1.0	GEOMBX	00270
365	YMTE(I) = YMTE(I)/B1BETA + 0.5	GEOMBX	00271
	GO TO (370,375),NSURF	GEOMBX	00272
370	CAPL = 0.	GEOMBX	00273
	PSIT = 0.	GEOMBX	00274
	PSIDIF = -PSIW	GEOMBX	00275
	MYBT = 0	GEOMBX	00276
	MYBST = 0	GEOMBX	00277
	MYBST = 0	GEOMBX	00278
	MYBBST = 0	GEOMBX	00279
	IF (ISMPLW .EQ. 0) GO TO 390	GEOMBX	00280
C	TRANSFORM ZLOC FOR THE SAMPLE WASH CHORDS TO A NON-DIMENSIONAL	GEOMBX	00281
C	UNROTATED LC COORDINATE HAVING ITS ZERO ON THE WING CENTER	GEOMBX	00282
C	LINE	GEOMBX	00283
	DO 372 I = 1,ISMPLW	GEOMBX	00284
	ZLOC(I) = (ZLOC(I) - WLAZ) /B1BETA	GEOMBX	00285
372	CONTINUE	GEOMBX	00286
	GO TO 390	GEOMBX	00287
375	XDIFF = WLAZ + XCENR - TLAX	GEOMBX	00288
	DO 380 I = 1,NTLE	GEOMBX	00289
	XTLE(I) = (XTLE(I)-XDIFF)/B1 + 1.0	GEOMBX	00290
380	YTLE(I) = YTLE(I)/B1BETA + 0.5	GEOMBX	00291
	DO 385 I = 1,NTTE	GEOMBX	00292
	XTTE(I) = (XTTE(I)-XDIFF)/B1 + 1.0	GEOMBX	00293
385	YTTE(I) = YTTE(I)/B1BETA + 0.5	GEOMBX	00294
	CAPL = (TLAZ-WLAZ)/B1BETA	GEOMBX	00295
	PSIDIF = PSIT - PSIW	GEOMBX	00296
C		GEOMBX	00297
C	CHECK FOR TAIL CROSSING WING	GEOMBX	00298
	IF (PSIDIF) 386,389,387	GEOMBX	00299
386	IF (CAPL .LE. 0) GO TO 389	GEOMBX	00300
	GO TO 388	GEOMBX	00301
387	IF (CAPL .GE. 0) GO TO 389	GEOMBX	00302
388	YCROSS = CAPL/(SIN(PSIW)-SIN(PSIT)) + .5	GEOMBX	00303
	IF (YMLE(NMLE)*COS(PSIW) .LT. YCROSS) GO TO 390	GEOMBX	00304
	IF (YTLE(NMTE)*COS(PSIT) .GE. YCROSS) GO TO 8080	GEOMBX	00305
	GO TO 390	GEOMBX	00306
389	YCROSS = .5	GEOMBX	00307
C		GEOMBX	00308
C	ZERO OUT THE BOX CODE ARRAYS	GEOMBX	00309
390	CONTINUE	GEOMBX	00310
	DO 430 J = 1,LBOXC	GEOMBX	00311
	DO 420 I = 1,LBOXCW	GEOMBX	00312
420	IBOXV(I,J) = 0	GEOMBX	00313
	DO 430 I = 1,LBOXCT	GEOMBX	00314

430 IBOKT(I,J) = 0	GEOMBX 00315
C	GEOMBX 00316
C GET THE (SUBDIVIDED) BOX CODE ARRAY FOR THE ON-PLANFORM WING	GEOMBX 00317
C BOXES	GEOMBX 00318
IXBW = 0	GEOMBX 00319
IXBT = 0	GEOMBX 00320
IXBST = 0	GEOMBX 00321
IXBT = 0	GEOMBX 00322
C	GEOMBX 00323
CALL BXCDPF(XMLE,YMLE,NMLE,XWTE,YWTE,NWTE, LBXCDW,IBOXW)	GEOMBX 00324
C RETURNS- IBOXW, ONES FOR ON PLANFORM BOXES	GEOMBX 00325
C IXBW = LOCATION OF FIRST UNSUBDIVIDED BOX CENTER	GEOMBX 00326
C MYBSW = NUMBER OF SUBDIVIDED ROWS ON THE WING	GEOMBX 00327
C MYBSW = NUMBER OF SUBDIVIDED CHORDS ON THE WING	GEOMBX 00328
C MXBW = NUMBER OF UNSUBDIVIDED ROWS	GEOMBX 00329
C MYBW = NUMBER OF UNSUBDIVIDED CHORDS	GEOMBX 00330
C FEXLOC = ARRAY OF LEADING EDGE LOCATIONS	GEOMBX 00331
C TEXLOC = ARRAY OF TRAILING EDGE LOCATIONS	GEOMBX 00332
C	GEOMBX 00333
MYBSW = MYBSW	GEOMBX 00334
MXBSW = MXBW	GEOMBX 00335
MYBSW = MYBSW	GEOMBX 00336
IF (.NOT. CHECKPR) GO TO 440	GEOMBX 00337
CALL PRINTBC(IBOXW,LBXCDW, 1, MXBSW, MYBSW, .T.)	GEOMBX 00338
WRITE (NT6,7040) (FEXLOC(I), I = 1,MYBSW)	GEOMBX 00339
WRITE (NT6,7045) (TEXLOC(I), I = 1,MYBSW)	GEOMBX 00340
440 CONTINUE	GEOMBX 00341
C	GEOMBX 00342
C SEARCH THE WING FOR THE FORWARD MOST DIAGONAL INTERSECTING	GEOMBX 00343
C AN ON-PLANFORM BOX. THIS DEFINES THE LIMIT FOR ANY TIP	GEOMBX 00344
C DIAPHRAGM.	GEOMBX 00345
C JDIAG = THE J-LOCATION (SUBDIVIDED) OF THE DIAGONAL AT	GEOMBX 00346
C THE FIRST ROW OF THE PATTERN.	GEOMBX 00347
JDIAG = 1	GEOMBX 00348
PREV = 0.	GEOMBX 00349
DO 530 J = 2,MYBSW	GEOMBX 00350
PREV = PREV + 1.0	GEOMBX 00351
IF (FEXLOC(J) .GT. PREV) GO TO 530	GEOMBX 00352
PREV = FLOAT(IFIX(FEXLOC(J)))	GEOMBX 00353
JDIAG = J - PREV	GEOMBX 00354
530 CONTINUE	GEOMBX 00355
C	GEOMBX 00356
C INITIALIZE THE IWAKE ARRAY	GEOMBX 00357
DO 540 J = 1,MYBSW	GEOMBX 00358
IWAKE(J) = TEXLOC(J)	GEOMBX 00359
540 CONTINUE	GEOMBX 00360
IF (MYBSW .EQ. LSCHDS) GO TO 548	GEOMBX 00361
MYBSW = MYBSW + 1	GEOMBX 00362
DO 544 J = MYBSW,LSCHDS	GEOMBX 00363
544 IWAKE(J) = 0	GEOMBX 00364
548 CONTINUE	GEOMBX 00365
IF (NSURF .NE. 2) GO TO 705	GEOMBX 00366
C THERE ARE 2 SURFACES. DETERMINE THE FIRST PLANFORM BOX OF THE	GEOMBX 00367
C SECOND SURFACE	GEOMBX 00368
YMIN = .5*(1.0 + 1.0/XSUBDV)	GEOMBX 00369
DELE = (XTLE(2)-XTLE(1)) / (YCLE(2)-YCLE(1))	GEOMBX 00370
XMINS = XTLE(1) + (YMIN-YCLE(1)) * DELE	GEOMBX 00371

IXBST = XSUBDV*(XMIN5-1.0) + IXBW + 1	GEOMBX 00372
IF (AINT(XMIN5) .EQ. XMIN5) IXBST = IXBST - 1	GEOMBX 00373
XMIN = XTLE(1) + (1.0 - YTLE(1)) * DELE	GEOMBX 00374
IXBT = XMIN	GEOMBX 00375
IF (FLOAT(IXBT) .EQ. XMIN) IXBT = IXBT - 1	GEOMBX 00376
IXBT = NSUBDV * IXBT + IXBW	GEOMBX 00377
C IXBST = LOCATION OF FIRST SUBDIVIDED TAIL BOX	GEOMBX 00378
C IXBT = LOCATION OF FIRST UNSUBDIVIDED TAIL BOX CENTER	GEOMBX 00379
ISUBT = 2 - IXBST	GEOMBX 00380
C ISUBT = THE SUBSCRIPT FOR ARRAY IBOXT WHICH WILL KEEP TAIL	GEOMBX 00381
C ROWS WITHIN THE BOUNDS OF IBOXT	GEOMBX 00382
IF (CAPL .NE. 0) GO TO 510	GEOMBX 00383
IF (PSIDIF .EQ. 0) GO TO 700	GEOMBX 00384
C	GEOMBX 00385
C THE TWO SURFACES ARE NOT COPLANAR	GEOMBX 00386
510 CONTINUE	GEOMBX 00387
COPLAN = .F.	GEOMBX 00388
C DETERMINE THE BOX CODES FOR THE SECOND PLANFORM	GEOMBX 00389
CALL BXCDPF(XTLE,YTLE,NTLE, XTTE,YTTE,NTTE, LBXCDT,IBOXT(ISUBT,1))	GEOMBX 00390
C RETURNS - IBOXT, ONES FOR ON-PLANFORM BOXES	GEOMBX 00391
C MXBST = NUMBER OF SUBDIVIDED ROWS TO END OF TAIL	GEOMBX 00392
C MYBST = NUMBER OF SUBDIVIDED CHORDS ON TAIL	GEOMBX 00393
C MXBT = NUMBER OF UNSUBDIVIDED ROWS, BOTH PLANFORMS	GEOMBX 00394
C MYBT = NUMBER OF UNSUBDIVIDED CHORDS ON TAIL	GEOMBX 00395
C FEXLOC = LEADING EDGE LOCATIONS, BOTH PLANFORMS	GEOMBX 00396
C TEXLOC = TRAILING EDGE LOCATIONS	GEOMBX 00397
C	GEOMBX 00398
C GET DIAPHRAGM VALUES FOR THE TAIL	GEOMBX 00399
MYBBST = MYBST	GEOMBX 00400
IF (.NOT. CHECKFR) GO TO 515	GEOMBX 00401
CALL PRNTBC(IBOXT(ISUBT,1),LBXCDT,IXBST, MXBST,MYBST,.T.)	GEOMBX 00402
II = MYBSW + 1	GEOMBX 00403
III = MYBSW + MYBST	GEOMBX 00404
WRITE (NT6,7040) (FEXLOC(I), I = II,III)	GEOMBX 00405
WRITE (NT6,7045) (TEXLOC(I), I = II,III)	GEOMBX 00406
515 CONTINUE	GEOMBX 00407
IWK = 0	GEOMBX 00408
CALL BXCDI (IWK, LBXCDT,LSCHDS, IBOXT(ISUBT,1))	GEOMBX 00409
C RETURNS - IBOXT, CODES 2 AND 3 ADDED FOR DIAPHRAGM AND WAKE	GEOMBX 00410
C MYBBST = NUMBER OF SUBDIVIDED CHORDS, INCLUDING	GEOMBX 00411
C DIAPHRAGM, FOR TAIL	GEOMBX 00412
C MYBBT = NUMBER OF UNSUBDIVIDED CHORDS	GEOMBX 00413
C	GEOMBX 00414
IF (.NOT. (PRBOX .OR. CHECKFR)) GO TO 520	GEOMBX 00415
CALL PRNTBC(IBOXT(ISUBT,1),LBXCDT, IXBST, MXBST,MYBBST, .T.)	GEOMBX 00416
IF (NSUBDV .EQ. 1) GO TO 520	GEOMBX 00417
IFR = (IXBT - IXBW)/NSUBDV + 1	GEOMBX 00418
CALL PRNTBC(IBOXT(ISUBT,1),LBXCDT, IFR, MXBT, MYBBT, .F.)	GEOMBX 00419
520 CONTINUE	GEOMBX 00420
C	GEOMBX 00421
C THE FOLLOWING LOOP DETERMINES THE LOCUS OF MAXIMUM AFTWARD	GEOMBX 00422
C PROJECTIONS OF THE INTERSECTIONS OF THE TAIL MACH CONES	GEOMBX 00423
C WITH THE WING PLANE (EXTENDED). MACH CONES FOR UNSUBDIVIDED	GEOMBX 00424
C TAIL CHORDS ARE USED, BUT ALL ARITHMETIC IS IN THE SUBDIVIDED	GEOMBX 00425
C COORDINATE SYSTEM.	GEOMBX 00426
C	GEOMBX 00427
C LOOP ON TAIL CHORDS	GEOMBX 00428

C	CAPLL = CAPL	GEOMBX	00429
	(DO 600 JT = NSUBCN, MYBST, NSUBDV)	FTNXI	00020
	JT = NSUBCN	FTNXI	00021
325	CONTINUE	FTNXI	00022
	YCT = JT - .5	GEOMBX	00431
C	Y-OFFSET OF RECEIVING CHORD FROM CENTER-LINE, TAIL PLANE	GEOMBX	00432
C	GET ICT, THE I-LOCATION OF AFTMOST RECEIVING BOX ON TAIL CHRD	GEOMBX	00433
C	IS THE TAIL CHORD ON-PLANFORM OR DIAPHRAGM -	GEOMBX	00434
	IF (JT .GT. MYBST) GO TO 550	GEOMBX	00435
	JJ = JT + MYBSW	GEOMBX	00436
	ICT = TEXLOC(JJ) + EPS - AMOD(TEXLOC(JJ)-IXBW, XSUBDV)	GEOMBX	00437
	GO TO 555	GEOMBX	00438
550	CONTINUE	GEOMBX	00439
	ICT = IXBT - NSUBDV	GEOMBX	00440
555	CONTINUE	GEOMBX	00441
	ICTP1 = ICT + NSUBDV	GEOMBX	00442
	IF (ICTP1 .GT. MYBST) GO TO 570	GEOMBX	00443
C	CHECK FOR WAKE DIAPHRAGM AFT OF TAIL CHORD	GEOMBX	00444
	CALL DCDER(IBOXT(ISUBT,1),LBXCDT, ICTP1,JT, MYBST,JT,	GEOMBX	00445
	1 .T., ICODE)	GEOMBX	00446
	II = 1	GEOMBX	00447
	DO 580 I = ICTP1,MYBST,NSUBDV	GEOMBX	00448
	IF (ICODE(II).EQ. 0) GO TO 570	GEOMBX	00449
	ICT = I	GEOMBX	00450
	II = II + 1	GEOMBX	00451
580	CONTINUE	GEOMBX	00452
570	CONTINUE	GEOMBX	00453
C	ICT = X-LOCATION OF AFT-MOST TAIL BOX ON THE CHORD	GEOMBX	00454
	EL = COS(PSIW)*CAPL*XSUBDV + SIN(PSIDIF)*YCT	GEOMBX	00455
C	EL = PERPENDICULAR DISTANCE FROM RECEIVING CHORD TO RIGHT	GEOMBX	00456
C	WING PLANE, POSITIVE DOWNWARD.	GEOMBX	00457
C		GEOMBX	00458
C	ENTRY INTO THE LOOP FOR WASH SAMPLING CHORDS, FROM 705*	GEOMBX	00459
590	CONTINUE	GEOMBX	00460
C	START OF LOOP ON WING CHORDS, ENDING AT 650	GEOMBX	00461
	JW = NSUBCN	GEOMBX	00462
600	CONTINUE	GEOMBX	00463
	YJW = JW - .5	GEOMBX	00464
C	YJW = Y-OFFSET OF SENDING CHORD FROM CENTER LINE,	GEOMBX	00465
C	WING PLANE	GEOMBX	00466
	YMUBAR = -YJW + COS(PSIDIF)*YCT + SIN(PSIW)*CAPLL*XSUBDV	GEOMBX	00467
C	YMUBAR = Y-DISTANCE BETWEEN CHORD CENTERS, SENDING (WING)	GEOMBX	00468
C	PLANE	GEOMBX	00469
	IF (ABS(YMUBAR) .LE. HALFBX) GO TO 630	GEOMBX	00470
	IF (YMUBAR .LT. -HALFBX) YMUBAR = YMUBAR + HALFBX	GEOMBX	00471
	IF (YMUBAR .GT. HALFBX) YMUBAR = YMUBAR - HALFBX	GEOMBX	00472
C	YMUBAR = Y-DISTANCE TO NEAREST BOX EDGE, WING PLANE	GEOMBX	00473
	XNUBAR = SORT(YMUBAR**2 + (EL*XSUBDV)**2)	GEOMBX	00474
C	XNUBAR = DISTANCE FORWARD FROM RECEIVING CENTER TO NEAR-	GEOMBX	00475
C	EST PORTION OF SENDING CHORD	GEOMBX	00476
	GO TO 635	GEOMBX	00477
630	CONTINUE	GEOMBX	00478
	XNUBAR = ABS(EL)*XSUBDV	GEOMBX	00479
635	CONTINUE	GEOMBX	00480
	XNUBAR = XNUBAR + HALFBX	GEOMBX	00481
	INTRST = ICT - IFIX(XNUBAR+EPS - AMOD(XNUBAR, XSUBDV))	GEOMBX	00482
	IF (JW .GT. MYBSW) GO TO 640	GEOMBX	00483

IWAKE(JW) = MAXD(IWAKE(JW),INTRST)	GEOMBX 00484
GO TO 650	GEOMBX 00485
640 CONTINUE	GEOMBX 00486
IF (INTRST .LE. JW-JDIAG) GO TO 660	GEOMBX 00487
MYBBSW = JW	GEOMBX 00488
IWAKE(JW) = INTRST	GEOMBX 00489
650 CONTINUE	GEOMBX 00490
IF (NSUBD2 .EQ. 0) GO TO 657	GEOMBX 00491
DO 655 I = 1,NSUBD2	GEOMBX 00492
IWAKE(JW-I) = IWAKE(JW) - I	GEOMBX 00493
IWAKE(JW+I) = IWAKE(JW) - I	GEOMBX 00494
655 CONTINUE	GEOMBX 00495
657 CONTINUE	GEOMBX 00496
JW = JW + NSUBDV	GEOMBX 00497
GO TO 600	GEOMBX 00498
C END OF LOOP ON WING CHORDS	GEOMBX 00499
C	GEOMBX 00500
680 CONTINUE	GEOMBX 00501
IF (ISMPLW .NE. 0) GO TO 706	GEOMBX 00502
680 CONTINUE	GEOMBX 00503
JT = JT + NSUBDV	FTNX1 00023
IF (JT .LE. MYBBST) GO TO 525	FTNX1 00024
C END OF LOOP ON TAIL CHORDS, FROM 548*	GEOMBX 00504
C	GEOMBX 00505
685 CONTINUE	GEOMBX 00506
MXBBSW = MXBSW	GEOMBX 00507
DO 690 JW = NSUBCN,MYBBSW,NSUBDV	GEOMBX 00508
MXBBSW = MAXD(MXBBSW,IWAKE(JW))	GEOMBX 00509
690 CONTINUE	GEOMBX 00510
MXBBW = MXBBSW	GEOMBX 00511
IF (NSUBDV .GT. 1) MXBBW = (MXBBW-IXBW)/NSUBDV + 1	GEOMBX 00512
IF (CHECKPR) WRITE(NT6,7010) (IWAKE(I),I=1,MYBBSW)	GEOMBX 00513
GO TO 720	GEOMBX 00514
C	GEOMBX 00515
C THE TWO SURFACES ARE COPLANAR. ENTER THE SECOND PLANFORM	GEOMBX 00516
C INTO THE SAME BOX ARRAY	GEOMBX 00517
700 CONTINUE	GEOMBX 00518
COPLAN = .T.	GEOMBX 00519
CALL BXCDPF(XTLE,YTLE,NTLE, XTTE,YTTE,NTTE, LBXCDW,IBOKW)	GEOMBX 00520
MYBBSW = MYBST	GEOMBX 00521
IF (.NOT. CHECKPR) GO TO 720	GEOMBX 00522
CALL PRINTBC(IBOKW,LBXCDW, IXBST, MXBST,MYBST,.T.)	GEOMBX 00523
II = MYBSW + 1	GEOMBX 00524
III = MYBSW + MYBST	GEOMBX 00525
WRITE (NT6,7040) (FEXLOC(I), I = II,III)	GEOMBX 00526
WRITE (NT6,7045) (TEXLOC(I), I = II,III)	GEOMBX 00527
GO TO 720	GEOMBX 00528
C NO TAIL IS DEFINED. IS DOWNWASH SAMPLING DESIRED-	GEOMBX 00529
705 CONTINUE	GEOMBX 00530
COPLAN = .F.	GEOMBX 00531
IF (ISMPLW .EQ. 0) GO TO 720	GEOMBX 00532
C BYPASS THE TAIL PLANFORM AND BOX CODE SETUP, AND LOOP ON SAMPL	GEOMBX 00533
C CHORDS TO DEFINE WING WAKE REGION	GEOMBX 00534
C (DO 708 JCHRD = 1,ISMPLW)	FTNX1 00025
JCHRD = 1	FTNX1 00026
704 CONTINUE	FTNX1 00027
JT = (JCHRD(JCHRD)-1)*NSUBDV + NSUBCN	GEOMBX 00536

IF (IBOXF(JCHRD) .EQ. INIT) IBOXF(JCHRD) = (TEXLOC(JT)-IXBW)/	GEOMBX 00537
1 NSUBDV + 1	GEOMBX 00538
IF (IBOXL(JCHRD) .EQ. INIT) IBOXL(JCHRD) = (TEXLOC(JT)-IXBW)/	GEOMBX 00539
1 NSUBDV + 1	GEOMBX 00540
YCT = JT - .5	GEOMBX 00541
ICT = (IBOXL(JCHRD)-1) * NSUBDV + IXBW	GEOMBX 00542
EL = COS(PSIW) * ZLOC(JCHRD) - SIN(PSIW) * YCT	GEOMBX 00543
CAPLL = ZLOC(JCHRD)	GEOMBX 00544
GO TO 580	GEOMBX 00545
C THE LOGIC FOR A TAIL CHORD IS USED. AFTER THE WING WAKE	GEOMBX 00546
C BOUNDS ARE DETERMINED FOR THIS TAIL CHORD; CONTROL IS RETURNED	GEOMBX 00547
C TO THIS LOOP	GEOMBX 00548
706 CONTINUE	GEOMBX 00549
708 CONTINUE	GEOMBX 00550
JCHRD = JCHRD + 1	FTNXL 00028
IF (JCHRD .LE. ISMFLW) GO TO 704	FTNXL 00029
C END OF LOOP ON SAMPLE CHORDS	GEOMBX 00551
GO TO 685	GEOMBX 00552
C	GEOMBX 00553
C GET DIAPHRAGM BOXES CODES FOR THE WING	GEOMBX 00554
720 CONTINUE	GEOMBX 00555
CALL BXCDI (IWAKE,LBXCW,LSCHDS, IBOXW)	GEOMBX 00556
C RETURNS - IBOXW, CODES 2 AND 3 ADDED FOR DIAPHRAGM REGIONS	GEOMBX 00557
C MYBBSW = NUMBER OF SUBDIVIDED CHORDS, INCLUDING DIA-	GEOMBX 00558
C PHRAGM	GEOMBX 00559
C MYBBW = NUMBER OF UNSUBDIVIDED CHORDS	GEOMBX 00560
C	GEOMBX 00561
C PRINT BOX CODES	GEOMBX 00562
IF (.NOT. (PRBOX .OR. CHECKPR)) GO TO 725	GEOMBX 00563
CALL PRNTBC(IBOXW,LBXCW, 1, MXBBSW, MYBBSW, .T.)	GEOMBX 00564
IF (NSUBDV .NE. 1) CALL PRNTBC(IBOXW,LBXCW,	GEOMBX 00565
1 1, (MXBBSW-IXBW+NSUBDV)/NSUBDV, MYBBW, .F.)	GEOMBX 00566
725 CONTINUE	GEOMBX 00567
C	GEOMBX 00568
C DETERMINE THE PLANAR AIC ARRAY SIZE	GEOMBX 00569
NFLKRN = MAX0(MXBBW, MXBT-IXBT/NSUBDV + 1)	GEOMBX 00570
IF (COPLAN) NFLKRN = MXBT	GEOMBX 00571
C	GEOMBX 00572
C WRITE THE BOX CODE ARRAYS INTO THE GEOMETRY SCRATCH FILE	GEOMBX 00573
REWIND IGEOSC	GEOMBX 00574
CALL RDINIT	GEOMBX 00575
ITYPE = 5HMIKED	GEOMBX 00576
IVAL = 5HIBOXW	GEOMBX 00577
PARM(1) = 0.	GEOMBX 00578
PARM(2) = 5HACH	GEOMBX 00579
M = MXBBSW	GEOMBX 00580
N = (MYBBSW-1)/NBWRD + 1	GEOMBX 00581
K = LBXCW	GEOMBX 00582
CALL WRTEMX(IGEOSC,MXWRIT,RANDU,NFS,NMS,LS,NMR,LWS,K,ID,	GEOMBX 00583
1 IBOXW, ITYPE, M,N, PARM, IRR)	GEOMBX 00584
IF (IRR .NE. 0) GO TO 8070	GEOMBX 00585
C	GEOMBX 00586
IF (NSURF .EQ. 1) GO TO 730	GEOMBX 00587
IF (COPLAN) GO TO 730	GEOMBX 00588
IVAL = 5HIBOXT	GEOMBX 00589
M = MXBST - IXBST + 1	GEOMBX 00590
N = (MYBBST - 1)/NBWRD + 1	GEOMBX 00591

K = LBXCDT	GEOMBX 00592
CALL WRTEMX(IGEOSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,K,ID,	GEOMBX 00593
1 IBOXT, ITYPE, M,N, PARM, IRR)	GEOMBX 00594
IF (IRR .NE. 0) GO TO 8070	GEOMBX 00595
C	GEOMBX 00596
C CHECK FOR DIAPHRAGMS CROSSING VERTICALLY	GEOMBX 00597
IF (YCROSS .LE. .5) GO TO 730	GEOMBX 00598
IF (FLOAT(MYBBT)*COS(PSIT) .LT. YCROSS) GO TO 730	GEOMBX 00599
IF (FLOAT(MYBBW)*COS(PSIW) .GE. YCROSS) GO TO 8080	GEOMBX 00600
C	GEOMBX 00601
C WRITE THE LEADING AND TRAILING EDGE LOCATIONS ONTO SCRATCH	GEOMBX 00602
730 CONTINUE	GEOMBX 00603
M = 1	GEOMBX 00604
N = MYBSW + MYBST	GEOMBX 00605
K = 1	GEOMBX 00606
IVAL = GHFEXLOC	GEOMBX 00607
CALL WRTEMX(IGEOSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS, K, ID,	GEOMBX 00608
1 FEXLOC, ITYPE, M,N, PARM, IRR)	GEOMBX 00609
IF (IRR .NE. 0) GO TO 8070	GEOMBX 00610
IVAL = GHTEXLOC	GEOMBX 00611
CALL WRTEMX(IGEOSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS, K, ID,	GEOMBX 00612
1 TEXLOC, ITYPE, M,N, PARM, IRR)	GEOMBX 00613
IF (IRR .NE. 0) GO TO 8070	GEOMBX 00614
C	GEOMBX 00615
C DETERMINE THE ON-PLANFORM FRACTIONAL PART OF ALL UNSUBDIVIDED	GEOMBX 00616
C BOXES CUT BY A PLANFORM EDGE	GEOMBX 00617
CALL GHAREA(IBOXW,LBXCOW, .T., ALPHA,IJALPH, NALPHW)	GEOMBX 00618
NALPH = NALPHW	GEOMBX 00619
IF (NSURF .EQ. 1 .OR. CORLAN) GO TO 740	GEOMBX 00620
CALL GHAREA(IBOXT(ISUBT,1),LBXCDT, .F., ALPHA(NALPHW+1),	GEOMBX 00621
1 IJALPH(NALPHW+1), NALPHT)	GEOMBX 00622
NALPH = NALPH + NALPHT	GEOMBX 00623
740 CONTINUE	GEOMBX 00624
IF (CHECKPR) WRITE(NT6,7030) (IJALPH(I), ALPHA(I),I=1,NALPH)	GEOMBX 00625
C	GEOMBX 00626
C WRITE THE AREA MULTIPLIERS	GEOMBX 00627
M = 1	GEOMBX 00628
N = NALPH	GEOMBX 00629
K = 1	GEOMBX 00630
IPARM(3) = NALPHW	GEOMBX 00631
IVAL = GHIALPHA	GEOMBX 00632
CALL WRTEMX(IGEOSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS, K, ID,	GEOMBX 00633
1 ALPHA, ITYPE, M,N, PARM, IRR)	GEOMBX 00634
IF (IRR .NE. 0) GO TO 8070	GEOMBX 00635
IVAL = GHIJALPH	GEOMBX 00636
CALL WRTEMX(IGEOSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS, K, ID,	GEOMBX 00637
1 IJALPH, ITYPE, M,N, PARM, IRR)	GEOMBX 00638
IF (IRR .NE. 0) GO TO 8070	GEOMBX 00639
C	GEOMBX 00640
C DETERMINE THE SPATIAL AIC PARAMETERS	GEOMBX 00641
C THE MUIC ARRAYS ARE WRITTEN TEMPORARILY ON IVPSC FOR EDITTING	GEOMBX 00642
C ONTO IWTFSC. AFTER THE KPT-- ARRAYS ARE WRITTEN ON	GEOMBX 00643
C IGEOSC, ALL NSPATK ARRAYS ARE TRANSFERRED TO IGEOSC	GEOMBX 00644
C ISCR = NUMBER OF MUICS TRANSFERRED TO IWTFSC	GEOMBX 00645
C NSCR = TOTAL NUMBER OF MUICS PRESENTLY ON IVPSC	GEOMBX 00646
C	GEOMBX 00647
ISCR = 0	GEOMBX 00648

NBCR = 0	GEOMX 00649
NRCBR = 0	GEOMX 00650
NRCBLL = 0	GEOMX 00651
IPARM(3) = 0	GEOMX 00652
IPARM(6) = 0	GEOMX 00653
REWIND IVPSC	GEOMX 00654
REWIND IVPSC	GEOMX 00655
M = 2	GEOMX 00656
K = 2	GEOMX 00657
NSPATK = 0	GEOMX 00658
NAK = 0	GEOMX 00659
IF (SYM .EQ. 0) GO TO 904	GEOMX 00660
IF (PSIW .EQ. 0 .OR. .NOT. DIHW) GO TO 800	GEOMX 00661
C	GEOMX 00662
C START OF LOOP FOR WING-WING PARAMETERS ON RECEIVING CHORDS,	GEOMX 00663
C UNSUBDIVIDED (DETERMINES SPATIAL MUAIC VALUES)	GEOMX 00664
C DO 790 JCCL = 1, MYBBW	GEOMX 00665
C	GEOMX 00666
C CALL PMAIC (.T., IBOXW, LBXCDW, IWAKE, JCCL)	GEOMX 00667
C COMPUTES MUAIC ARRAY FOR THE LEFT SURFACE CONTRIBUTION TO	GEOMX 00668
C CHORD JCCL OF THE RIGHT SURFACE	GEOMX 00669
C SURF = .T. INDICATES SOME RIGHT SURFACE CONTRIBUTION WAS	GEOMX 00670
C FOUND	GEOMX 00671
C IF (.NOT. SURF) GO TO 800	GEOMX 00672
C NAK = NAK + 1	GEOMX 00673
C NSPATK = NAK	GEOMX 00674
C KPTW(NAK) = NSPATK	GEOMX 00675
C IVAL = 10H WING-WING	GEOMX 00676
C IF (CHECKFR) WRITE(NT6, 7020) IVAL, JCCL, YBAR, EL, NROWS, (MUAIC(1,	GEOMX 00677
C 1 NROWS-I+1), MUAIC(2, NROWS-I+1), I=1, NROWS)	GEOMX 00678
C	GEOMX 00679
C WRITE MUAIC ARRAY ON THE SCRATCH FILE	GEOMX 00680
C N = NROWS	GEOMX 00681
C PARM(4) = YBAR	GEOMX 00682
C PARM(5) = EL	GEOMX 00683
C CALL WRTEMX(IVPSC, MXWRIT, RANDOU, NFS, NMS, LS, NMR, LWS, K, ID,	GEOMX 00684
C 1 MUAIC, ITYPE, M, N, PARM, IRR)	GEOMX 00685
C IF (IRR .NE. 0) GO TO 8075	GEOMX 00686
C NBCR = NBCR + 1	GEOMX 00687
C	GEOMX 00688
C 790 CONTINUE	GEOMX 00689
C END OF LOOP ON RECEIVING CHORDS FOR WING-WING PARAMETERS,	GEOMX 00690
C	GEOMX 00691
C 800 CONTINUE	GEOMX 00692
C NTK = 0	GEOMX 00693
C IF (NSURF .NE. 2) GO TO 900	GEOMX 00694
C IF (SYMT .EQ. 0) GO TO 900	GEOMX 00695
C IF (PSIT .EQ. 0 .OR. .NOT. DIHT) GO TO 900	GEOMX 00696
C IF (PSIDIF .EQ. 0.) REWIND IVPSC	GEOMX 00697
C	GEOMX 00698
C START OF LOOP FOR TAIL-TAIL PARAMETERS ON RECEIVING CHORDS,	GEOMX 00699
C UNSUBDIVIDED	GEOMX 00700
C DO 895 JCCL = 1, MYBBT	GEOMX 00701
C	GEOMX 00702
C CALL PMAIC (.F., IBOXT(ISUBT,1), LBXCDT, IWAKE, JCCL)	GEOMX 00703
C COMPUTES MUAIC ARRAY FOR THE CONTRIBUTION OF THE LEFT TAIL ON	GEOMX 00704
C CHORD JCCL OF THE RIGHT TAIL SURFACE	GEOMX 00705

C	IF (.NOT. SURF) GO TO 900	GEOMBX 00706
	NTTK = NTK + 1	GEOMBX 00707
C		GEOMBX 00708
	IVAL = 10H TAIL-TAIL	GEOMBX 00709
C	IF THE WING AND TAIL HAVE THE SAME DIHEDRAL, MODIFY THE WING	GEOMBX 00710
C	MUAIC ARRAY WHERE NEEDED BY THE TAIL, AND USE IT	GEOMBX 00711
	IF (PSIDIF .EQ. 0.) GO TO 840	GEOMBX 00712
C	OTHERWISE, WRITE THE MUAICS FOUND ONTO IVPSC	GEOMBX 00713
	NSPATK = NSPATK + 1	GEOMBX 00714
	KPTTT(NTTK) = NSPATK	GEOMBX 00715
	IVAL = 10H TAIL-TAIL	GEOMBX 00716
	IF (CHECKPR) WRITE(NT6,7020) IVAL,JCOL,YBAR,EL, NROWS, (MUAIC(1,	GEOMBX 00717
1	NROWS-I+1),MUAIC(2,NROWS-I+1), I=1,NROWS)	GEOMBX 00718
C		GEOMBX 00719
C	WRITE MUAIC ARRAY ON SCRATCH FILE	GEOMBX 00720
	N = NROWS	GEOMBX 00721
	PARM(4) = YBAR	GEOMBX 00722
	PARM(5) = EL	GEOMBX 00723
	CALL WRTEMX(IVPSC, MXWRT,RANDCU,NFS,NMS,LS,NMR,LWS, K, ID,	GEOMBX 00724
1	MUAIC, ITYPE, M,N, PARM, IRR)	GEOMBX 00725
	IF (IRR .NE. 0) GO TO 8075	GEOMBX 00726
	NSCR = NSCR + 1	GEOMBX 00727
C		GEOMBX 00728
	GO TO 895	GEOMBX 00729
840	CONTINUE	GEOMBX 00730
	IF (ISCR .GE. NSCR) GO TO 850	GEOMBX 00731
	CALL RDINIT	GEOMBX 00732
	CALL READMX(IVPSC, MXWRT,RANDCU,NFS,NMS,LS,NMR, K, NID,ID,	GEOMBX 00733
1	ITYPE, LRS, MUAICL, M,N, PARM, IRR)	GEOMBX 00734
	IF (IRR .NE. 0) GO TO 8090	GEOMBX 00735
	ISCR = ISCR + 1	GEOMBX 00736
C	MERGE THE TWO MUAIC ARRAYS	GEOMBX 00737
	DO 845 I = 1,N	GEOMBX 00738
	IF (I .GT. NROWS) GO TO 842	GEOMBX 00739
	IF (MUAICL(1,I) .EQ. 0) GO TO 845	GEOMBX 00740
	IF (MUAIC(1,I) .EQ. 0) GO TO 842	GEOMBX 00741
	MUAIC(1,I) = MIN0(MUAIC(1,I),MUAICL(1,I))	GEOMBX 00742
	MUAIC(2,I) = MAX0(MUAIC(2,I),MUAICL(2,I))	GEOMBX 00743
	GO TO 845	GEOMBX 00744
842	MUAIC(1,I) = MUAICL(1,I)	GEOMBX 00745
	MUAIC(2,I) = MUAICL(2,I)	GEOMBX 00746
845	CONTINUE	GEOMBX 00747
	NROWS = MAX0(NROWS,N)	GEOMBX 00748
	KPTTT(NTTK) = ISCR	GEOMBX 00749
	IF (CHECKPR) WRITE(NT6,7020) IVAL,JCOL,YBAR,EL, NROWS,	GEOMBX 00750
1	(MUAIC(1,NROWS-I+1),MUAIC(2,NROWS-I+1), I = 1,NROWS)	GEOMBX 00751
	GO TO 855	GEOMBX 00752
C	THERE WERE NO MATRICES TO BE MERGED	GEOMBX 00753
850	NSPATK = NSPATK + 1	GEOMBX 00754
	KPTTT(NTTK) = NSPATK	GEOMBX 00755
C	WRITE MERGED AICS ONTO 2ND SCRATCH FILE	GEOMBX 00756
855	CONTINUE	GEOMBX 00757
	N = NROWS	GEOMBX 00758
	CALL WRTEMX(IWTFSC, MXWRT,RANDCU,NFS,NMS,LS,NMR,LWS, K, ID,	GEOMBX 00759
1	MUAIC, ITYPE, M,N, PARM, IRR)	GEOMBX 00760
	IF (IRR .NE. 0) GO TO 8110	GEOMBX 00761
		GEOMBX 00762

C		GEOMBX	00763
	895 CONTINUE	GEOMBX	00764
C	END OF LOOP FOR TAIL-TAIL PARAMETERS, FROM 800*	GEOMBX	00765
C		GEOMBX	00766
	900 CONTINUE	GEOMBX	00767
C	COMPLETE ANY COPY FROM FIRST TO SECOND SCRATCH FILE	GEOMBX	00768
	IF (ISCR .EQ. 0) REWIND IVPSC	GEOMBX	00769
	IF (ISCR .GE. NSCR) GO TO 904	GEOMBX	00770
	II = ISCR + 1	GEOMBX	00771
	DO 902 I = II, NSCR	GEOMBX	00772
	CALL RDINIT	GEOMBX	00773
	CALL READMX(IVPSC, MXWRIT, RANDOU, NFS, NMS, LS, NMR, K, NID, ID,	GEOMBX	00774
	1 ITYPE, LRS, MUAIC, M, N, PARM, IRR)	GEOMBX	00775
	IF (IRR .NE. 0) GO TO 8090	GEOMBX	00776
	CALL WRTEMX(IWTFSC, MXWRIT, RANDOU, NFS, NMS, LS, NMR, LWS, K, ID,	GEOMBX	00777
	1 MUAIC, ITYPE, M, N, PARM, IRR)	GEOMBX	00778
	IF (IRR .NE. 0) GO TO 8110	GEOMBX	00779
	902 CONTINUE	GEOMBX	00780
	904 CONTINUE	GEOMBX	00781
C	COMPUTE THE RIGHT AND LEFT WING INFLUENCE PARAMETERS ON THE	GEOMBX	00782
C	TAIL OR SAMPLE CHORDS	GEOMBX	00783
	NRWTK = 0	GEOMBX	00784
	NLWTK = 0	GEOMBX	00785
	ISCR = 0	GEOMBX	00786
	NSCR2 = 0	GEOMBX	00787
	REWIND IVPSC	GEOMBX	00788
	IPARM(6) = 1	GEOMBX	00789
C	INITIALIZE THE MUAIC ARRAYS	GEOMBX	00790
	DO 908 I = 1, 50	GEOMBX	00791
	MUAIC(1, I) = I + I	GEOMBX	00792
	MUAIC(2, I) = 0	GEOMBX	00793
	MUAICL(1, I) = I + I	GEOMBX	00794
	MUAICL(2, I) = 0	GEOMBX	00795
	908 CONTINUE	GEOMBX	00796
	IF (CCPLAN) GO TO 1015	GEOMBX	00797
	IF (NSURF .NE. 2) GO TO 1120	GEOMBX	00798
	CAPLL = CAPL	GEOMBX	00799
	YMUVP = CAPLL * SIN(PSIW)	GEOMBX	00800
	JTCOL = MYBSW - NSUBD2	GEOMBX	00801
C		GEOMBX	00802
C	START OF LOOP ON TAIL CHORDS, TO COMPUTE	GEOMBX	00803
C	WING - TAIL INFLUENCE PARAMETERS	GEOMBX	00804
C	(DO 1010 JCQL = 1, MYBBT)	FTNX1	00030
	JCQL = 1	FTNX1	00031
	909 CONTINUE	FTNX1	00032
	IF (JCQL .LE. MYBT) GO TO 910	GEOMBX	00806
	IROW = (IXBT - IXBW) / NSUBDV + 1	GEOMBX	00807
	GO TO 915	GEOMBX	00808
	910 CONTINUE	GEOMBX	00809
	JTCOL = JTCOL + NSUBDV	GEOMBX	00810
	IROW = (TEXLOC(JTCOL) - IXBW) / NSUBDV + 1	GEOMBX	00811
	IF (IROW .EQ. MXBT) GO TO 930	GEOMBX	00812
	915 CONTINUE	GEOMBX	00813
	CALL DCDER(IXBT(IUBT, 1), LBXCDT, IROW, JCQL, MXBT, JCQL, .F., ICODE)	GEOMBX	00814
	II = 1	GEOMBX	00815
	DO 917 I = IROW, MXBT	GEOMBX	00816
	IF (ICODE(II) .NE. 0) GO TO 918	GEOMBX	00817

II = II + 1	GEOMX 00818
917 CONTINUE	GEOMX 00819
918 CONTINUE	GEOMX 00820
III = 1	GEOMX 00821
DO 920 I = III, MXBT	GEOMX 00822
IF (ICODF(II) .EQ. 0) GO TO 925	GEOMX 00823
II = II + 1	GEOMX 00824
920 CONTINUE	GEOMX 00825
925 IROW = IROW + II - 2	GEOMX 00826
C ENTRY INTO THE LOOP FROM SAMPLE WASH LOOP, FROM 1120*	GEOMX 00827
930 CONTINUE	GEOMX 00828
CALL PUTAIC(IBOXW, LBXCOW, IROW, JCCL, CAPLL, YMUVP)	GEOMX 00829
C GETS THE MUAIC AND MUAICL ARRAYS FOR RIGHT AND LEFT CONTRI-	GEOMX 00830
C BUTIONS TO THE TAIL	GEOMX 00831
IF (SURF) GO TO 935	GEOMX 00832
IF (.NOT. SURFL) GO TO 1015	GEOMX 00833
GO TO 985	GEOMX 00834
935 NRWTX = NRWTX + 1	GEOMX 00835
C DETERMINE WHETHER WING AND TAIL ARE PARALLEL	GEOMX 00836
IF (PSIDIF .EQ. 0 .AND. NSURF .EQ. 2) GO TO 940	GEOMX 00837
NSPATK = NSPATK + 1	GEOMX 00838
KPTRWT(NRWTX) = NSPATK	GEOMX 00839
C WRITE NEW MUAIC ARRAY ON SCRATCH	GEOMX 00840
IVAL = 10HR WING-TAIL	GEOMX 00841
N = NROWS	GEOMX 00842
PARM(4) = YBAR	GEOMX 00843
PARM(5) = EL	GEOMX 00844
CALL WRTEX(IVPSC, MXWRIT, RANDOU, NFS, NMS, LS, NMR, LWS, K, ID,	GEOMX 00845
1 MUAIC, ITYPE, M, N, PARM, IRR)	GEOMX 00846
IF (IRR .NE. 0) GO TO 8075	GEOMX 00847
NSCR2 = NSCR2 + 1	GEOMX 00848
IF (CHECKPR) WRITE(NT6, 7020) IVAL, JCCL, YBAR, EL, NROWS, (MUAIC(1,	GEOMX 00849
1 NROWS-I+1), MUAIC(2, NROWS-I+1), I=1, NROWS)	GEOMX 00850
C REINITIALIZE THE RIGHT MUAIC ARRAY	GEOMX 00851
DO 937 I = 1, NROWS	GEOMX 00852
MUAIC(1, I) = I+1	GEOMX 00853
MUAIC(2, I) = 0	GEOMX 00854
937 CONTINUE	GEOMX 00855
C	GEOMX 00856
GO TO 982	GEOMX 00857
C OLD MUAIC ARRAY HAS BEEN FOUND WHICH MATCHES	GEOMX 00858
940 CONTINUE	GEOMX 00859
C ALL RIGHT HAND MUATCS ARE THE SAME, PARALLEL SURFACES	GEOMX 00860
IF (JCCL .GT. 1) GO TO 945	GEOMX 00861
NSPATK = NSPATK + 1	GEOMX 00862
IPARAL = NSPATK	GEOMX 00863
945 CONTINUE	GEOMX 00864
KPTRWT(NRWTX) = IPARAL	GEOMX 00865
NROWSR = MAX0(NROWSR, NROWS)	GEOMX 00866
C	GEOMX 00867
980 CONTINUE	GEOMX 00868
IVAL = 10HR WING-TAIL	GEOMX 00869
IF (CHECKPR) WRITE(NT6, 7020) IVAL, JCCL, YBAR, EL, NROWS, (MUAIC(1,	GEOMX 00870
1 NROWS-I+1), MUAIC(2, NROWS-I+1), I=1, NROWS)	GEOMX 00871
C	GEOMX 00872
982 CONTINUE	GEOMX 00873
IF (.NOT. SURFL) GO TO 1000	GEOMX 00874

985 CONTINUE	GEOMBX 00875
NLWTK = NLWTK + 1	GEOMBX 00876
C DETERMINE WHETHER BOTH SURFACES HAVE NO DIHEDRAL	GEOMBX 00877
IF (PSIW.EQ. 0. .AND. PSIDIF.EQ. 0. .AND. NSURF.EQ. 2) GO TO 990	GEOMBX 00878
NSPATK = NSPATK + 1	GEOMBX 00879
KPTLWT(NLWTK) = NSPATK	GEOMBX 00880
C WRITE NEW MUAIC ARRAY ON SCRATCH	GEOMBX 00881
IVAL = 10HL WNG-TAIL	GEOMBX 00882
N = NROWSL	GEOMBX 00883
PARM(4) = YBARL	GEOMBX 00884
PARM(5) = ELL	GEOMBX 00885
CALL WRTEMX(IVPSC, MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS, K, ID,	GEOMBX 00886
1 MUAICL, ITYPE, M,N, PARM, IRR)	GEOMBX 00887
IF (IRR .NE. 0) GO TO 8075	GEOMBX 00888
NSCR2 = NSCR2 + 1	GEOMBX 00889
IF (CHECKPR) WRITE(NT6,7020) IVAL,JCOL,YBARL,ELL, NROWSL,	GEOMBX 00890
1 (MUAICL(1,NROWSL-I+1),MUAICL(2,NROWSL-I+1), I=1,NROWSL)	GEOMBX 00891
C REINITIALIZE THE LEFT MUAIC ARRAY	GEOMBX 00892
DO 987 I = 1,NROWSL	GEOMBX 00893
MUAICL(1,I) = I+I	GEOMBX 00894
MUAICL(2,I) = 0	GEOMBX 00895
987 CONTINUE	GEOMBX 00896
GO TO 1000	GEOMBX 00897
C OLD MUAIC ARRAY HAS BEEN FOUND WHICH MATCHES	GEOMBX 00898
990 CONTINUE	GEOMBX 00899
C USE THE SAME ARRAY FOR RIGHT AND LEFT CONTRIBUTIONS	GEOMBX 00900
KPTLWT(NLWTK) = IPARAL	GEOMBX 00901
NROWSL = MAX0(NROWSL,NROWSL)	GEOMBX 00902
C	GEOMBX 00903
995 CONTINUE	GEOMBX 00904
IVAL = 10HL WNG-TAIL	GEOMBX 00905
IF (CHECKPR) WRITE(NT6,7020) IVAL,JCOL,YBARL,ELL,NROWSL,	GEOMBX 00906
1 (MUAICL(1,NROWSL-I+1),MUAICL(2,NROWSL-I+1), I=1,NROWSL)	GEOMBX 00907
1000 CONTINUE	GEOMBX 00908
IF (ISMPW .NE. 0) GO TO 1150	GEOMBX 00909
1010 CONTINUE	GEOMBX 00910
JCOL = JCOL + 1	FTNDX 00033
IF (JCOL .LE. MYBBT) GO TO 909	FTNDX 00034
C END OF LOOP ON CHORDS, FOR WING-TAIL PARAMETERS	GEOMBX 00911
C	GEOMBX 00912
C PLACE ANY BUILT UP ARRAY DUE TO PARALLEL SURFACES ON SCRATCH	GEOMBX 00913
IF (PSIDIF .NE. 0 .OR. PSIW .NE. 0) GO TO 1014	GEOMBX 00914
C MERGE THE RIGHT AND LEFT ARRAYS	GEOMBX 00915
NROWX = MIN0(NROWR,NROWSL)	GEOMBX 00916
DO 1012 I = 1,NROWX	GEOMBX 00917
MUAIC(1,I) = MIN0(MUAIC(1,I),MUAICL(1,I))	GEOMBX 00918
MUAIC(2,I) = MAX0(MUAIC(2,I),MUAICL(2,I))	GEOMBX 00919
1012 CONTINUE	GEOMBX 00920
1014 IF (PSIDIF .NE. 0) GO TO 1015	GEOMBX 00921
C WRITE THE ARRAY ONTO IUTFSC	GEOMBX 00922
IVAL = 10HPARAL TAIL	GEOMBX 00923
PARM(4) = YBAR	GEOMBX 00924
PARM(5) = EL	GEOMBX 00925
IPARM(6) = 2	GEOMBX 00926
N = MAX0(NROWR,NROWSL)	GEOMBX 00927
CALL WRTEMX(IUTFSC, MXWRIT,RANDOU,NFS,NIS,LS,NMR,LWS, K, ID,	GEOMBX 00928
1 MUAIC, ITYPE, M,N, PARM, IRR)	GEOMBX 00929

IF (IRR .NE. 0) GO TO 8110	GEOMBX 00930
NSCR = NSCR + 1	GEOMBX 00931
C PLACE THE KPT-- ARRAYS ON GEOMETRY SCRATCH AS ONE MATRIX.	GEOMBX 00932
1015 CONTINUE	GEOMBX 00933
IF (NSPATK .EQ. 0) ENDFILE IGEOSC	GEOMBX 00934
IF (NSPATK .EQ. 0) GO TO 2000	GEOMBX 00935
IPARM(3) = NNAK	GEOMBX 00936
IPARM(4) = NNTK	GEOMBX 00937
IPARM(5) = NEWTK	GEOMBX 00938
IPARM(6) = NLWTK	GEOMBX 00939
M = 0	GEOMBX 00940
N = 0	GEOMBX 00941
IF (NNAK .EQ. 0) GO TO 1030	GEOMBX 00942
M = 1	GEOMBX 00943
N = NNAK	GEOMBX 00944
DO 1020 I = 1, NNAK	GEOMBX 00945
1020 KPT(I, I) = KFTWM(I)	GEOMBX 00946
1030 CONTINUE	GEOMBX 00947
IF (NNTK .EQ. 0) GO TO 1050	GEOMBX 00948
M = M + 1	GEOMBX 00949
N = MAX0(N, NNTK)	GEOMBX 00950
DO 1040 I = 1, NNTK	GEOMBX 00951
1040 KPT(M, I) = KPTTT(I)	GEOMBX 00952
1050 CONTINUE	GEOMBX 00953
IF (NEWTK .EQ. 0) GO TO 1070	GEOMBX 00954
M = M + 1	GEOMBX 00955
N = MAX0(N, NEWTK)	GEOMBX 00956
DO 1060 I = 1, NEWTK	GEOMBX 00957
1060 KPT(M, I) = KPTRWT(I)	GEOMBX 00958
1070 CONTINUE	GEOMBX 00959
IF (NLWTK .EQ. 0) GO TO 1090	GEOMBX 00960
M = M + 1	GEOMBX 00961
N = MAX0(N, NLWTK)	GEOMBX 00962
DO 1080 I = 1, NLWTK	GEOMBX 00963
1080 KPT(M, I) = KPTLWT(I)	GEOMBX 00964
1090 CONTINUE	GEOMBX 00965
K = 4	GEOMBX 00966
IVAL = 3HKPT	GEOMBX 00967
CALL WRTEMX(IGEOSC, MXWRIT, RANDOU, NFS, NMS, LS, NMR, LWS, K, ID,	GEOMBX 00968
1 KPT, ITYPE, M, N, PARM, IRR)	GEOMBX 00969
IF (IRR .NE. 0) GO TO 8070	GEOMBX 00970
END FILE IGEOSC	GEOMBX 00971
C	GEOMBX 00972
C MOVE THE MUAIC ARRAYS TO THE GEOMETRY SCRATCH TAPE	GEOMBX 00973
REWIND IWTFC	GEOMBX 00974
REWIND IVPSC	GEOMBX 00975
IVAL = 6HMMUIC	GEOMBX 00976
K = 2	GEOMBX 00977
IF (NSCR .EQ. 0) GO TO 1096	GEOMBX 00978
DO 1094 I = 1, NSCR	GEOMBX 00979
CALL RDINIT	GEOMBX 00980
CALL READMX(IWTFC, MXWRIT, RANDOU, NFS, NMS, LS, NMR, K, NID, ID,	GEOMBX 00981
1 ITYPE, LRS, MUAIC, M, N, PARM, IRR)	GEOMBX 00982
IF (IRR .NE. 0) GO TO 8100	GEOMBX 00983
CALL WRTEMX(IGEOSC, MXWRIT, RANDOU, NFS, NMS, LS, NMR, LWS, K, ID,	GEOMBX 00984
1 MUAIC, ITYPE, M, N, PARM, IRR)	GEOMBX 00985
IF (IRR .NE. 0) GO TO 8070	GEOMBX 00986

1094 CONTINUE	GEOMBX 00987
1096 IF (NSCR2 .LE. 0) GO TO 1110	GEOMBX 00988
DO 1100 I = 1, NSCR2	GEOMBX 00989
CALL READMX(IVPSC, MXWRIT, RANDOU, NFS, NMS, LS, NMR, K, NID, ID,	GEOMBX 00990
1 ITYPE, LRS, MUAIC, M, N, PARM, IRR)	GEOMBX 00991
IF (IRR .NE. 0) GO TO 8090	GEOMBX 00992
CALL WRITEMX(IGEOSC, MXWRIT, RANDOU, NFS, NMS, LS, NMR, LWS, K, ID,	GEOMBX 00993
1 MUAIC, ITYPE, M, N, PARM, IRR)	GEOMBX 00994
IF (IRR .NE. 0) GO TO 8070	GEOMBX 00995
1100 CONTINUE	GEOMBX 00996
1110 CONTINUE	GEOMBX 00997
END FILE IGEOSC	GEOMBX 00998
GO TO 2000	GEOMBX 00999
C	GEOMBX 01000
C LOOP ON SAMPLE WASH CHORDS (USED IF ISMPLW .NE. 0) TO	GEOMBX 01001
C DETERMINE MUAIC ARRAYS FOR RIGHT AND LEFT WING INFLUENCE	GEOMBX 01002
C ON EACH CHORD	GEOMBX 01003
1120 CONTINUE	GEOMBX 01004
NRWTK = 0	GEOMBX 01005
NLWTK = 0	GEOMBX 01006
IF (ISMPLW .EQ. 0) GO TO 1015	GEOMBX 01007
C (DO 1200 JCHRD = 1, ISMPLW)	FTNXL 00035
JCHRD = 1	FTNXL 00036
1130 CONTINUE	FTNXL 00037
JCOL = JCHORD(JCHRD)	GEOMBX 01009
CAPLL = ZLOC(JCHRD)	GEOMBX 01010
YMUWSP = CAPLL * SIN(PSIW)	GEOMBX 01011
ITROW = IBOML(JCHRD)	GEOMBX 01012
GO TO 930	GEOMBX 01013
C THE LOGIC FOR * TAIL CHORD IS USED. AFTER THE MUAIC ARRAYS	GEOMBX 01014
C ARE DETERMINED AND STORED, CONTROL IS RETURNED TO THIS LOOP.	GEOMBX 01015
1150 CONTINUE	GEOMBX 01016
1200 CONTINUE	GEOMBX 01017
JCHRD = JCHRD + 1	FTNXL 00038
IF (JCHRD .LE. ISMPLW) GO TO 1130	FTNXL 00039
C END OF LOOP ON SAMPLE WASH CHORDS	GEOMBX 01018
C	GEOMBX 01019
GO TO 1015	GEOMBX 01020
C	GEOMBX 01021
2000 CONTINUE	GEOMBX 01022
ENDFILE IGEOSC	GEOMBX 01023
REWIND IGEOSC	GEOMBX 01024
REWIND IWTWSC	GEOMBX 01025
REWIND IVPSC	GEOMBX 01026
RETURN	GEOMBX 01027
C	GEOMBX 01028
C OUTPUT FORMATS	GEOMBX 01029
C CARD F	GEOMBX 01030
0010 FORMAT(1H0,15X,39H - - - GEOMETRIC PARAMETERS - - - /	GEOMBX 01031
1 1H0,29HCARDF -LOCAL AXES DEFINITION-,	GEOMBX 01032
2 4X, 10HX-LOCATION, 4X,10HZ-LOCATION, 4X,	GEOMBX 01033
2 DIHEDRAL ANGLE (PSI) / 27X,5HWING , F10.3,4X, F10.3, 8X,	GEOMBX 01034
3 .2, 8H DEGREES)	GEOMBX 01035
0012 FC 4T(27X,5HTAIL , F10.3,4X, F10.3, 8X, F7.2, 8H DEGREES)	GEOMBX 01036
C CARD G	GEOMBX 01037
0015 FORMAT(1H0,30HCARDG -BOX PATTERN DEFINITION-,5X,6HCHORDS,10X,	GEOMBX 01038
1 6HXCENTR,5X,2HCR,4X,5HXEDGE /37X, 13, 8X, F10.4, 6X,F10.4)	GEOMBX 01039

8017	FORMAT(17X,13H-SAMPLE WASH,I3, 8H CHORDS-,5X,6H1CHORD,6X,5H1BOXF,	GEOMBX	01040
1	5X,5H1BOXL, 6X, 4H2LOC / (37X, 13,8X, 13, 7X, 13, 6X, F7.2))	GEOMBX	01041
C	CARD H	GEOMBX	01042
8021	FORMAT(11H0,40HCARDH -PLANFORM DEFINITION POINT COUNTS-,5X,	GEOMBX	01043
1	12HLEADING EDGE,4X,13HTRAILING EDGE /42X,4HWING, 17, 9X, 17)	GEOMBX	01044
8022	FORMAT(42X,4HTAIL, 17,9X,17)	GEOMBX	01045
C	CARDS I TO L	GEOMBX	01046
8029	FORMAT(11H0,37HCARDI TO CARDL -PLANFORM DEFINITIONS-,9X,1HX,9X,1HY,	GEOMBX	01047
1	4X,12H(LOCAL AXES))	GEOMBX	01048
8030	FORMAT(32X,A10, F9.3, F10.3/ (41X,2F10.3))	GEOMBX	01049
8040	FORMAT(11H0,7X,31H-BOX DIMENSIONS- B1 (LENGTH) = , E18.8,5X,	GEOMBX	01050
1	17HB1/BETA (WIDTH) =, E18.8)	GEOMBX	01051
C		GEOMBX	01052
C	CHECK PRINT FORMATS, USED ONLY WHEN CHECFR = .T.	GEOMBX	01053
7010	FORMAT(13H0IWAKE ARRAY - , 40I3 / (15X,40I3))	GEOMBX	01054
7020	FORMAT(17H0MUMIC ARRAY FOR ,A10, 8H, CHORD I2, 9H, YBAR = F6.3,	GEOMBX	01055
1	7H, EL = F7.2 / 10X, 3HROW,I3,2I4 / (16X,2I4))	GEOMBX	01056
7030	FORMAT(51H0 IJALPH (= J*1000 + I OCTAL) AND ALPHA ARRAYS, AS	GEOMBX	01057
1	7H STORED / 6(5X,13HIJALPH ALPHA) / (6(5X,06, F7.4)))	GEOMBX	01058
7040	FORMAT(21H0 CHECK PRINT, FEXLOC/(10F12.7))	GEOMBX	01059
7045	FORMAT(1H0 14X, 6HTEALOC / (10F12.7))	GEOMBX	01060
C		GEOMBX	01061
C	DIAGNOSTIC FORMATS	GEOMBX	01062
9010	FORMAT(52H0*** WARNING - XEDGE AND XCENR WERE BOTH SPECIFIED.	GEOMBX	01063
1	20H XEDGE WILL BE IGNORED ***)	GEOMBX	01064
9020	FORMAT(43H0*** WARNING - SAMPLE WASH SPECIFICATION SET I2,8H IS IN	GEOMBX	01065
1	55H ERROR. ONLY THE PRECEDING ONES WILL BE CALCULATED ***)	GEOMBX	01066
9030	FORMAT(53H0*** WARNING - SAMPLING OF UPWASHES CANNOT BE DONE IF	GEOMBX	01067
1	35H A TAIL HAS BEEN DEFINED. 1SMPLW =,I3,16H WILL BE IGNORED	GEOMBX	01068
2	4H ***)	GEOMBX	01069
9110	FORMAT(23H0*** ERROR - PARAMETER ,A6,23H WAS NOT SPECIFIED. IT	GEOMBX	01070
1	25H MUST ALWAYS BE GIVEN ***)	GEOMBX	01071
9120	FORMAT(53H0*** ERROR - EITHER XEDGE OR XCENR MUST BE SPECIFIED	GEOMBX	01072
1	4H ***)	GEOMBX	01073
9130	FORMAT(13H0*** ERROR - ,A6,29H IS OUTSIDE ALLOWED RANGE ***)	GEOMBX	01074
9150	FORMAT(13H0*** ERROR - ,A10,23HDEFINITION POINTS ERROR, I3,	GEOMBX	01075
1	25H, A COMBINATION OF- *** / 13X,24H1, NON-MONOTONIC Y-VALUE	GEOMBX	01076
2	1H8, 10X,25H2, NON-MONOTONIC X-VALUES / 13X,11H4, FIRST Y-	GEOMBX	01077
3	14HVALUE NON-ZERO, 10X,34H8, TIP T.E. Y-VALUE DISAGREES WITH	GEOMBX	01078
4	15H TIP L.E. VALUE)	GEOMBX	01079
9160	FORMAT(51H0*** ERROR - XCENR NOT WITHIN 50 BOX LENGTHS (B1 = ,	GEOMBX	01080
1	E15.8,20H) OF THE WING L.E. (,E15.8, 5H) ***)	GEOMBX	01081
9170	FORMAT(52H0*** ERROR - WHILE WRITING ON GEOMETRY SCRATCH FILE A10,	GEOMBX	01082
1	15H, ERROR CODE = I4, 4H ***)	GEOMBX	01083
9172	FORMAT(14X, 6HARRAY ,A6,15H, DIMENSIONED (I4,1H,I4,11H) WAS BEING	GEOMBX	01084
1	3H WRITTEN)	GEOMBX	01085
9175	FORMAT(14X,20HTHE MUMIC ARRAY FOR A10,15H, DIMENSIONED (I4,1H,I4,	GEOMBX	01086
1	15H) WAS BEING WRITTEN)	GEOMBX	01087
9180	FORMAT(52H0*** ERROR - THE TAIL AND WING, OR THEIR DIAPHRAGMS,	GEOMBX	01088
1	28H CROSS - ABOVE TO BELOW ***)	GEOMBX	01089
9190	FORMAT(54H0*** ERROR - WHILE READING FROM GEOMETRY SCRATCH FILE	GEOMBX	01090
1	A10,15H, ERROR CODE = I4, 4H ***)	GEOMBX	01091
9192	FORMAT(14X, 6HARRAY ,A6,15H, DIMENSIONED (I4,1H,I4,11H) WAS BEING	GEOMBX	01092
1	3H READ)	GEOMBX	01093
C		GEOMBX	01094
C	ERRORS - ALL ERRORS CALL FLUSH	GEOMBX	01095
9010	CONTINUE	GEOMBX	01096

```

      WRITE (NT6,9110) IVAL
      GO TO 8900
8015 CONTINUE
      WRITE (NT6,9130) IVAL
      GO TO 8900
8020 CONTINUE
      WRITE (NT6,9120)
      GO TO 8900
8030 CONTINUE
      WRITE (NT6,9130) IVAL
      GO TO 8900
8040 CONTINUE
      GO TO 8900
8050 CONTINUE
      WRITE (NT6,9150) IVAL,IRR
      GO TO 8900
8060 CONTINUE
      WRITE (NT6,9160) B1, XEDGEW
      GO TO 8900
8070 CONTINUE
      WRITE (NT6,9170) IGEOSC, IRR
      WRITE (NT6,9172) IVAL, M,N
      GO TO 8900
8075 CONTINUE
      WRITE (NT6,9170) IVPSC,IRR
      WRITE (NT6,9175) IVAL, M,N
      GO TO 8900
8080 CONTINUE
      WRITE (NT6,9180)
      GO TO 8900
8090 CONTINUE
      WRITE (NT6,9190) IVPSC,IRR
      WRITE (NT6,9192) IVAL, M,N
      GO TO 8900
8100 WRITE (NT6,9190) IWFSC,IRR
      WRITE (NT6,9192) IVAL, M,N
      GO TO 8900
8110 WRITE (NT6,9170) IWFSC,IRR
      WRITE (NT6,9175) IVAL, M,N
8900 CALL FLUSH(1)
      END

```

```

GEOMBX 01097
GEOMBX 01098
GEOMBX 01099
GEOMBX 01100
GEOMBX 01101
GEOMBX 01102
GEOMBX 01103
GEOMBX 01104
GEOMBX 01105
GEOMBX 01106
GEOMBX 01107
GEOMBX 01108
GEOMBX 01109
GEOMBX 01110
GEOMBX 01111
GEOMBX 01112
GEOMBX 01113
GEOMBX 01114
GEOMBX 01115
GEOMBX 01116
GEOMBX 01117
GEOMBX 01118
GEOMBX 01119
GEOMBX 01120
GEOMBX 01121
GEOMBX 01122
GEOMBX 01123
GEOMBX 01124
GEOMBX 01125
GEOMBX 01126
GEOMBX 01127
GEOMBX 01128
GEOMBX 01129
GEOMBX 01130
GEOMBX 01131
GEOMBX 01132
GEOMBX 01133
GEOMBX 01134
GEOMBX 01135
GEOMBX 01136
GEOMBX 01137

```

SUBROUTINE EDGCHK(XEDGE,YEDGE,NEDGE,IEDGE,IRR)	EDGCHK 00002
C	EDGCHK 00003
C CHECKS FOR MONOTONIC EDGE VALUES OF X FOR LEADING EDGES, AND	EDGCHK 00004
C Y FOR EITHER LEADING OR TRAILING. CHECKS Y FIRST AND LAST	EDGCHK 00005
C VALUES TO ENSURE DEFINITION FROM CENTERLINE TO TIP	EDGCHK 00006
C XEDGE = X-VALUES FOR ONE PLANFORM EDGE	EDGCHK 00007
C YEDGE = Y-VALUES	EDGCHK 00008
C NEDGE = NUMBER OF (XEDGE,YEDGE) SETS DEFINING THE PLAN-	EDGCHK 00009
C FORM EDGE	EDGCHK 00010
C IEDGE = 1 FOR LEADING EDGE, 2 FOR TRAILING EDGE	EDGCHK 00011
C IRR = ERROR RETURN, 0 SUCCESSFUL	EDGCHK 00012
C 1, NON-MONOTONIC Y-VALUES	EDGCHK 00013
C 2, NON-MONOTONIC X-VALUES, LEADING EDGE ONLY	EDGCHK 00014
C 4, FIRST Y-VALUE NON-ZERO	EDGCHK 00015
C 8, TIP Y-VALUE OF A T.E. DISAGREES WITH PREVIOUS	EDGCHK 00016
C L.E. TIP VALUE	EDGCHK 00017
C	EDGCHK 00018
C DIMENSION XEDGE(1),YEDGE(1)	EDGCHK 00019
C	EDGCHK 00020
C IRR = 0	EDGCHK 00021
C IF (YEDGE(1) .NE. 0.) IRR = 4	EDGCHK 00022
C DO 100 I = 2,NEDGE	EDGCHK 00023
C IF (YEDGE(I) .LT. YEDGE(I-1)) GO TO 150	EDGCHK 00024
C GO TO (50,100), IEDGE	EDGCHK 00025
C 50 IF (XEDGE(I) .LT. XEDGE(I-1)) GO TO 200	EDGCHK 00026
C 100 CONTINUE	EDGCHK 00027
C GO TO 250	EDGCHK 00028
C 150 IRR = IRR + 1	EDGCHK 00029
C GO TO 250	EDGCHK 00030
C 200 IRR = IRR + 2	EDGCHK 00031
C 250 CONTINUE	EDGCHK 00032
C GO TO (300,350), IEDGE	EDGCHK 00033
C 300 YTIP = YEDGE(NEDGE)	EDGCHK 00034
C GO TO 500	EDGCHK 00035
C 350 IF (YEDGE(NEDGE) .NE. YTIP) IRR = IRR + 8	EDGCHK 00036
C	EDGCHK 00037
C 500 RETURN	EDGCHK 00038
C	EDGCHK 00039
C	EDGCHK 00040
END	

	SUBROUTINE EDGCHK(XEDGE,YEDGE,NEDGE,IEDGE,IRR)	EDGCHK 00002
C		EDGCHK 00003
C	CHECKS FOR MONOTONIC EDGE VALUES OF X FOR LEADING EDGES, AND	EDGCHK 00004
C	Y FOR EITHER LEADING OR TRAILING. CHECKS Y FIRST AND LAST	EDGCHK 00005
C	VALUES TO ENSURE DEFINITION FROM CENTERLINE TO TIP	EDGCHK 00006
C	XEDGE = X-VALUES FOR ONE PLANFORM EDGE	EDGCHK 00007
C	YEDGE = Y-VALUES	EDGCHK 00008
C	NEDGE = NUMBER OF (XEDGE,YEDGE) SETS DEFINING THE PLAN-	EDGCHK 00009
C	FORM EDGE	EDGCHK 00010
C	IEDGE = 1 FOR LEADING EDGE, 2 FOR TRAILING EDGE	EDGCHK 00011
C	IRR = ERROR RETURN, 0 SUCCESSFUL	EDGCHK 00012
C	1, NON-MONOTONIC Y-VALUES	EDGCHK 00013
C	2, NON-MONOTONIC X-VALUES, LEADING EDGE ONLY	EDGCHK 00014
C	4, FIRST Y-VALUE NON-ZERO	EDGCHK 00015
C	8, TIP Y-VALUE OF A T.E. DISAGREES WITH PREVIOUS	EDGCHK 00016
C	L.E. TIP VALUE	EDGCHK 00017
C		EDGCHK 00018
	DIMENSION XEDGE(1),YEDGE(1)	EDGCHK 00019
C		EDGCHK 00020
	IRR = 0	EDGCHK 00021
	IF (YEDGE(1) .NE. 0.) IRR = 4	EDGCHK 00022
	DO 100 I = 2,NEDGE	EDGCHK 00023
	IF (YEDGE(I) .LT. YEDGE(I-1)) GO TO 150	EDGCHK 00024
	GO TO (50,100), IEDGE	EDGCHK 00025
	50 IF (XEDGE(I) .LT. XEDGE(I-1)) GO TO 200	EDGCHK 00026
	100 CONTINUE	EDGCHK 00027
	GO TO 250	EDGCHK 00028
	150 IRR = IRR + 1	EDGCHK 00029
	GO TO 250	EDGCHK 00030
	200 IRR = IRR + 2	EDGCHK 00031
	250 CONTINUE	EDGCHK 00032
	GO TO (300,350), IEDGE	EDGCHK 00033
	300 YTIP = YEDGE(NEDGE)	EDGCHK 00034
	GO TO 500	EDGCHK 00035
	350 IF (YEDGE(NEDGE) .NE. YTIP) IRR = IRR + 8	EDGCHK 00036
C		EDGCHK 00037
	500 RETURN	EDGCHK 00038
C		EDGCHK 00039
	END	EDGCHK 00040

```

      IB = IB + 1
1000 CONTINUE
      GO TO 3000
C
C      PROGRAM WILL RETRIEVE NJ BOXES FROM CHORD J
1100 CONTINUE
      JSB = (J-1)/NBWRD + 1
      JB = (NBWRD - MOD(J,NBWRD)) * 3
      IF (JB .EQ. 00) JB = 0
      IJMASK = SHIFT(MASK,JB)
      NJB = -JB
      DO 2000 II = 1,IEND,ISKIP
      IJWORD = IBOX(II,JSB)
      IJCODE = IJWORD.AND.IJMASK
      ICODE(IB) = SHIFT(IJCODE,NJB)
      IB = IB + 1
2000 CONTINUE
C
3000 CONTINUE
      RETURN
      END

```

```

DCODER 00055
DCODER 00056
DCODER 00057
DCODER 00058
DCODER 00059
DCODER 00060
DCODER 00061
DCODER 00062
DCODER 00063
DCODER 00064
DCODER 00065
DCODER 00066
DCODER 00067
DCODER 00068
DCODER 00069
DCODER 00070
DCODER 00071
DCODER 00072
DCODER 00073
DCODER 00074
DCODER 00075

```

[illegible]

SUBROUTINE PRNTBC(IBOX,LBXCD, IFRST,ILAST, JLAST, SUBD)	PRNTBC 00002
DIMENSION IBOX(LBXCD,1),ICCODE(150)	PRNTBC 00003
LOGICAL SUBD	PRNTBC 00004
C	PRNTBC 00005
C PRINTS BOX CODES, SUBDIVIDED OR UNSUBDIVIDED	PRNTBC 00006
C IBOX - COMPRESSED BOX CODE ARRAY	PRNTBC 00007
C LBXCD - ROW DIMENSION OF BOX CODE ARRAY	PRNTBC 00008
C IFRST - FIRST ROW DESIRED TO PRINT	PRNTBC 00009
C ILAST - LAST ROW DESIRED	PRNTBC 00010
C JLAST - LAST CHORD DESIRED (FIRST IS ALWAYS ONE)	PRNTBC 00011
C SUBD - .T., SUBDIVIDED CODES DESIRED	PRNTBC 00012
C .F., UNSUBDIVIDED CODES DESIRED	PRNTBC 00013
C	PRNTBC 00014
COMMON /GEOMTY/ COPLAN,NSUBDV,XSUBDV,NSUBD2,NSUBCN,NSURF,	GEOMTY 00002
1 B1,B1BETA,B1S,B1BTAS,WLAX,WLAZ,PSIW,	GEOMTY 00003
2 MXBW,MXBBW,MYBW,MYBBW,MXBSW,MYBSW,MYBBSW,	GEOMTY 00004
3 IXBW,XCENTR	GEOMTY 00005
LOGICAL COPLAN	GEOMTY 00006
COMMON /CONTRL/ PREVECH,OMACH, TITLE(8), PRVGEOM,PRVMODE,DIHW,DIHT,	CONTRL 00002
1 DEFAULT	CONTRL 00003
LOGICAL PRVGEOM,PRVMODE,DIHW,DIHT,DEFAULT	CONTRL 00004
COMMON /PROBLM/ XMACH,NMOCES,NTSLOP,NKVALS,SMOOTH,NDEG,CRDFIT,	PROBLM 00002
1 EXAIC,SUBDV,PLYWOOD	PROBLM 00003
LOGICAL SMOOTH,CRDFIT,EXAIC,SUBDV,PLYWOOD	PROBLM 00004
COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUTP,	FILES 00002
1 ICUFSP,MOCESC,IVPSC,IGEOSC,IWTFSC,IAICSC	FILES 00003
DIMENSION BCD(4)	PRNTBC 00020
INTEGER BCD	PRNTBC 00021
DATA BCD/ 1H ,1H1,1H2,1H3 /	PRNTBC 00022
DATA NBWRD /20 /	FTNXL 00040
6001 FORMAT(1H1, 5X,8A10/1H0,20X,16HBOX CODE PATTERN)	PRNTBC 00023
6002 FORMAT(19X,20HFOR SUBDIVIDED BOXES,25X, 6HIXBW =,12,11H (SUBDIVID	PRNTBC 00024
1 30HED ROW OF UNSUBDIVIDED CENTER))	PRNTBC 00025
6005 FORMAT(22X,4HMACH,F11.7,56X,*CCOE - 1 = PLANFORM BOX* /	PRNTBC 00026
1 19X,20(1H-),61X,*2 = DIAPHRAGM BOX* /100X,*3 = WAKE BOX *)	PRNTBC 00027
6010 FORMAT(1H0,4X,3I14 / (9X,30I4))	PRNTBC 00028
6012 FORMAT(1H)	PRNTBC 00029
6020 FORMAT(1X,13,2X,63A2 / (12X,60A2))	PRNTBC 00030
C	PRNTBC 00031
WRITE (NT6,6001) TITLE	PRNTBC 00032
IF (NSUBDV .EQ. 1) GO TO 100	PRNTBC 00033
IF (.NOT.SUBD) GO TO 100	PRNTBC 00034
WRITE (NT6,6002) IXBW	PRNTBC 00035
100 CONTINUE	PRNTBC 00036
WRITE (NT6,6005) XMACH	PRNTBC 00037
C	PRNTBC 00038
WRITE (NT6,6010) (I, I=2,JLAST,2)	PRNTBC 00039
WRITE (NT6,6012)	PRNTBC 00040
DO 250 IROW = IFRST,ILAST	PRNTBC 00041
CALL DCODER(IBOX,LBXCD, IROW,1, IROW,JLAST, SUBD, ICCODE)	PRNTBC 00042
C	PRNTBC 00043
C CHANGE INTEGER CODES TO ALPHANUMERIC	PRNTBC 00044
C	PRNTBC 00045
DO 200 J= 1,JLAST	PRNTBC 00046
IF(ICODE(J).EQ.0) ICCODE(J) = BCD(1)	PRNTBC 00047
IF(ICODE(J).EQ.1) ICCODE(J) = BCD(2)	PRNTBC 00048
IF(ICODE(J).EQ.2) ICCODE(J) = BCD(3)	PRNTBC 00049


```
      IF(ICODE(J).EQ.3) ICODE(J) =BCD(4)
200 CONTINUE
      WRITE (NT6,6020) IROW, (ICODE(J), J = 1,JLAST )
230 CONTINUE
      RETURN
      END
```

```
PRNTBC 00050
PRNTBC 00051
PRNTBC 00052
PRNTBC 00053
PRNTBC 00054
PRNTBC 00055
```

C	SUBROUTINE BXCDPF(XLE,YLE,NLE,XTE,YTE,NTE, LSRONS,IBOX)	BXCDPF 00002
C		BXCDPF 00003
C	GENERATES THE BOX CODES FOR THE ON-PLANFORM BOXES OF ONE	BXCDPF 00004
C	SURFACE.	BXCDPF 00005
C	INPUT PARAMETERS	BXCDPF 00006
C	XLE = X-VALUES, LEADING EDGE, NON-DIMENSIONAL	BXCDPF 00007
C	YLE = Y-VALUES, LEADING EDGE	BXCDPF 00008
C	NLE = NUMBER OF LEADING EDGE POINTS	BXCDPF 00009
C	XTE = X-VALUES, TRAILING EDGE	BXCDPF 00010
C	YTE = Y-VALUES, TRAILING EDGE	BXCDPF 00011
C	NTE = NUMBER OF TRAILING EDGE POINTS	BXCDPF 00012
C	LSRONS = MAXIMUM NUMBER OF SUBDIVIDED ROWS ALLOWED	BXCDPF 00013
C		BXCDPF 00014
C	OUTPUT PARAMETERS	BXCDPF 00015
C		BXCDPF 00016
C	IBOX = COMPRESSED BOX CODES, SET 1 FOR PLANFORM BOXES,	BXCDPF 00017
C	UNCHANGED ELSEWHERE	BXCDPF 00018
C		BXCDPF 00019
C	COMMON /FILES / NT5,NT6,INTAPE,INFSP,MPLAIC,NSPAIC,NOUTP,	FILES 00002
C	1 IOUFSP,MODESC,IVFSC,IGEOSC,IWTFSC,IAICSC	FILES 00003
C	COMMON /GEOMTY/ COPLAN,NSUBDV,XSUBDV,NSUBD2,NSUBCN,NSURF,	GEOMTY 00002
C	1 B1,B1BETA,B1S,B1BTAS,WLAX,WLAZ,PSIW,	GEOMTY 00003
C	2 MGBW,MXBBW,MYBW,MYBBW,MXBSW,MYBSW,MYBBSW,	GEOMTY 00004
C	3 IXBW,XCENTR	GEOMTY 00005
C	LOGICAL COPLAN	GEOMTY 00006
C	COMMON /GEOM2 / TLAX,TLAZ,PSIT,MXBT,MYBT,MYBBT,MXBST,MYBST,	GEOM2 00002
C	1 MYBBST,IXBT,IXBST,CAPL	GEOM2 00003
C	COMMON /EDGES / FELOC(250), TELOC(250),JDIAG	EDGES 00002
C		BXCDPF 00024
C	LOGICAL WING	BXCDPF 00025
C	DIMENSION XLE(1),YLE(1),XTE(1),YTE(1)	BXCDPF 00026
C	DIMENSION IBOX(LSRONS,1)	BXCDPF 00027
C		BXCDPF 00028
C	INPUT COMMON PARAMETERS -	BXCDPF 00029
C	IXBT = SUBDIVIDED ROW OF FIRST UNSUBDIVIDED BOX ON TAIL	BXCDPF 00030
C	NSUBDV = (INTEGER) NUMBER OF SUBDIVISIONS	BXCDPF 00031
C	XSUBDV (REAL)	BXCDPF 00032
C	NSUB2 = NSUBDV/2	BXCDPF 00033
C	NSUBCN = NSUBDV/2 +1 , CENTER SUBDIVIDED BOX	BXCDPF 00034
C		BXCDPF 00035
C	IN/OUT COMMON PARAMETERS -	BXCDPF 00036
C	IXBW = 0, WING BEING DONE, CHANGED TO SUBDIVIDED ROW	BXCDPF 00037
C	OF FIRST UNSUBDIVIDED BOX CENTER ON WING	BXCDPF 00038
C	.NE. 0, TAIL BEING DONE, NOT CHANGED	BXCDPF 00039
C		BXCDPF 00040
C	OUTPUT COMMON PARAMETERS -	BXCDPF 00041
C	MGBSW = NUMBER OF SUBDIVIDED ROWS TO AFT END OF (WING)	BXCDPF 00042
C	MYBST (TAIL)	BXCDPF 00043
C	MYBSW = NUMBER OF SUBDIVIDED CHORDS ON THE (WING)	BXCDPF 00044
C	MYBST (TAIL)	BXCDPF 00045
C	MXBW = NUMBER OF UNSUBDIVIDED ROWS ON (WING)	BXCDPF 00046
C	MXBT (TAIL)	BXCDPF 00047
C	MYBW = NUMBER OF UNSUBDIVIDED CHORDS ON THE (WING)	BXCDPF 00048
C	MYBT (TAIL)	BXCDPF 00049
C	FELOC = ARRAY OF (LEADING) EDGE X-LOCATIONS, SUBDIVIDED	BXCDPF 00050
C	TELOC (TRAILING)	BXCDPF 00051
C		BXCDPF 00052

MYBS = 0	BXCDPF 00053
DEL = 1.0/XSUBDV	BXCDPF 00054
C LOCATION OF FIRST CHORD	BXCDPF 00055
YMIN = .5*(1.0 + DEL)	BXCDPF 00056
C SLOPES OF FIRST LEADING AND TRAILING EDGE SEGMENTS	BXCDPF 00057
XREFLE = XLE(1)	BXCDPF 00058
YREFLE = YLE(1)	BXCDPF 00059
XREFTE = XTE(1)	BXCDPF 00060
YREFTE = YTE(1)	BXCDPF 00061
DELE = (XLE(2)-XREFLE) / (YLE(2)-YREFLE)	BXCDPF 00062
DELTE = (XTE(2)-XREFTE) / (YTE(2)-YREFTE)	BXCDPF 00063
ILE = 2	BXCDPF 00064
ITE = 2	BXCDPF 00065
IERR = 0	BXCDPF 00066
C	BXCDPF 00067
C WAS THIS CALL FOR WING OR TAIL -	BXCDPF 00068
IF (IXBW .EQ. 0) GO TO 120	BXCDPF 00069
C SET UP COUNTERS FOR TAIL	BXCDPF 00070
SURF = 4HTAIL	BXCDPF 00071
WING = .F.	BXCDPF 00072
MYBT = IFIX(YLE(NLE))	BXCDPF 00073
NSCHRD = MYBT * NSUBDV	BXCDPF 00074
JEXLOC = MYBW * NSUBDV + 1	BXCDPF 00075
IXB = IXBST	BXCDPF 00076
LSRR = LSROWS	BXCDPF 00077
IF (.NOT. COPLAN) LSRR = LSRR + IXBST - 1	BXCDPF 00078
GO TO 130	BXCDPF 00079
C	BXCDPF 00080
C SET UP COUNTERS FOR WING	BXCDPF 00081
120 CONTINUE	BXCDPF 00082
SURF = 4HWING	BXCDPF 00083
WING = .T.	BXCDPF 00084
NSCHRD = MYBW * NSUBDV	BXCDPF 00085
JEXLOC = 1	BXCDPF 00086
XMIN = XREFLE + (YMIN-YREFLE)*DELE	BXCDPF 00087
IXBW = (1.-XMIN)*XSUBDV + 1	BXCDPF 00088
IXB = 1	BXCDPF 00089
LSRR = LSROWS	BXCDPF 00090
C	BXCDPF 00091
C START LOOP ON SUBDIVIDED CHORDS	BXCDPF 00092
130 CONTINUE	BXCDPF 00093
YCHORD = YMIN	BXCDPF 00094
DO 350 JCHRD = 1, NSCHRD	BXCDPF 00095
C FIND LEADING EDGE OF THIS CHORD	BXCDPF 00096
140 CONTINUE	BXCDPF 00097
C IS THE CURRENT L.E. SEGMENT STILL GOOD -	BXCDPF 00098
IF (YCHORD - YLE(ILE)) 160,170,150	BXCDPF 00099
C NO, ANOTHER SEGMENT IS NEEDED	BXCDPF 00100
150 CONTINUE	BXCDPF 00101
YREFLE = YLE(ILE)	BXCDPF 00102
ILE = ILE + 1	BXCDPF 00103
C	BXCDPF 00104
C CHECK FOR EXCEEDING LIMIT	BXCDPF 00105
IF (.LE. .GT. NLE) GO TO 710	BXCDPF 00106
C CHECK FOR EDGE SEGMENT PARALLEL TO (SKIP THE SEGMENT) OR	BXCDPF 00107
C CUTTING BACK TOWARD CENTER-LINE (ERROR)	BXCDPF 00108
IF (YREFLE - YLE(ILE)) 160,150,730	BXCDPF 00109
C SEGMENT HAS POSITIVE SLOPE	

160	XREFLE = XLE(ILE-1)	BXCDPF	00110
	DELE = (XLE(ILE)-XREFLE) / (YLE(ILE) - YREFLE)	BXCDPF	00111
	GO TO 140	BXCDPF	00112
C		BXCDPF	00113
	CHORD CENTER LIES ON SEGMENT ENDPOINT	BXCDPF	00114
170	CONTINUE	BXCDPF	00115
	FEXLOC(JEXLOC) = (XLE(ILE)-1.0)*XSUBDV + IXBW	BXCDPF	00116
	GO TO 190	BXCDPF	00117
C		BXCDPF	00118
	CHORD CENTER LIES WITHIN THE SEGMENT	BXCDPF	00119
180	CONTINUE	BXCDPF	00120
	FEXLOC(JEXLOC) = (XREFLE + DELE*(YCHORD-YREFLE) - 1.0) *XSUBDV	BXCDPF	00121
	1 + IXBW	BXCDPF	00122
C		BXCDPF	00123
190	CONTINUE	BXCDPF	00124
	FEXLOC(JEXLOC) = IFIX(FEXLOC(JEXLOC)) + .5	BXCDPF	00125
	ISTART = FEXLOC(JEXLOC) + 1	BXCDPF	00126
C		BXCDPF	00127
	THE FOLLOWING CODE FINDS THE TRAILING EDGE OF THIS CHORD IN	BXCDPF	00128
C	THE SAME MANNER AS ABOVE.	BXCDPF	00129
240	CONTINUE	BXCDPF	00130
	IF (YCHORD - YTE(ITE)) 280,270,250	BXCDPF	00131
250	CONTINUE	BXCDPF	00132
	YREFTE = YTE(ITE)	BXCDPF	00133
	ITE = ITE + 1	BXCDPF	00134
	IF (ITE .GT. NTE) GO TO 720	BXCDPF	00135
	IF (YREFTE - YTE(ITE)) 260,250,740	BXCDPF	00136
260	XREFTE = XTE(ITE-1)	BXCDPF	00137
	DELTE = (XTE(ITE)-XREFTE) / (YTE(ITE)-YREFTE)	BXCDPF	00138
	GO TO 240	BXCDPF	00139
270	CONTINUE	BXCDPF	00140
	TEXLOC(JEXLOC) = (XTE(ITE)-1.0)*XSUBDV + IXBW	BXCDPF	00141
	GO TO 290	BXCDPF	00142
280	CONTINUE	BXCDPF	00143
	TEXLOC(JEXLOC) = (XREFTE + DELTE*(YCHORD-YREFTE) - 1.0) *XSUBDV	BXCDPF	00144
	1 + IXBW	BXCDPF	00145
290	CONTINUE	BXCDPF	00146
	TEXLOC(JEXLOC) = IFIX(TEXLOC(JEXLOC)) + .5	BXCDPF	00147
	IEND = TEXLOC(JEXLOC)	BXCDPF	00148
	IF (IEND .GT. LSRR) GO TO 770	BXCDPF	00149
C		BXCDPF	00150
	SET BOX CODES TO 1 FOR PLANFORM BOXES OF THIS CHORD	BXCDPF	00151
300	CONTINUE	BXCDPF	00152
	CALL NCODER(IXBOX,LSROWS,ISTART, JCHRD, IEND, 1)	BXCDPF	00153
C		BXCDPF	00154
	MXBS = MAX0(MXBS,IEND)	BXCDPF	00155
	JEXLOC = JEXLOC+1	BXCDPF	00156
	YCHORD = YCHORD + DEL	BXCDPF	00157
350	CONTINUE	BXCDPF	00158
C	END OF LOOP ON CHORDS	BXCDPF	00159
C		BXCDPF	00160
	IF (MING) GO TO 360	BXCDPF	00161
	MXBS = MXBS	BXCDPF	00162
	MBST = (MXBS-IXBW)/NSUBDV + 1	BXCDPF	00163
	MYBST = NS(CHRD)	BXCDPF	00164
	GO TO 370	BXCDPF	00165
360	MX33W = M'BS	BXCDPF	00166

MXBW = (MXBS-IXBW)/NSUBDV + 1	BXCDFP 00167
MYBSW = NSCHRD	BXCDFP 00168
370 CONTINUE	BXCDFP 00169
IF (IERR .NE. 0) GO TO 750	BXCDFP 00170
C	BXCDFP 00171
RETURN	BXCDFP 00172
C	BXCDFP 00173
C ERROR DIAGNOSTICS	BXCDFP 00174
C	BXCDFP 00175
C LIMIT EXCEEDED	BXCDFP 00176
710 IERR = 1	BXCDFP 00177
EDGE = 8H LEADING	BXCDFP 00178
ISEC = ILE - 1	BXCDFP 00179
GO TO 750	BXCDFP 00180
720 IERR = 1	BXCDFP 00181
EDGE = 8H TRAILING	BXCDFP 00182
ISEC = ITE - 1	BXCDFP 00183
GO TO 750	BXCDFP 00184
C BAD EDGE DEFINITION	BXCDFP 00185
730 IERR = 2	BXCDFP 00186
EDGE = 8H LEADING	BXCDFP 00187
ISEC = ILE - 1	BXCDFP 00188
GO TO 750	BXCDFP 00189
740 IERR = 2	BXCDFP 00190
EDGE = 8H TRAILING	BXCDFP 00191
ISEC = ITE - 1	BXCDFP 00192
C	BXCDFP 00193
750 CONTINUE	BXCDFP 00194
WRITE(NT6,7500) SURF, EDGE	BXCDFP 00195
7500 FORMAT(22H0*** ERROR PROCESSING ,A5,10HGEOMETRY, ,A8, 9H EDGE ***)	BXCDFP 00196
GO TO (755,760,800) IERR	BXCDFP 00197
755 WRITE(NT6,7550) ISEC	BXCDFP 00198
7550 FORMAT(5X, 8HSECTION ,I2,24H IS BEYOND THOSE DEFINED)	BXCDFP 00199
GO TO 800	BXCDFP 00200
760 WRITE(NT6,7600) ISEC	BXCDFP 00201
7600 FORMAT(5X, 8HSECTION ,I2,36H OF THE EDGE DOUBLES BACK TOWARD THE	BXCDFP 00202
1 12H CENTER LINE)	BXCDFP 00203
GO TO 800	BXCDFP 00204
C PLANFORM EXCEEDS BOX PATTERN LIMIT	BXCDFP 00205
770 CONTINUE	BXCDFP 00206
IERR = 3	BXCDFP 00207
EDGE = 8H TRAILING	BXCDFP 00208
ISEC = ITE - 1	BXCDFP 00209
WRITE (NT6,7700) ISEC,SURF,JOHND,IEND	BXCDFP 00210
7700 FORMAT(20H *** ERROR - SECTION,I3,20H OF THE TRAILING EDGE OF THE	BXCDFP 00211
1 A4,14H CAUSES CHORD I3,14H TO GO TO ROW I3,15H, WHICH EXCEEDS	BXCDFP 00212
2 14H THE LIMIT ***)	BXCDFP 00213
IEND = LSRCHS	BXCDFP 00214
C GO BACK TO FINISH THE SURFACE, THEN PRINT PLANFORM AND FLUSH	BXCDFP 00215
GO TO 300	BXCDFP 00216
C	BXCDFP 00217
800 CA PRINTC(IXBW,LSRCHS, IXB,MXB, NSCHRD, .T.)	BXCDFP 00218
C	BXCDFP 00219
8000 CALL FLUSH(1)	BXCDFP 00220
C	BXCDFP 00221
END	BXCDFP 00222

	SUBROUTINE BXCDI (IWAKE,LSROWS,LSCHDS,IBOX)	BXCDI	00002
	DIMENSION IWAKE(1),IBOX(LSROWS,1)	BXCDI	00003
C		BXCDI	00004
C	DETERMINES BOX CODES FOR DIAPHRAGM REGIONS	BXCDI	00005
C		BXCDI	00006
C	IWAKE - ARRAY OF WAKE LIMITS, AS DICTATED BY A TAIL SURFACE	BXCDI	00007
C	LSROWS - ROW DIMENSION OF THE BOX CODE ARRAY	BXCDI	00008
C	LSCHDS - MAXIMUM NUMBER OF BOX CODES ALLOWED PER ROW	BXCDI	00009
C	IBOX - BOX CODE ARRAY, COMPRESSED TO 20 CODES PER WORD	BXCDI	00010
C		BXCDI	00011
	COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSFAIC,NOUTP,	FILES	00002
1	IOUFSP,MODESC,IVPSC,IGEO6C,IWTFSC,IAICSC	FILES	00003
	COMMON /GEOMTY/ COPLAN,NSUBDV,XSUBDV,NSUBD2,NSUBCN,NSURF,	GEOMTY	00002
1	B1,B1BETA,B1S,B1BTAS,WLAX,WLAZ,PSIW,	GEOMTY	00003
2	MXBW,MXBBW,MYBW,MYBBW,MXBSW,MYBSW,MYBBSW,	GEOMTY	00004
3	IXBW,XCENTR	GEOMTY	00005
	LOGICAL COPLAN	GEOMTY	00006
	COMMON /GEOM2 / TLAX,TLAZ,PSI, MXBT,MYBT,MYBBT,MXBST,MYBST,	GEOM2	00002
1	MYBBST,IXB,IXBST,CAPL	GEOM2	00003
	COMMON /SAMFLW/ ISMFLW,ICHORD(10),IBOXF(10),IBOXL(10),ZLOC(10)	SAMFLW	00002
C	COMMON INPUT VALUES-	BXCDI	00015
C	MXBBSW, MXBBST,IXBST, MYBSW,MYBST,MYBBSW,MYBBST, NSUBDV	BXCDI	00016
C	ISMFLW	BXCDI	00017
C	COMMON OUTPUT VALUES-	BXCDI	00018
C	MYBBSW, MYBBST, MYBBW,MYBBT	BXCDI	00019
	DIMENSION ICODE(160)	BXCDI	00020
	LOGICAL WING	BXCDI	00021
C		BXCDI	00022
C	DETERMINE WHETHER THIS IS A WING OR TAIL	BXCDI	00023
	IF (IWAKE(1) .EQ. 0) GO TO 80	BXCDI	00024
	WING = .T.	BXCDI	00025
	IXBS = 1	BXCDI	00026
	IXBS1 = 2	BXCDI	00027
	MXBBS = MXBBW+XSUBDV + IXBW - NSUBCN	BXCDI	00028
	IF (COPLAN) MXBBS = MXBST	BXCDI	00029
	JEXLOC = 1	BXCDI	00030
	MYBBS = MYBBSW	BXCDI	00031
	MYBS = MYBSW	BXCDI	00032
	GO TO 100	BXCDI	00033
C	THIS IS A TAIL SURFACE	BXCDI	00034
	80 WING = .F.	BXCDI	00035
	IXBS = IXBST	BXCDI	00036
	IXBS1 = IXBS + 1	BXCDI	00037
	MXBBS = MXBST	BXCDI	00038
	JEXLOC = MYBSW + 1	BXCDI	00039
	MYBBS = MYBBST	BXCDI	00040
	MYBS = MYBST	BXCDI	00041
	100 CONTINUE	BXCDI	00042
	MXBBS1 = MXBBS-1	BXCDI	00043
C		BXCDI	00044
C	DETERMINE LEADING EDGE DIAPHRAGM	BXCDI	00045
	DO 130 J = 2,MYBS	BXCDI	00046
	CALL DCODER (IBOX,LSROWS, IXBS,J-1, IXBS1,J-1, .T., ICODE(2))	BCSGEB	00001
	DO 120 I = IXBS1,MXBBS1	BXCDI	00048
	CALL DCODER (IBOX,LSROWS, I,J, I,J, .T., ICODE)	BXCDI	00049
	ICODE(1) = ICODE(2)	BXCDI	00050
	ICODE(2) = ICODE(3)	BXCDI	00051

CALL DCDER(1BOX,LSROWS, I+1,J-1, I+1,J-1, .T., ICODE(3))	BXCDI	00052
IF (ICOD .NE. 0) GO TO 120	BXCDI	00053
IF (ICOD(1) .EQ. 0 .OR. ICODE(3) .EQ. 0) GO TO 120	BXCDI	00054
CALL NCDER(1BOX,LSROWS, I,J, I, 2)	BXCDI	00055
120 CONTINUE	BXCDI	00056
130 CONTINUE	BXCDI	00057
C END OF DOUBLE LOOP TO DETERMINE LEADING EDGE DIAPHRAGM AREAS	BXCDI	00058
C	BXCDI	00059
C DETERMINE TRAILING EDGE (WAKE) DIAPHRAGM	BXCDI	00060
IWK = 0	BXCDI	00061
DO 180 J = 1,MYBS	BXCDI	00062
IF (WING) IWK = IWAKE(J)	BXCDI	00063
CALL DCDER(1BOX,LSROWS, IXBS,J, IXBS,J, .T., ICD)	BXCDI	00064
DO 170 I = IXBS1,MXBBS	BXCDI	00065
ICODM1 = ICD	BXCDI	00066
CALL DCDER(1BOX,LSROWS, I,J, I,J, .T., ICD)	BXCDI	00067
IF (ICOD .EQ. 1) GO TO 170	BXCDI	00068
IF (ICODM1 .NE. 1 .AND. ICDM1 .NE. 3) GO TO 170	BXCDI	00069
IF (I .LE. IWK) GO TO 180	BXCDI	00070
IF (I .EQ. MXBBS) GO TO 180	BXCDI	00071
C THE BOX IS A CANDIDATE. SEARCH DIAGONALLY FOR POSSIBLE	BXCDI	00072
C RECEIVING BOXES DOWNSTREAM.	BXCDI	00073
JP = J	BXCDI	00074
JM = J	BXCDI	00075
IS = I+1	BXCDI	00076
DO 190 II = IS,MXBBS	BXCDI	00077
IF (JM .GT. 1) JM = JM - 1	BXCDI	00078
CALL DCDER(1BOX,LSROWS, II,JM, II,JM, .T., ICD)	BXCDI	00079
IF (ICOD .NE. 0) GO TO 180	BXCDI	00080
IF (JP .GE. MYBBS) GO TO 190	BXCDI	00081
JP = JP + 1	BXCDI	00082
CALL DCDER(1BOX,LSROWS, II,JP, II,JP, .T., ICD)	BXCDI	00083
IF (ICOD .NE. 0) GO TO 180	BXCDI	00084
IF (.NOT. WING) GO TO 190	BXCDI	00085
IF (II .LE. IWAKE(JM)) GO TO 180	BXCDI	00086
IF (II .LE. IWAKE(JP)) GO TO 180	BXCDI	00087
190 CONTINUE	BXCDI	00088
C END OF LOOP ON DIAGONAL SEARCH	BXCDI	00089
GO TO 170	BXCDI	00090
C	BXCDI	00091
C CONDITIONS HAVE BEEN FOUND FOR A VALID WAKE BOX	BXCDI	00092
180 CONTINUE	BXCDI	00093
CALL NCDER(1BOX,LSROWS, I,J, I, 3)	BXCDI	00094
ICOD = 3	BXCDI	00095
170 CONTINUE	BXCDI	00096
C END OF LOOP ON ROWS, AND	BXCDI	00097
160 CONTINUE	BXCDI	00098
C END OF LOOP ON CHORDS, FOR WAKE DIAPHRAGM, FROM 130+	BXCDI	00099
C	BXCDI	00100
C DETERMINE THE TIP DIAPHRAGM REGION	BXCDI	00101
LBB = 2	BXCDI	00102
DO 300 I = IXBS1,MXBBS1	BXCDI	00103
C SEARCH FOR LAST NON-ZERO BOX CODE ON THE ROW, FROM LBB OUTWARD	BXCDI	00104
CALL DCDER(1BOX,LSROWS, I,LBB, I,MYBBS, .T., ICODE(LBB))	BXCDI	00105
200 LBB = LBB+1	BXCDI	00106
DO 2 J = LBBP1,MYBBS	BXCDI	00107
IF (ICOD(J) .EQ. 0) GO TO 220	BXCDI	00108

210 CONTINUE	BXCDI 00109
LBB = MYBBS	BXCDI 00110
GO TO 225	BXCDI 00111
220 LBB = J - 1	BXCDI 00112
C LBB = THE SUBSCRIPT OF THE LAST NON-ZERO BOX ON THE ROW	BXCDI 00113
C TEST BOX DIRECTLY AHEAD FOR NON-ZERO VALUE	BXCDI 00114
225 CONTINUE	BXCDI 00115
CALL DCDER(IXOX,LSROWS, I-1,LBB, I-1,LBB, .T., ICOD)	BXCDI 00116
IF (ICOD .EQ. 0) GO TO 280	BXCDI 00117
C SEARCH FOR A NON-ZERO CODE ON THE INBOARD DIAGONAL AFT	BXCDI 00118
J = LBB	BXCDI 00119
IP1 = I + 1	BXCDI 00120
IF (IP1 .GT. MXBBS) GO TO 310	BXCDI 00121
DO 230 II = IP1, MXBBS	BXCDI 00122
CALL DCDER(IXOX,LSROWS, II,J, II,J, .T., ICOD)	BXCDI 00123
IF (ICOD .NE. 0) GO TO 255	BXCDI 00124
J = J - 1	BXCDI 00125
230 CONTINUE	BXCDI 00126
C NO DIAGONAL BOX WAS FOUND	BXCDI 00127
C IF (NBURF .EQ. 1 .AND. ISMPLW .EQ. 0) GO TO 310	BXCDI 00128
IF (.NOT. WING) GO TO 310	BXCDI 00129
C SEARCH BACK ALONG THE DIAGONAL FOR A DIAPHRAGM REGION	BXCDI 00130
C CAUSED BY A TAIL SURFACE	BXCDI 00131
II = MXBBS + 1	BXCDI 00132
DO 240 III = IP1, MXBBS	BXCDI 00133
II = II - 1	BXCDI 00134
J = J + 1	BXCDI 00135
IF (IWAKE(J) .GE. II) GO TO 250	BXCDI 00136
240 CONTINUE	BXCDI 00137
C NO DIAPHRAGM FOUND	BXCDI 00138
GO TO 310	BXCDI 00139
C CONDITION FOUND REQUIRING DIAPHRAGM BOXES ON THE DIAGONAL.	BXCDI 00140
250 JJ = J	BXCDI 00141
GO TO 260	BXCDI 00142
255 JJ = J + 1	BXCDI 00143
II = II - 1	BXCDI 00144
260 CONTINUE	BXCDI 00145
C TEST FOR EXCEEDING BOX CODE ARRAY	BXCDI 00146
LBB = LBB + 1	BXCDI 00147
IF (LBB .GT. LSCHDS) GO TO 8500	BXCDI 00148
C SET DIAGONAL ELEMENTS	BXCDI 00149
DO 270 J = JJ, LBB	BXCDI 00150
CALL MCDER(IXOX,LSROWS, II,J, II, 2)	BXCDI 00151
II = II - 1	BXCDI 00152
270 CONTINUE	BXCDI 00153
ICODE(LBB) = 2	BXCDI 00154
MYBBS = MAX0(MYBBS, LBB)	BXCDI 00155
GO TO 200	BXCDI 00156
C DETERMINE LAST NON-ZERO BOX ON NEXT ROW	BXCDI 00157
280 CONTINUE	BXCDI 00158
MYBBS = MAX0(MYBBS, LBB)	BXCDI 00159
II = I + 1	BXCDI 00160
DO 290 K = 1, LBB	BXCDI 00161
J = LBB - K + 1	BXCDI 00162
	BXCDI 00163
	BXCDI 00164
	BXCDI 00165

CALL DCODER(IXBOX,LSROWS, II,J, II,J, .T., ICOD)	BXCDI	00166
IF (ICOD .NE. 0) GO TO 295	BXCDI	00167
290 CONTINUE	BXCDI	00168
GO TO 300	BXCDI	00169
295 LBB = J	BXCDI	00170
300 CONTINUE	BXCDI	00171
C END OF LOOP ON ROWS DETERMINING TIP DIAPHRAGM CODES, FROM 180*	BXCDI	00172
C	BXCDI	00173
310 CONTINUE	BXCDI	00174
IF (.NENG) GO TO 350	BXCDI	00175
MYBBST = MYBBS	BXCDI	00176
MYBBT = (MYBBS+NSUBD2)/NSUBDV	BXCDI	00177
GO TO 500	BXCDI	00178
350 MYBBSW = MYBBS	BXCDI	00179
MYBBW = (MYBBS+NSUBD2)/NSUBDV	BXCDI	00180
IF (.NOT. CORLAN) GO TO 500	BXCDI	00181
MYBBST = MYBBSW	BXCDI	00182
MYBBT = MYBBW	BXCDI	00183
C	BXCDI	00184
500 RETURN	BXCDI	00185
C	BXCDI	00186
8500 WRITE (NT6,9500)	BXCDI	00187
9500 FORMAT(51H0*** ERROR - TOO MANY CHORDS FOR BOX CODE ARRAY ***)	BXCDI	00188
CALL PRNTBC(IXBOX,LSROWS,IXBS,MXBBS,MYBBS, .T.)	BXCDI	00189
CALL FLUSH(1)	BXCDI	00190
C	BXCDI	00191
END	BXCDI	00192

	SUBROUTINE GMAREA (IBOX, LBXCD, WING, ALPHA, IJALPH, NALPH)	GMAREA 00002
C		GMAREA 00003
C	IBOX = ARRAY OF BOX TYPE CODES	GMAREA 00004
C	LBXCD = ROW DIMENSION OF BOX CODE ARRAY	GMAREA 00005
C	WING = PLANFORM INDICATOR	GMAREA 00006
C	ALPHA = ARRAY OF ALPHAS (NORMALIZED AREAS)	GMAREA 00007
C	IJALPH = SUBSCRIPTS FOR IBOX ARRAY OF CELLS THAT HAVE ALPHAS	GMAREA 00008
C	NOT EQUAL TO 0.0 OR 1.0	GMAREA 00009
C	NALPH = NUMBER OF ALPHAS STORED	GMAREA 00010
C		GMAREA 00011
	COMMON /GEOMTY/ CORPLAN, NSUBDV, XSUBDV, NSUBD2, NSUBCN, NSURF,	GEOMTY 00002
1	B1, B1BETA, B1S, B1BTAS, WLAX, WLAZ, PSIW,	GEOMTY 00003
2	MXBW, MXBBW, MYBW, MYBBW, MXBSW, MYBSW, MYBBSW,	GEOMTY 00004
3	IXBW, XCENR	GEOMTY 00005
	LOGICAL CORPLAN	GEOMTY 00006
	COMMON /GEOM2/ TLAX, TLAZ, PSI1, MXBT, MYBT, MYBBT, MXBST, MYBST,	GEOM2 00002
1	MYBBST, IXBT, IXBST, CARL	GEOM2 00003
	COMMON /PLANDY/ NMLE, NMTE, NTLE, NTTE, XMLE(10), YMLE(10),	PLANDY 00002
1	XWTE(10), YWTE(10), XTLE(10), YTLE(10),	PLANDY 00003
2	XTTE(10), YTTE(10)	PLANDY 00004
	COMMON /EDGES/ FEXLOC(250), TEXLOC(250), JDIAG	EDGES 00002
C		GMAREA 00016
C	COMMON PARAMETERS USED	GMAREA 00017
C	MXB = LENGTH OF BOX PATTERN (X-DIRECTION)	GMAREA 00018
C	MYB = MAXIMUM ON-PLANFORM SPAN (Y-DIRECTION)	GMAREA 00019
C	CORPLAN = .T., SECOND SURFACE EXISTS FOR PLANFORM	GMAREA 00020
C	= .F., SINGLE SURFACE	GMAREA 00021
C	NMLE = NUMBER OF POINTS DEFINING LEADING EDGE OF THE WING	GMAREA 00022
C	NMTE = NUMBER OF POINTS DEFINING TRAILING EDGE OF THE WING	GMAREA 00023
C	NTLE = NUMBER OF POINTS DEFINING LEADING EDGE OF THE TAIL	GMAREA 00024
C	NTTE = NUMBER OF POINTS DEFINING TRAILING EDGE OF THE TAIL	GMAREA 00025
C	XMLE = X COORDINATE OF THE LEADING EDGE DEFINITION POINT	GMAREA 00026
C	FOR THE FIRST PLANFORM	GMAREA 00027
C	YMLE = Y COORDINATE OF THE LEADING EDGE DEFINITION POINT	GMAREA 00028
C	FOR THE FIRST PLANFORM	GMAREA 00029
C	XWTE = X COORDINATE OF THE TRAILING EDGE DEFINITION POINT	GMAREA 00030
C	FOR THE FIRST PLANFORM	GMAREA 00031
C	YWTE = Y COORDINATE OF THE TRAILING EDGE DEFINITION POINT	GMAREA 00032
C	FOR THE FIRST PLANFORM	GMAREA 00033
C	XTLE = X COORDINATE OF THE LEADING EDGE DEFINITION POINT	GMAREA 00034
C	FOR THE SECOND PLANFORM	GMAREA 00035
C	YTLE = Y COORDINATE OF THE LEADING EDGE DEFINITION POINT	GMAREA 00036
C	FOR THE SECOND PLANFORM	GMAREA 00037
C	XTTE = X COORDINATE OF THE TRAILING EDGE DEFINITION POINT	GMAREA 00038
C	FOR THE SECOND PLANFORM	GMAREA 00039
C	YTTE = Y COORDINATE OF THE TRAILING EDGE DEFINITION POINT	GMAREA 00040
C	FOR THE SECOND PLANFORM	GMAREA 00041
C		GMAREA 00042
	COMMON /LAREA / LEFT, RIGHT, ICODE	LAREA 00002
	DIMENSION IBOX(50)	GMAREA 00044
	DIMENSION ALPHA(1), IJALPH(1)	GMAREA 00045
	LOGICAL WING	GMAREA 00046
	REAL LINE, LINE2, LINE3, LIM, LEFT	GMAREA 00047
	NALPH = 1	GMAREA 00048
	IF (WING) GO TO 5	GMAREA 00049
	MYB = MYBT	GMAREA 00050
	IXB = (IXBT-IXBW)/NSUBDV + 1	GMAREA 00051

MXB = MXBT	GMAREA 00052
CEN2 = 0.	GMAREA 00053
GO TO 8	GMAREA 00054
5 MYB = MYBW	GMAREA 00055
IF (COPLAN) MYB = MAX0(MYB,MYBT)	GMAREA 00056
IXB = 1	GMAREA 00057
MXB = MXBW	GMAREA 00058
IF (COPLAN) MXB = MXBT	GMAREA 00059
8 CONTINUE	GMAREA 00060
C	GMAREA 00061
C LOOP ON CHORDS	GMAREA 00062
DO 90 J=1,MYB	GMAREA 00063
YJ = J	GMAREA 00064
RIGHT = YJ + 0.5	GMAREA 00065
LEFT = YJ - 0.5	GMAREA 00066
IF (.NOT. WING) GO TO 110	GMAREA 00067
C	GMAREA 00068
C CALL NTRCEP TO DETERMINE LEFT, RIGHT AND CENTER LINE	GMAREA 00069
C INTERCEPTS, AND THE BREAK POINTS OF EDGES OVER THIS	GMAREA 00070
C CHORD.	GMAREA 00071
C	GMAREA 00072
IF (YJ .GT. YWLE(NWLE)) GO TO 10	GMAREA 00073
CALL NTRCEP(J, YWLE,XWLE, LIN1,CEN1,RIN1,NBK1,KINK1, 1)	GMAREA 00074
CALL NTRCEP(J, YWTE,XWTE, LIN2,CEN2,RIN2,NBK2,KINK2, 2)	GMAREA 00075
10 IF (NSURF .EQ. 1 .OR. .NOT. COPLAN) GO TO 20	GMAREA 00076
C COMPUTE SLOPE AND INTERCEPTS FOR SECOND PLANFORM.	GMAREA 00077
110 IF (YJ .GT. YTLE(NTLE)) GO TO 20	GMAREA 00078
CALL NTRCEP(J, YTLE,XTLE, LIN3,CEN3,RIN3,NBK3,KINK3, 1)	GMAREA 00079
CALL NTRCEP(J, YTTE,XTTE, LIN4,CEN4,RIN4,NBK4,KINK4, 2)	GMAREA 00080
20 CONTINUE	GMAREA 00081
C SLOPE AND INTERVALS COMPLETED.	GMAREA 00082
C	GMAREA 00083
C LOOP DOWN THE CHORD	GMAREA 00084
CALL DCODER(IBOX,LBXCD, IXB,J, MXB,J, .F., IBX)	GMAREA 00085
II = 1	GMAREA 00086
DO 85 I=IXB,MXB	GMAREA 00087
XI = I	GMAREA 00088
IF (IBX(II) .NE. 1) GO TO 80	GMAREA 00089
BOKLE = XI - 0.5	GMAREA 00090
BOKTE = XI + 0.5	GMAREA 00091
IF (.NOT. WING) GO TO 40	GMAREA 00092
IF (YJ .GT. YWLE(NWLE)) GO TO 40	GMAREA 00093
IF (XI .GT. CEN2) GO TO 40	GMAREA 00094
C BOX IS ON PLANFORM 1	GMAREA 00095
ICODE = 1	GMAREA 00096
C ICODE = 1, 1ST L.E. BOX ON CHORD	GMAREA 00097
C = 2, LAST T.E. BOX ON CHORD	GMAREA 00098
C = 3, INTERNAL CUT BOX	GMAREA 00099
IF (II .EQ. 1) GO TO 24	GMAREA 00100
IF (IBX(II-1) .NE. 1) GO TO 24	GMAREA 00101
ICODE = 2	GMAREA 00102
IF (I .EQ. MXB) GO TO 24	GMAREA 00103
IF ((II+1) .NE. 1) GO TO 24	GMAREA 00104
IF (I-1 .GT. CEN2) GO TO 24	GMAREA 00105
ICODE = 3	GMAREA 00106
IU=0	GMAREA 00107
IB=0	GMAREA 00108

IF(RIN3.GT.BOXLE) IU=1	GMAREA 00109
IF(RIN2.LT.BOXTE) IB=1	GMAREA 00110
IF(LIN2.LT.BOXTE) IB=1	GMAREA 00111
IF (IU+IB.EQ.0) GO TO 80	GMAREA 00112
C BOX IS NOT ENTIRELY ON PLANFORM. COMPUTE AREA.	GMAREA 00113
24 CONTINUE	GMAREA 00114
CALL ALPHAC(XI, XWLE,YWLE,XWTE,YWTE,	GMAREA 00115
1LIN3,CEN3,RIN3,NBK3,KINK3,LIN2,CEN2,RIN2,NBK2,KINK2,ALPHA(NALPH))	GMAREA 00116
GO TO 75	GMAREA 00117
C	GMAREA 00118
40 CONTINUE	GMAREA 00119
IF (NSURF .EQ. 1) GO TO 80	GMAREA 00120
C BOX IS ON PLANFORM 2.	GMAREA 00121
ICCODE =2	GMAREA 00122
IF(I.EQ.MXB) GO TO 44	GMAREA 00123
ICCODE = 1	GMAREA 00124
IF(XI-1. .LT.CEN3) GO TO 44	GMAREA 00125
IF(1BX(II-1) .NE. 1) GO TO 44	GMAREA 00126
ICCODE =2	GMAREA 00127
IF(1BX(II+1) .NE. 1) GO TO 44	GMAREA 00128
ICCODE =3	GMAREA 00129
IU=0	GMAREA 00130
IB=0	GMAREA 00131
IF(RIN3.GT.BOXLE) IU=1	GMAREA 00132
IF(RIN4.LT.BOXTE) IB=1	GMAREA 00133
IF(LIN4.LT.BOXTE) IB=1	GMAREA 00134
IF (IU+IB.EQ.0) GO TO 80	GMAREA 00135
C BOX IS NOT ENTIRELY ON PLANFORM. COMPUTE AREA.	GMAREA 00136
44 CONTINUE	GMAREA 00137
CALL ALPHAC(XI, XTLE,YTLE,XTTE,YTTE,	GMAREA 00138
1LIN3,CEN3,RIN3,NBK3,KINK3,LIN4,CEN4,RIN4,NBK4,KINK4,ALPHA(NALPH))	GMAREA 00139
75 IJALPH(NALPH) = J*512 + I	GMAREA 00140
NALPH = NALPH + 1	GMAREA 00141
80 CONTINUE	GMAREA 00142
II = II + 1	GMAREA 00143
85 CONTINUE	GMAREA 00144
90 CONTINUE	GMAREA 00145
NALPH = NALPH -1	GMAREA 00146
RETURN	GMAREA 00147
END	GMAREA 00148

SUBROUTINE ALPHAC(X,XLED,YLED,XTED,YTED,	ALPHAC	00002
1 L1,C1,R1,NBK1,K1,L2,C2,R2,NBK2,K2,AREA)	ALPHAC	00003
C	ALPHAC	00004
C X = X COORDINATE OF CELL CENTER	ALPHAC	00005
C L1 = X COORDINATE OF LEADING EDGE LEFT SIDE INTERSECTION	ALPHAC	00006
C C1 = X COORDINATE OF LEADING EDGE CENTER LINE INTERSECTION	ALPHAC	00007
C R1 = X COORDINATE OF LEADING EDGE RIGHT SIDE INTERSECTION	ALPHAC	00008
C K1 = FLAG TO INDICATE LEADING EDGE KINK	ALPHAC	00009
C L2 = X COORDINATE OF TRAILING EDGE LEFT SIDE INTERSECTION	ALPHAC	00010
C C2 = X COORDINATE OF TRAILING EDGE CENTER LINE INTERSECT.	ALPHAC	00011
C R2 = X COORDINATE OF TRAILING EDGE RIGHT SIDE INTESECTION	ALPHAC	00012
C K2 = FLAG TO INDICATE TRAILING EDGE KINK	ALPHAC	00013
C AREA = AREA COMPUTED FOR THE CELL	ALPHAC	00014
C	ALPHAC	00015
COMMON /LAREA / LEFT,RIGHT,ICODE	LAREA	00002
C LEFT = Y COORDINATE OF LEFT SIDE OF CHORD	ALPHAC	00017
C RIGHT = Y COORDINATE OF RIGHT SIDE OF CHORD	ALPHAC	00018
C ICODE = 1, 1ST L.E. BOX ON CHORD	ALPHAC	00019
C = 2, LAST T.E. BOX ON CHORD	ALPHAC	00020
C = 3, INTERNAL CUT BOX	ALPHAC	00021
C	ALPHAC	00022
DIMENSION XC(6), YC(6)	ALPHAC	00023
DIMENSION XLED(1),YLED(1),XTED(1),YTED(1)	ALPHAC	00024
REAL LEFT, L1, L2	ALPHAC	00025
EPS = 1.0E-04	ALPHAC	00026
BOXLE = X-0.5	ALPHAC	00027
BOXTE = X + 0.5	ALPHAC	00028
XU = X - 1.0	ALPHAC	00029
XL = X + 1.0	ALPHAC	00030
AREA = 0.0	ALPHAC	00031
ISLICE = 0	ALPHAC	00032
IF (ICODE.EQ.3) GO TO 5000	ALPHAC	00033
IF(C1.GT.XU.AND.C2.LT.XL) GO TO 3000	ALPHAC	00034
1110 IF(ICODE.EQ.1) GO TO 1000	ALPHAC	00035
1120 IF(ICODE.EQ.2) GO TO 2000	ALPHAC	00036
GO TO 4000	ALPHAC	00037
C	ALPHAC	00038
C LEADING EDGE BOX	ALPHAC	00039
1000 CONTINUE	ALPHAC	00040
NTRAPS = NBK1 + 1	ALPHAC	00041
NTM1 = NTRAPS - 1	ALPHAC	00042
NXC = NTRAPS + 1	ALPHAC	00043
XC(1) = L1	ALPHAC	00044
YC(1) = LEFT	ALPHAC	00045
XC(NXC) = R1	ALPHAC	00046
YC(NXC) = RIGHT	ALPHAC	00047
IF (NTRAPS.EQ.1) GO TO 110	ALPHAC	00048
DO 100 NA=2,NTRAPS	ALPHAC	00049
KIDX = K1+NA-2	ALPHAC	00050
XC(NA) = XLED(KIDX)	ALPHAC	00051
YC(NA) = YLED(KIDX)	ALPHAC	00052
100 CONTINUE	ALPHAC	00053
110 CONTINUE	ALPHAC	00054
DO 300 NX = 1,NTRAPS	ALPHAC	00055
IF(XC(NX).GE.BOXTE) GO TO 300	ALPHAC	00056
A = BOXTE - XC(NX)	ALPHAC	00057
DY = YC(NX+1) - YC(NX)	ALPHAC	00058

IF(DY.LT.EPS) GO TO 300	ALPHAC 00059
IF(XC(NX+1).GT.BOXTE) GO TO 250	ALPHAC 00060
B = BOXTE - XC(NX+1)	ALPHAC 00061
200 AREA = AREA + 0.5*(A+B)*DY	ALPHAC 00062
GO TO 300	ALPHAC 00063
C	ALPHAC 00064
C EDGE CROSSES BOXTE. COMPUTE INTERSECTION FOR DY	ALPHAC 00065
250 CONTINUE	ALPHAC 00066
B = 0.0	ALPHAC 00067
DX = XC(NX+1) - XC(NX)	ALPHAC 00068
S = DY/DX	ALPHAC 00069
DY = S*A	ALPHAC 00070
GO TO 200	ALPHAC 00071
300 CONTINUE	ALPHAC 00072
IF (BOXTE-BOXLE.LT.EPS) GO TO 2000	ALPHAC 00073
IF(R2.LT.BOXTE) ISLICE =2	ALPHAC 00074
IF(L2.LT.BOXTE) ISLICE =1	ALPHAC 00075
IF(ISLICE.NE.0) GO TO 5000	ALPHAC 00076
GO TO 4000	ALPHAC 00077
C	ALPHAC 00078
C TRAILING EDGE BOX	ALPHAC 00079
2000 CONTINUE	ALPHAC 00080
NTRAPS =NBK2 + 1	ALPHAC 00081
NTM1 = NTRAPS- 1	ALPHAC 00082
NXC = NTRAPS+ 1	ALPHAC 00083
XC(1)= L2	ALPHAC 00084
YC(1)= LEFT	ALPHAC 00085
XC(NXC)=R2	ALPHAC 00086
YC(NXC)=RIGHT	ALPHAC 00087
IF(NTRAPS.EQ.1) GO TO 2110	ALPHAC 00088
DO 2100 NA = 2,NTRAPS	ALPHAC 00089
KIDX = K2 + NA -2	ALPHAC 00090
XC(NA) = XTED(KIDX)	ALPHAC 00091
YC(NA) = YTED(KIDX)	ALPHAC 00092
2100 CONTINUE	ALPHAC 00093
2110 CONTINUE	ALPHAC 00094
DO 2300 NX=1,NTRAPS	ALPHAC 00095
IF(XC(NX).LT.BOXLE.AND.XC(NX+1).LT.BOXLE) GO TO 2300	ALPHAC 00096
DY = YC(NX+1) - YC(NX)	ALPHAC 00097
IF(DY.LT.EPS) GO TO 2300	ALPHAC 00098
IF(XC(NX).LT.BOXLE.OR.XC(NX+1).LT.BOXLE) GO TO 2250	ALPHAC 00099
C	ALPHAC 00100
C DOES NOT INTERSECT BOXLE	ALPHAC 00101
A = XC(NX) - BOXLE	ALPHAC 00102
B = XC(NX+1) - BOXLE	ALPHAC 00103
2225 AREA = AREA + 0.5*(A+B)*DY	ALPHAC 00104
GO TO 2300	ALPHAC 00105
C	ALPHAC 00106
C INTERSECTS BOXLE	ALPHAC 00107
2250 CONTINUE	ALPHAC 00108
DX = XC(NX+1) -XC(NX)	ALPHAC 00109
S = DY/DX	ALPHAC 00110
A = BOXLE - XC(NX)	ALPHAC 00111
YINT = YC(NX) + S*A	ALPHAC 00112
IF(S.LT.0.0) GO TO 2275	ALPHAC 00113
C	ALPHAC 00114
C SLOPE POSITIVE	ALPHAC 00115

```

      A = 0
      B = XC(NX+1) - BOXLE
      DY = YC(NX+1) - YINT
      GO TO 2225
C
C      SLOPE NEGATIVE
2275 CONTINUE
      A = -A
      B = 0
      DY = YINT - YC(NX)
      GO TO 2225
2300 CONTINUE
      IF(R1.LT.BOXLE) ISLICE = 3
      IF(ISLICE.NE.0) GO TO 5000
      GO TO 4000
C
C      CHORD HAS ONLY 1 BOX
3000 CONTINUE
      BOXLE = R1
      BOXTE = R1
      GO TO 1000
C
C
C      THE FOLLOWING IS THE CALCULATIONS FOR A BOX WITH FORE AND
C      AFT BOXES ON PLANFORM. ONE OF 3 CORNER (L.L.,L.R., OR U.B.)
C      IS CUT OFF.
C
5000 CONTINUE
      IF(ISLICE.NE.0) GO TO 5005
      AREA = (RIGHT-LEFT)
5005 CONTINUE
      TA = 0.0
      IF(ISLICE.EQ.3) GO TO 5020
      IF(L2.LT.BOXTE) GO TO 5100
5010 IF(R2.LT.BOXTE) GO TO 5102
5020 CONTINUE
      IF(ISLICE.EQ.1.OR.ISLICE.EQ.2) GO TO 5400
      IF(R1.GT.BOXLE) GO TO 5300
      GO TO 5400
5100 ITAG = 1
      I = 1
      XC(I) = L2
      YC(I) = LEFT
      GO TO 5110
5102 ITAG = 2
      I = 1
      XC(I) = R2
      YC(I) = RIGHT
5110 I = I + 1
      IF(NBK2.EQ.0) GO TO 5150
      KIDX = K2 + I - 2
      IF(I .GT. 2) KIDX = K2 + NBK2 - I + 2
      IF(XTED(KIDX).GT.BOXTE) GO TO 5125
      XC(I) = XTED(KIDX)
      YC(I) = YTED(KIDX)
      GO TO 5110
5125 CONTINUE

```

```

ALPHAC 00116
ALPHAC 00117
ALPHAC 00118
ALPHAC 00119
ALPHAC 00120
ALPHAC 00121
ALPHAC 00122
ALPHAC 00123
ALPHAC 00124
ALPHAC 00125
ALPHAC 00126
ALPHAC 00127
ALPHAC 00128
ALPHAC 00129
ALPHAC 00130
ALPHAC 00131
ALPHAC 00132
ALPHAC 00133
ALPHAC 00134
ALPHAC 00135
ALPHAC 00136
ALPHAC 00137
ALPHAC 00138
ALPHAC 00139
ALPHAC 00140
ALPHAC 00141
ALPHAC 00142
ALPHAC 00143
ALPHAC 00144
ALPHAC 00145
ALPHAC 00146
ALPHAC 00147
ALPHAC 00148
ALPHAC 00149
ALPHAC 00150
ALPHAC 00151
ALPHAC 00152
ALPHAC 00153
ALPHAC 00154
ALPHAC 00155
ALPHAC 00156
ALPHAC 00157
ALPHAC 00158
ALPHAC 00159
ALPHAC 00160
ALPHAC 00161
ALPHAC 00162
ALPHAC 00163
ALPHAC 00164
ALPHAC 00165
ALPHAC 00166
ALPHAC 00167
ALPHAC 00168
ALPHAC 00169
ALPHAC 00170
ALPHAC 00171
ALPHAC 00172

```

IF(ITAG.EQ.2) KIDX=KIDX+1	ALPHAC 00173
XC(I) = BOXTE	ALPHAC 00174
DX = XTED(KIDX) - XTED(KIDX-1)	ALPHAC 00175
DY = YTED(KIDX) - YTED(KIDX-1)	ALPHAC 00176
IF(ITAG.EQ.2) GO TO 5130	ALPHAC 00177
C	ALPHAC 00178
C LOWER LEFT CORNER	ALPHAC 00179
DIST = BOXTE - XC(I-1)	ALPHAC 00180
YC(I) = YTED(KIDX-1) + DIST*(DY/DX)	ALPHAC 00181
GO TO 5160	ALPHAC 00182
C	ALPHAC 00183
C LOWER RIGHT CORNER	ALPHAC 00184
5130 DIST = BOXTE - XC(I-1)	ALPHAC 00185
YC(I) = YC(I-1) + DIST*(DY/DX)	ALPHAC 00186
GO TO 5160	ALPHAC 00187
5150 CONTINUE	ALPHAC 00188
XC(I) = BOXTE	ALPHAC 00189
DX = R2 - L2	ALPHAC 00190
DY = RIGHT - LEFT	ALPHAC 00191
IF(ITAG.EQ.2) GO TO 5155	ALPHAC 00192
DIST = BOXTE - L2	ALPHAC 00193
YC(I) = LEFT + DIST*(DY/DX)	ALPHAC 00194
GO TO 5160	ALPHAC 00195
5155 DIST = BOXTE - R2	ALPHAC 00196
YC(I) = RIGHT + DIST*(DY/DX)	ALPHAC 00197
5160 CONTINUE	ALPHAC 00198
NTRAPS = I-1	ALPHAC 00199
DO 5175 NX=1,NTRAPS	ALPHAC 00200
A= BOXTE - XC(NX)	ALPHAC 00201
B= BOXTE - XC(NX+1)	ALPHAC 00202
IF(A.GT.1.) A = 1.0	ALPHAC 00203
IF(B.GT.1.) B = 1.0	ALPHAC 00204
H= YC(NX+1) - YC(NX)	ALPHAC 00205
IF(ITAG.EQ.2) H = -H	ALPHAC 00206
TA = TA + 0.5*(A+B)*H	ALPHAC 00207
5175 CONTINUE	ALPHAC 00208
IF(ITAG.EQ.2) GO TO 5020	ALPHAC 00209
GO TO 5013	ALPHAC 00210
C	ALPHAC 00211
C COMPUTE FOR UPPER RIGHT HAND CORNER	ALPHAC 00212
5300 CONTINUE	ALPHAC 00213
I = 1	ALPHAC 00214
XC(I) = R1	ALPHAC 00215
YC(I) = RIGHT	ALPHAC 00216
5310 I = I+1	ALPHAC 00217
IF(NBK1.EQ.0) GO TO 5350	ALPHAC 00218
KIDX = K1 + NBK1-1 -I +2	ALPHAC 00219
IF(XLED(KIDX).LT.BOXLE) GO TO 5325	ALPHAC 00220
XC(I) = XLED(KIDX)	ALPHAC 00221
YC(I) = YLED(KIDX)	ALPHAC 00222
GO TO 5310	ALPHAC 00223
5325 CONTINUE	ALPHAC 00224
XC(I) = BOXLE	ALPHAC 00225
DX = XLED(KIDX+1)-XLED(KIDX)	ALPHAC 00226
DY = YLED(KIDX+1)-YLED(KIDX)	ALPHAC 00227
DIST = BOXLE - XLED(KIDX)	ALPHAC 00228
YC(I)= YLED(KIDX) + DIST*(DY/DX)	ALPHAC 00229


```

      GO TO 5360
5350 CONTINUE
      XC(I) = BOXLE
      DX = R1 - L1
      DY = RIGHT - LEFT
      DIST = BOXLE - L1
      YC(I) = LEFT + DIST*(DY/DX)
5360 CONTINUE
      NTRAPS = I-1
      DO 5375 NX = 1, NTRAPS
      A = XC(NX) - BOXLE
      B = XC(NX+1) - BOXLE
      IF(A.LT.1..AND.B.LT.1.) GO TO 5370
      IF(A.GT.1..AND.B.GT.1.) GO TO 5365
C
      A.GT.B1 AND B.LT.B1
C
      DX = XC(NX) - XC(NX+1)
      DY = YC(NX) - YC(NX+1)
      IF(DY.LT.EPS) GO TO 5375
      DIST = BOXLE - XC(NX+1)
      YINT = YC(NX+1) + DIST*(DY/DX)
      TA = (YC(NX)-YINT) + TA
      YC(NX) = YINT
      A = 1.0
      GO TO 5370
5365 CONTINUE
      TA = TA + (YC(NX)-YC(NX+1))
      GO TO 5375
5370 CONTINUE
      H = YC(NX) - YC(NX+1)
      TA = TA + 0.5*(A+B)*H
5375 CONTINUE
5400 AREA = AREA - TA
4000 CONTINUE
      RETURN
      END

```

```

ALPHAC 00230
ALPHAC 00231
ALPHAC 00232
ALPHAC 00233
ALPHAC 00234
ALPHAC 00235
ALPHAC 00236
ALPHAC 00237
ALPHAC 00238
ALPHAC 00239
ALPHAC 00240
ALPHAC 00241
ALPHAC 00242
ALPHAC 00243
ALPHAC 00244
ALPHAC 00245
ALPHAC 00246
ALPHAC 00247
ALPHAC 00248
ALPHAC 00249
ALPHAC 00250
ALPHAC 00251
ALPHAC 00252
ALPHAC 00253
ALPHAC 00254
ALPHAC 00255
ALPHAC 00256
ALPHAC 00257
ALPHAC 00258
ALPHAC 00259
ALPHAC 00260
ALPHAC 00261
ALPHAC 00262
ALPHAC 00263
ALPHAC 00264
ALPHAC 00265

```

	SUBROUTINE NTRCEP(J, YEDG,XEDG,	L1,C1,R1,NBK1,K1,INDEX)	NTRCEP 00002
C			NTRCEP 00003
C	J = INDEX OF CHORD NUMBER		NTRCEP 00004
C	NBK1 = NUMBER OF BREAK POINTS ON EDGE FOR THIS CHORD.		NTRCEP 00005
C	L1 = X COORDINATE OF LEADING EDGE LEFT SIDE INTERSECTION		NTRCEP 00006
C	C1 = X COORDINATE OF LEADING EDGE CENTER LINE INTERSECTION		NTRCEP 00007
C	R1 = X COORDINATE OF LEADING EDGE RIGHT SIDE INTERSECTION		NTRCEP 00008
C	K1 = INDEX OF XLEA AND YLEA ARRAYS THAT DEFINE A KINK IF		NTRCEP 00009
C	ONE EXISTS		NTRCEP 00010
	COMMON /LAREA / LEFT,RIGHT,ICODE	LAREA	00002
	REAL LEFT,L1,L2	NTRCEP	00012
C	LEFT = Y COORDINATE OF LEFT SIDE OF CHORD	NTRCEP	00013
C	RIGHT = Y COORDINATE OF RIGHT SIDE OF CHORD	NTRCEP	00014
	DIMENSION XEDG(1), YEDG(1)	NTRCEP	00015
	YJ = J	NTRCEP	00016
	EPS = 1.0E-04	NTRCEP	00017
	K=2	NTRCEP	00018
	1 IF(LEFT.LT.YEDG(K)-EPS) GO TO 2	NTRCEP	00019
	K= K+1	NTRCEP	00020
	GO TO 1	NTRCEP	00021
	2 DX = XEDG(K) - XEDG(K-1)	NTRCEP	00022
	DY = YEDG(K) - YEDG(K-1)	NTRCEP	00023
	DIST = LEFT - YEDG(K-1)	NTRCEP	00024
	L1 = XEDG(K-1) + (DX/DY) * DIST	NTRCEP	00025
C		NTRCEP	00026
C	FIND CENL AND BEGIN COUNTING BREAKS	NTRCEP	00027
	NBK1 = 0	NTRCEP	00028
	K1 = 0	NTRCEP	00029
	3 IF(YJ .LT.YEDG(K) +EPS) GO TO 4	NTRCEP	00030
C		NTRCEP	00031
C	KINK(S) BETWEEN LEFT AND CENTER LINE	NTRCEP	00032
	IF(K1.EQ.0) K1 = K	NTRCEP	00033
	NBK1 = NBK1 + 1	NTRCEP	00034
	K = K + 1	NTRCEP	00035
	GO TO 3	NTRCEP	00036
	4 IF(INDEX.EQ.1) GO TO 40	NTRCEP	00037
	IF(ABS(YEDG(K)-YJ) .GT.EPS) GO TO 40	NTRCEP	00038
	IF(K1.EQ.0) K1 = K	NTRCEP	00039
	NBK1 = NBK1 +1	NTRCEP	00040
	IF (YEDG(K+1)-YEDG(K).GT.EPS) GO TO 104	NTRCEP	00041
	NBK1 = NBK1 + 1	NTRCEP	00042
	K = K + 1	NTRCEP	00043
	104 CONTINUE	NTRCEP	00044
	C1 = XEDG(K)	NTRCEP	00045
	GO TO 5	NTRCEP	00046
	40 DX = XEDG(K) - XEDG(K-1)	NTRCEP	00047
	DY = YEDG(K) - YEDG(K-1)	NTRCEP	00048
	DIST = YJ - YEDG(K-1)	NTRCEP	00049
	C1 = XEDG(K-1) + (DX/DY) * DIST	NTRCEP	00050
C		NTRCEP	00051
C	FIND R1 IN SAME MANNER AS CENL	NTRCEP	00052
	5 IF(RIGHT.LT.YEDG(K)+EPS) GO TO 6	NTRCEP	00053
C		NTRCEP	00054
C	KINKS BETWEEN CENTER LINE AND RIGHT SIDE OF CHORD	NTRCEP	00055
	IF(K1.EQ.0) K1 = K	NTRCEP	00056
	NBK1 = NBK1 + 1	NTRCEP	00057
	K = K + 1	NTRCEP	00058

```

GO TO 5
6 DX = XEDG(K) - XEDG(K-1)
  DY = YEDG(K) - YEDG(K-1)
  DIST = RIGHT - YEDG(K-1)
  R1 = XEDG(K-1) + (DX/DY) * DIST
RETURN
END

```

```

NTRCEP 00059
NTRCEP 00060
NTRCEP 00061
NTRCEP 00062
NTRCEP 00063
NTRCEP 00064
NTRCEP 00065

```

	SUBROUTINE PWAIC(WING,IBOX,LBOX,IWAKE, JCQL)	PWAIC 00002
C		PWAIC 00003
C	COMPUTES THE POINTER ARRAY (MUAIC) FOR THE SPATIAL AIC ARRAY	PWAIC 00004
C	OF THE LEFT WING (TAIL) ON ONE CHORD OF THE RIGHT WING (TAIL)	PWAIC 00005
C		PWAIC 00006
C	WING = WING/TAIL INDICATOR	PWAIC 00007
C	IBOX = BOX CODE ARRAY TO USE	PWAIC 00008
C	IWAKE = ARRAY OF WAKE EDGE LOCATIONS FOR WING	PWAIC 00009
C	JCQL = THE (UNSUBDIVIDED) CHORD NUMBER OF INTEREST	PWAIC 00010
C	SURF = INDICATOR OF WHETHER ANY LEFT SURFACE IS INTER-	PWAIC 00011
C	CEPTED BY THE MACH CONE FOR THIS CORD	PWAIC 00012
C		PWAIC 00013
	COMMON /GEOMTY/ COPLAN,NSUBDV,XSUBDV,NSUBD2,NSUBCN,NSURF,	GEOMTY 00002
1	B1,B1BETA,B1S,B1BTAS,WLAX,WLAZ,PSIW,	GEOMTY 00003
2	MXBW,MXBBW,MYBW,MYBBW,MXBSW,MYBSW,MYBBSW,	GEOMTY 00004
3	IXBW,XCENTR	GEOMTY 00005
	LOGICAL COPLAN	GEOMTY 00006
	COMMON /GEOM2 / TLAX,TLAZ,PSIT,MXBT,MYBT,MYBBT,MXBST,MYBST,	GEOM2 00002
1	MYBBST,IXBT,IXBST,CAPL	GEOM2 00003
	COMMON /MUAICS/ YBAR,EL,MUAIC(2,50),NROWS,SURF,	MUAICS 00002
1	YBARL,ELL, MUAICL(2,50),NROWSL,SURFL,PSIDIF	MUAICS 00003
	LOGICAL SURF,SURFL	MUAICS 00004
	COMMON /EDGES / FEXLOC(250), TEXLOC(250),JDIAG	EDGES 00002
C		PWAIC 00018
	LOGICAL WING	PWAIC 00019
	DIMENSION IBOX(LBOX,8), ICODE(50), IWAKE(1)	PWAIC 00020
	DATA EPS / 1.0E-4 /	PWAIC 00021
C		PWAIC 00022
	IF (WING) GO TO 100	PWAIC 00023
C		PWAIC 00024
	THE CALL IS FOR A TAIL CHORD	PWAIC 00025
	PSI2 = PSIT + PSIT	PWAIC 00026
	IXB = (IXBT-IXBW)/NSUBDV + 1	PWAIC 00027
	MXB = MXBT	PWAIC 00028
	IF (JCQL .LE. MYBT) GO TO 120	PWAIC 00029
C	THE CHORD IS ON THE TIP DIAPHRAGM	PWAIC 00030
	IFIRST = IXB	PWAIC 00031
	NI = MXBT-IFIRST+1	PWAIC 00032
	GO TO 130	PWAIC 00033
C		PWAIC 00034
	THE CALL IS FOR A WING CHORD	PWAIC 00035
100	CONTINUE	PWAIC 00036
	PSI2 = PSIW + PSIW	PWAIC 00037
	IXB = 1	PWAIC 00038
	IF (COPLAN) GO TO 110	PWAIC 00039
	MXB = MXBBW	PWAIC 00040
	IF (JCQL .GT. MYBW) GO TO 115	PWAIC 00041
	ISUB = JCQL*NSUBDV - NSUBD2	PWAIC 00042
	GO TO 125	PWAIC 00043
C		PWAIC 00044
	THE CALL IS FOR A COPLANAR WING-TAIL	PWAIC 00045
110	CONTINUE	PWAIC 00046
	MXB = MXBT	PWAIC 00047
	IF (JCQL .LE. MYBT) GO TO 120	PWAIC 00048
C	THE CHORD IS ON THE TIP DIAPHRAGM	PWAIC 00049
115	CONTINUE	PWAIC 00050
	IFIRST = 1	PWAIC 00051

NJ = MXB	PMAIC 00052
GO TO 130	PMAIC 00053
C	PMAIC 00054
C THE CHORD IS ON PLANFORM	PMAIC 00055
120 CONTINUE	PMAIC 00056
ISUB = MYBSW + JCQL*NSUBDV - NSUBD2	PMAIC 00057
125 CONTINUE	PMAIC 00058
IFRST = (TEXLOC(ISUB)-IXBW) / NSUBDV + 1	PMAIC 00059
NJ = MXB - IFRST + 1	PMAIC 00060
C	PMAIC 00061
130 CONTINUE	PMAIC 00062
CALL DCDER(IBOX,LBOX, IFRST,JCQL, MXB,JCQL, .F., ICODE)	PMAIC 00063
IROW = IFRST	PMAIC 00064
DO 135 I = 1,NJ	PMAIC 00065
IA = I	PMAIC 00066
IF (ICODE(I) .NE. 0) GO TO 140	PMAIC 00067
IROW = IROW + 1	PMAIC 00068
135 CONTINUE	PMAIC 00069
140 CONTINUE	PMAIC 00070
IF (IA .GE. NJ) GO TO 155	PMAIC 00071
DO 145 I = IA,NJ	PMAIC 00072
IF (ICODE(I) .EQ. 0) GO TO 150	PMAIC 00073
IROW = IROW + 1	PMAIC 00074
145 CONTINUE	PMAIC 00075
150 CONTINUE	PMAIC 00076
IROW = IROW - 1	PMAIC 00077
155 CONTINUE	PMAIC 00078
NROWS = IROW - IXB + 1	PMAIC 00079
C	PMAIC 00080
C COMPUTE HORIZONTAL AND VERTICAL OFFSETS	PMAIC 00081
YMSND = (JCQL-.5)*COS(PSI2)	PMAIC 00082
C = Y-OFFSET ON THE SENDING SURFACE OF THE PROJECTION OF THE	PMAIC 00083
C RECEIVING CHORD	PMAIC 00084
JBAR = IFIX(YMSND) + 1	PMAIC 00085
C = CHORD CONTAINING YMSND	PMAIC 00086
YBAR = YMSND - JBAR + .5	PMAIC 00087
C = DISTANCE FROM NEAREST SENDING CHORD CENTER TO PROJEC-	PMAIC 00088
C TION OF THE RECEIVING CHORD, POSITIVE RIGHT.	PMAIC 00089
EL = (JCQL - .5) * SIN(PSI2)	PMAIC 00090
C = VERTICAL SEPARATION BETWEEN THE SENDING PLANE AND THE	PMAIC 00091
C RECEIVING CHORD	PMAIC 00092
IF (YBAR) 160,165,170	PMAIC 00093
160 JMIN = JBAR - 1	PMAIC 00094
NBOXES = 2	PMAIC 00095
GO TO 180	PMAIC 00096
165 JMIN = JBAR	PMAIC 00097
NBOXES = 1	PMAIC 00098
GO TO 180	PMAIC 00099
170 JMIN = JBAR	PMAIC 00100
NBOXES = 2	PMAIC 00101
C	PMAIC 00102
180 CONTINUE	PMAIC 00103
JL = 1	PMAIC 00104
SURF = .F.	PMAIC 00105
C	PMAIC 00106
C START OF LOOP ON ROWS, FORWARD FROM RECEIVING BOX CENTER, TO	PMAIC 00107
C DEFINE THE MUAIC ARRAY	PMAIC 00108

DO 260 I = 1, NROWS	PWAIC 00109
XI = I - .5 - ABS(EL)	PWAIC 00110
IF (XI .LE. EPS) GO TO 260	PWAIC 00111
IF (JMIN .GT. 0) GO TO 260	PWAIC 00112
C CENTER LINE HAS BEEN CROSSED, THEREFORE THERE MAY BE CONTRIBU-	PWAIC 00113
C TION FROM THE LEFT WING FOR THIS ROW	PWAIC 00114
JM = -JMIN + 1	PWAIC 00115
JMM = JM - JL + 1	PWAIC 00116
CALL DCODER(IBOX, LBOX, IROW, 1, IROW, JMM, .F., ICODE)	PWAIC 00117
DO 240 J = JL, JM	PWAIC 00118
IF (ICODE(JMM) .NE. 0) GO TO 250	PWAIC 00119
JMM = JMM - 1	PWAIC 00120
240 CONTINUE	PWAIC 00121
NROWS = I - 1	PWAIC 00122
GO TO 290	PWAIC 00123
C CONTRIBUTING BOXES HAVE BEEN FOUND FOR THIS ROW	PWAIC 00124
250 CONTINUE	PWAIC 00125
SURF = .T.	PWAIC 00126
JL = J	PWAIC 00127
IF (YBAR .GE. 0) GO TO 255	PWAIC 00128
MUAIC(1, I) = NBOXES - JM + 1	PWAIC 00129
MUAIC(2, I) = NBOXES - JL + 1	PWAIC 00130
GO TO 270	PWAIC 00131
255 CONTINUE	PWAIC 00132
MUAIC(1, I) = JL	PWAIC 00133
MUAIC(2, I) = JM	PWAIC 00134
GO TO 270	PWAIC 00135
C	PWAIC 00136
C CENTER LINE HAS NOT BEEN CROSSED	PWAIC 00137
260 MUAIC(1, I) = 0	PWAIC 00138
MUAIC(2, I) = 0	PWAIC 00139
C	PWAIC 00140
270 CONTINUE	PWAIC 00141
NBOXES = NBOXES + 2	PWAIC 00142
JMIN = JMIN - 1	PWAIC 00143
IROW = IROW - 1	PWAIC 00144
280 CONTINUE	PWAIC 00145
C END OF LOOP FORWARD ON ROWS, FROM 180*	PWAIC 00146
C	PWAIC 00147
290 CONTINUE	PWAIC 00148
RETURN	PWAIC 00149
END	PWAIC 00150

OVERLAY (AFHBOX,1,3)	MODES	00002
PROGRAM MODES	MODES	00003
COMMON /PROBLM/ XMACH,NMODES,NTSLOP,NKVALS,SMOOTH,NDEG,CRDFIT,	PROBLM	00002
1 EXAIC,SUBDV,PLYWOOD	PROBLM	00003
LOGICAL SMOOTH,CRDFIT,EXAIC,SUBDV,PLYWOOD	PROBLM	00004
COMMON /CONTRL/ PREVEX,OMACH, TITLE(8), PRVGEQ4,PRVMODE,DIHW,DIHT,	CONTRL	00002
1 DEFAULT	CONTRL	00003
LOGICAL PRVGEOM,PRVMODE,DIHW,DIHT,DEFAULT	CONTRL	00004
COMMON /GEOMTY/ COPLAN,NSUBDV,XSUBDV,NSUBD2,NSUBCN,NSURF,	GEOMTY	00002
1 B1,B1BETA,B1S,B1BTAS,WLAX,WLAZ,PSIW,	GEOMTY	00003
2 MXBW,MXBBW,MYBW,MYBBW,MXBSW,MYBSW,MYBBSW,	GEOMTY	00004
3 IXBW,XCENTR	GEOMTY	00005
LOGICAL COPLAN	GEOMTY	00006
COMMON /GEOM2 / TLAX,TLAZ,PSIT,MXBT,MYBT,MYBBT,MXBST,MYBST,	GEOM2	00002
1 MYBBST,IXBT,IXBST,CAPL	GEOM2	00003
COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUTP,	FILES	00002
1 IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC	FILES	00003
EQUIVALENCE (IWTFSC,ITSLSC)	MODES	00009
COMMON /IOCONT/ OPLAIC,OSPAIC,WTGEOM,WTGNAF,WTSL,WTBL,FRBOX,	IOCONT	00002
1 PRPAIC,PRSAIC,PRMODS,PRCOEF,PRDW,FRSW,FRVP,	IOCONT	00003
2 FRBL,PRDCP,PRGNAC,PRGNAC,FRSL,FRWL,FRNW,PRCM	BCSFRB	00001
EQUIVALENCE (FRWL,FRDW)	IOCONT	00005
LOGICAL OPLAIC,OSPAIC,WTGEOM,WTGNAF,WTSL,WTBL,FRBOX,PRPAIC,	IOCONT	00006
1 PRSAIC,PRMODS,PRCOEF,PRDW,FRSW,FRVP,FRBL,FRSL,PRGNAC,	IOCONT	00007
2 PRDCP,PRGNAC,FRWL,FRWL,FRNW,PRCM	BCSFRB	00002
COMMON /TAPEIO/ NFS,NMS,LS,NMR,ID(20),NID,ITYPE,LRS,LWS,M,N,	TAPEIO	00002
1 PARM(10),IRR	TAPEIO	00003
DIMENSION I PARM(10)	TAPEIO	00004
EQUIVALENCE (FARM,I PARM)	TAPEIO	00005
COMMON /ARRAYS/ KBXCDW,LBXCDW,LBOXC,KBXCDT,LBXCDT,KJALPH,LJALPH,	ARRAYS	00002
1 KALPHA,KKERNL,LKERNL,KPNTRM,LPNTRM,DEFSL,KELPHI,	ARRAYS	00003
2 LMODES,KPNTSD,L PNTSD,KSDW,LSDW,KPNTDW,L PNTDW,	ARRAYS	00004
3 KDW,LDW,KTVP,LTPV	ARRAYS	00005
COMMON / MODES/ SYH,SYMT,MTYPEW,MTYPEP	MODECOM	00002
COMMON /CHECKPR/ DPPPCR,GEOPCR,MODPCR,AICPCR,NMSCPR,SMPCR,GAFPCR	CHECKPR	00002
LOGICAL DPPPCR,GEOPCR,MODPCR,AICPCR,NMSCPR,SMPCR,GAFPCR	CHECKPR	00003
EQUIVALENCE (CHECKPR,MODPCR)	MODES	00015
LOGICAL CHECKPR	MODES	00016
C DEFSL(2,NBOXES), XX(NPTS), YY,ZZ SAME, A(NO OF COEF)	MODES	00017
DIMENSION DEFSL(2,1000), XX(100),YY(100),ZZ(100), A(21)	MODES	00018
COMMON /INDEX/ JS(100),NOC(100),JS(100),JOC(100)	MODES	00019
DIMENSION I PNTM(2,100)	MODES	00020
DIMENSION XP(6),YP(6),X1(100),Y1(100)	MODES	00021
DIMENSION DOB(50)	MODES	00022
DIMENSION FEXLOC(250), TEXLOC(250)	MODES	00023
C FEXLOC(MYBSW*MYBST), TEXLOC SAME)	MODES	00024
C	MODES	00026
LOGICAL MREAD,RANDIN,MWRIT,RANDOU	MODES	00027
NAMLIST /CARDM / NMODES,NTSLOP	FTNXI	00043
MREAD = .FALSE.	MODES	00028
MWRIT = .FALSE.	MODES	00029
RANDIN = .FALSE.	MODES	00030
RANDOU = .FALSE.	MODES	00031
EPS = 1.0E-04	MODES	00032
GAMMA = 1.4	MODES	00033
GAMC = XMACH*(GAMMA+1.0)/2.	MODES	00034
IF(.NOT.PRVMODE) GO TO 100	MODES	00035

IF(.NOT.PRVGCOM) GO TO 50	MODES	00036
WRITE (NT6,7005)	MODES	00037
RETURN	MODES	00038
C	MODES	00039
50 CONTINUE	MODES	00040
IF(MTYPEW.EQ.3) GO TO 75	MODES	00041
IF(NSURF.EQ.2.AND.MTYPET.EQ.3) GO TO 75	MODES	00042
WRITE (NT6,7010)	MODES	00043
GO TO 125	MODES	00044
C	MODES	00045
75 CONTINUE	MODES	00046
WRITE (NT6,7015)	MODES	00047
CALL FLUSH(1)	MODES	00048
C	MODES	00049
100 CONTINUE	MODES	00050
C	MODES	00051
C	MODES	00052
NMODES = 0	MODES	00053
NMSLOP = 0	MODES	00054
READ(NT5,CARDM)	MODES	00055
125 CONTINUE	MODES	00056
REXIND IGEOBC	MODES	00057
C	MODES	00058
C READ FEXLOC AND TEXLOC ARRAY FROM GEOMETRY SCRATCH FILE	MODES	00059
MNAME =GHFEXLOC	MODES	00060
CALL RDINIT	MODES	00061
ITYPE = SHMIXED	MODES	00062
NMS = 1	MODES	00063
IF(.NOT.COPLAN.AND.NSURF.EQ.2) NMS = 2	MODES	00064
CALL READMX(IGEOBC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,	MODES	00065
1 LRS,FEXLOC, M,N,PARM,IRR)	MODES	00066
IF(IRR.NE.0) GO TO 6010	MODES	00067
C	MODES	00068
MNAME =GHTEXLOC	MODES	00069
CALL RDINIT	MODES	00070
ITYPE = SHMIXED	MODES	00071
CALL READMX(IGEOBC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,	MODES	00072
1 LRS,TEXLOC,M,N,PARM,IRR)	MODES	00073
IF(IRR.NE.0) GO TO 6010	MODES	00074
C	MODES	00075
C	MODES	00076
C ZERO OUT THE ROW AND COL POINTERS	MODES	00077
DO 150 I = 1,400	MODES	00078
IS(I) = 0	MODES	00079
150 CONTINUE	MODES	00080
C	MODES	00081
C	MODES	00082
C DETERMINE STARTING BOXES AND NUMBER OF BOXES PER CHORD.	MODES	00083
IYB1 = (NSUBDV+1)/2	MODES	00084
DO 300 NB=1,NSURF	MODES	00085
IF(NB.EQ.2) GO TO 200	MODES	00086
NC = 1	MODES	00087
NCH = NYBW	MODES	00088
NCF = 0	MODES	00089
ICW = IXBW - IYB1	MODES	00090
IXB = IXBW	MODES	00091
GO TO 225	MODES	00092

200 CONTINUE	MODES 00093
NC = MYBW + 1	MODES 00094
NCH = MYBT + MYBT	MODES 00095
NCF = MYBW * NSUBDV	MODES 00096
IFBT = (IXBT-IXBW)/NSUBDV + 1	MODES 00097
225 CONTINUE	MODES 00098
IYB = IYB1 + NCF	MODES 00099
DO 250 J=NC,NCH	MODES 00100
IS(J) = FEXLOC(IYB) + 1.0	MODES 00101
ITEI = TEXLOC(IYB)	MODES 00102
IF(NSUBDV.EQ.1) GO TO 240	MODES 00103
IS(J) = (IS(J)-ICN)/NSUBDV + 1	MODES 00104
ITEI = (ITEI-IXB)/NSUBDV + 1	MODES 00105
240 CONTINUE	MODES 00106
NOC(J) = ITEI-IS(J) + 1	MODES 00107
IYB = IYB + NSUBDV	MODES 00108
250 CONTINUE	MODES 00109
300 CONTINUE	MODES 00110
CALL ROPER	MODES 00111
C	MODES 00112
C	MODES 00113
C FIND OVERLAP OF 2 PLANFORMS IF THEY ARE NON-COPLANAR	MODES 00114
IOVLAP = 0	MODES 00115
NPNTRS = MXBW+1	MODES 00116
IF(NSURF.EQ.1) GO TO 325	MODES 00117
IF(COPLAN) GO TO 324	MODES 00118
IF(IFBT.GT.MXBW) GO TO 324	MODES 00119
IOVLAP = MXBW - IFBT + 1	MODES 00120
NPNTRS = MXBT + IOVLAP + 1	MODES 00121
GO TO 325	MODES 00122
324 CONTINUE	MODES 00123
NPNTRS = MXBT + 1	MODES 00124
325 CONTINUE	MODES 00125
C COMPUTE PCINTER ARRAY AND STORE ON MODESC	MODES 00126
REWIND MODESC	MODES 00127
C	MODES 00128
IPNTRM(1,1) = 1	MODES 00129
IPNTRM(2,1) = JS(1)	MODES 00130
DO 320 I=2,NPNTRS	MODES 00131
IPNTRM(1,I) = IPNTRM(1,I-1) + JOC(I-1)	MODES 00132
IPNTRM(2,I) = JS(I)	MODES 00133
320 CONTINUE	MODES 00134
CALL RDINIT	MODES 00135
IPARM(3) = IOVLAP	MODES 00136
ITYPE = 5H MIXED	MODES 00137
CALL WRTEMX(MODESC,MXWRT,RANDOU,NFS,NMS,LS,NMR,LWS,2,1D,	MODES 00138
1 IPNTRM,ITYPE,2,NPNTRS,PARM,IRR)	MODES 00139
IF(IRR.NE.0) GO TO 6030	MODES 00140
C FIRST LOOP DETERMINES MODE SHAPES.	MODES 00141
C SECOND LOOP DETERMINES THICKNESS SLOPES.	MODES 00142
C	MODES 00143
DO 3 0 IPASS=1,2	MODES 00144
IF(1 .SS.EQ.2) GO TO 2100	MODES 00145
C LOOP ON NUMBER OF SURFACES	MODES 00146
DO 2000 MS=1,NSURF	MODES 00147
IF(NS.EQ.NSURF) GO TO 330	MODES 00148
C	MODES 00149

IFILE = IVPSC	MODES 00150
REWIND IVPSC	MODES 00151
GO TO 340	MODES 00152
330 CONTINUE	MODES 00153
IFILE = MODESC	MODES 00154
340 CONTINUE	MODES 00155
C	MODES 00156
C LOOP ON NUMBER OF MODES	MODES 00157
DO 1500 NM = 1, NMODES	MODES 00158
C	MODES 00159
C	MODES 00160
C ZERO OUT THE DEFSL ARRAY	MODES 00161
DO 350 I = 1, LMODES	BCSMOA 00002
DEFSL(1,I) = 0.0	MODES 00163
DEFSL(2,I) = 0.0	MODES 00164
350 CONTINUE	MODES 00165
C	BCSMOA 00003
C ZERO OUT THE COEFFICIENT ARRAY	BCSMOA 00004
DO 355 I = 1, 21	BCSMOA 00005
A(I) = 0.0	BCSMOA 00006
355 CONTINUE	BCSMOA 00007
C	MODES 00166
C INPUT FIRST PLANFORM IF THERE IS A TAIL SECTION	MODES 00167
IF(NS.EQ.1) GO TO 400	MODES 00168
READ (IVPSC) DEFSL	MODES 00169
C	MODES 00170
C	MODES 00171
400 CONTINUE	MODES 00172
IF(.NOT.FRMVODE) GO TO 450	MODES 00173
CALL RDINIT	MODES 00174
IF(NS.EQ.1.AND.NM.EQ.1) NFS = 2	MODES 00175
MNAME = GH COEF.	MODES 00176
CALL READMX(IGEOBC,MOREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,	MODES 00177
1 LRS,A,M,N,PARM,IRR)	MODES 00178
IF(IRR.NE.0) GO TO 6010	MODES 00179
NFS = 0	MODES 00180
C	MODES 00181
GO TO 551	MODES 00182
450 CONTINUE	MODES 00183
ITYPE = MTYPEW	MODES 00184
IF(NS.EQ.2) ITYPE = MTYPET	MODES 00185
GO TO (501,502,503), ITYPE	MODES 00186
C	MODES 00187
C READ IN POLYNOMIAL COEFFICIENTS	MODES 00188
501 CONTINUE	MODES 00189
READ(NT5,8010) IDEG	MODES 00190
IF (IDEG .LT. 0 .OR. IDEG .GT. 5) GO TO 6000	MODES 00191
8010 FORMAT(2I5)	MODES 00192
MDEG = IDEG + 1	MODES 00193
DEG = MDEG	MODES 00194
DEG2 = DEG/2.	MODES 00195
NC = DEG+DEG2 + DEG2 + EPS	MODES 00196
READ(NT5,8015) (A(I), I=1, NC)	MODES 00197
8015 FORMAT(7E10.0)	MODES 00198
IFLAG = 1	MODES 00199
GO TO 530	MODES 00200
C	MODES 00201

C	READ IN DEFLECTIONS AT SELECTED LOCATIONS AND FIT A POLYNOMIAL	MODES	00202
C	OF DEGREE IDEG TO THE POINTS USING METHOD OF LEAST SQUARES.	MODES	00203
	502 CONTINUE	MODES	00204
	READ (NTS,8010) IDEG,NPTS	MODES	00205
	IF (IDEG .LT. 0 .OR. IDEG .GT. 5) GO TO 6005	MODES	00206
	IF (NPTS .GT. 100 .OR. NPTS .LT. 1) GO TO 6005	MODES	00207
	READ (NTS,8020) (XX(I),YY(I),ZZ(I),I=1,NPTS)	MODES	00208
	8020 FORMAT(6E10.0)	MODES	00209
	IDIM = 1	MODES	00210
	CN = 1.0	MODES	00211
C		MODES	00212
C	CN IS A SCALE FACTOR TO REDUCE THE MAGNITUDE OF THE NUMBERS	MODES	00213
C	IDIM IS A DIMENSION VARIABLE SET TO 1 TO INDICATE FIT IS	MODES	00214
C	BEING MADE ON REAL VALUES . IDIM = 2 FOR COMPLEX Z VALUES.	MODES	00215
	CALL FITTER(IDEG,NPTS,XX,YY,ZZ,A,CN,IDIM)	MODES	00216
C		MODES	00217
	IFLAG = 2	MODES	00218
	MDEG = IDEG + 1	MODES	00219
	DEG = IDEG + 1	MODES	00220
	DEG2 = DEG/2.	MODES	00221
	NC = DEG*DEG2 + DEG2 + EPS	MODES	00222
C		MODES	00223
C		MODES	00224
	550 CONTINUE	MODES	00225
C		MODES	00226
C	STORE THE COEFFICIENTS ON THE THIRD FILE OF THE IGEOSC FILE.	MODES	00227
C	IF THE COEFFICIENTS ARE TO BE PRINTED THE ONES FOR THE FIRST	MODES	00228
C	SURFACE MUST BE STORED ON A SCRATCH FILE TEMPORARILY.	MODES	00229
C		MODES	00230
	CALL RDINIT	MODES	00231
	IF(NS.EQ.1.AND.NM.EQ.1) NFS = 2	MODES	00232
	IPARM(3) = IDEG	MODES	00233
	IPARM(4) = IFLAG	MODES	00234
	ITYPE = SHIMIXED	MODES	00235
	CALL WRTEMX(IGEOSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,1,ID,	MODES	00236
	1 A,ITYPE,1,NC,FARM,IRR)	MODES	00237
	IF(IRR.NE.0) GO TO 6050	MODES	00238
	NFS = 0	MODES	00239
C		MODES	00240
	IF(.NOT.PRCOEF) GO TO 3550	MODES	00241
	IF(NS.EQ.2.OR.NSURF.EQ.1) GO TO 3550	MODES	00242
	IF(NM.EQ.1) REWIND IAICSC	MODES	00243
C		MODES	00244
	CALL WRTEMX(IAICSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,1,ID,	MODES	00245
	1 A,ITYPE,1,NC,FARM,IRR)	MODES	00246
	IF(IRR.NE.0) GO TO 6060	MODES	00247
C		MODES	00248
	3550 CONTINUE	MODES	00249
	IF(NS.EQ.2.AND.NM.EQ.1) REWIND IAICSC	MODES	00250
C		MODES	00251
C	EVALUATE THE POLYNOMIAL EQUATION FOR DEFLECTIONS.	MODES	00252
C	IE PARTIAL DERIVATIVE WITH RESPECT TO X TO GET SLOPES.	MODES	00253
C		MODES	00254
C		MODES	00255
	551 CONTINUE	MODES	00256
	IF(NM.NE.1) GO TO 560	MODES	00257
	IF(NS.EQ.2) GO TO 556	MODES	00258

C		MODES	00259
C	CALCULATE X,Y COORDINATES FOR EVALUATION OF POLYNOMIAL	MODES	00260
	X1(1) = XCENR	MODES	00261
	Y1(1) = 0.5*B1BETA	MODES	00262
	IF(NSURF.EQ.2) GO TO 552	MODES	00263
	MMAX = MAXD(MXBW,MYBW)	MODES	00264
	GO TO 554	MODES	00265
552	CONTINUE	MODES	00266
	MMAX = MAXD(MXBT,MYBT,MYBW)	MODES	00267
554	CONTINUE	MODES	00268
	DO 555 I= 2,MMAX	MODES	00269
	X1(I) = X1(I-1)+B1	MODES	00270
	Y1(I) = Y1(I-1)+B1BETA	MODES	00271
555	CONTINUE	MODES	00272
	GO TO 580	MODES	00273
C		MODES	00274
556	CONTINUE	MODES	00275
	XADJ = TLAX - WLAX	MODES	00276
	DO 557 I=1,MMAX	MODES	00277
	X1(I) = X1(I) - XADJ	MODES	00278
557	CONTINUE	MODES	00279
	GO TO 580	MODES	00280
C		MODES	00281
C		MODES	00282
580	CONTINUE	MODES	00283
	IF(NS.EQ.2) GO TO 580	MODES	00284
	IC = 0	MODES	00285
	ILIM = MXBW	MODES	00286
	IBEG = 1	MODES	00287
	NCH = 0	MODES	00288
	GO TO 564	MODES	00289
580	CONTINUE	MODES	00290
	IBEG = IFBT	MODES	00291
	ILIM = MXBT	MODES	00292
	NCH = MYBW	MODES	00293
	IC = 0	MODES	00294
	IUP = MXBW	MODES	00295
	IF(COPLAN) IUP = IFBT-1	MODES	00296
	DO 563 I=1,IUP	MODES	00297
	IC = IC + JOC(I)	MODES	00298
563	CONTINUE	MODES	00299
564	CONTINUE	MODES	00300
	DO 575 IX=IBEG,ILIM	MODES	00301
	I = IX	MODES	00302
	IF(NS.EQ.2) I = IX + ICALAP	MODES	00303
	XP(1) = 1.	MODES	00304
	DO 561 IP=2,MDEG	MODES	00305
561	XP(IP) = XP(IP-1) * X1(IX)	MODES	00306
	J1= JS(I)	MODES	00307
	JT= JOC(I)+ J1 -1	MODES	00308
	DO 570 J=J1,JT	MODES	00309
	IC = IC +1	MODES	00310
	IB = IS(J+NCH)	MODES	00311
	IT = IB + NOC(J+NCH) -1	MODES	00312
	IF(IX.LT.IB) GO TO 570	MODES	00313
	IF(IX.GT.IT) GO TO 570	MODES	00314
	YP(1) = 1.	MODES	00315

DO 562 JP =2,MDEG	MODES	00316
562 YP(JP) = YP(JP-1)*Y1(J)	MODES	00317
D = A(1)	MODES	00318
S = 0.0	MODES	00319
IF (MDEG .LT. 2) GO TO 567	BCSMDBA	00008
IA = 1	MODES	00320
DO 565 L2=2,MDEG	MODES	00321
DO 565 L3=1,L2	MODES	00322
L4 = L2-L3+1	MODES	00323
IA = IA + 1	MODES	00324
D = D + XP(L4)*YP(L3)*A(IA)	MODES	00325
IF(L4.EQ.1) GO TO 565	MODES	00326
L5 = L4 - 1	MODES	00327
S = S + L5*XP(L5)*YP(L3)*A(IA)	MODES	00328
565 CONTINUE	MODES	00329
567 CONTINUE	BCSMDBA	00009
DEFSL(1,IC) = D	MODES	00330
DEFSL(2,IC) = S	MODES	00331
570 CONTINUE	MODES	00332
575 CONTINUE	MODES	00333
GO TO 900	MODES	00334
C	MODES	00335
C READ IN DEFLECTIONS AND SLOPES AT BOX CENTERS	MODES	00336
503 CONTINUE	MODES	00337
IF(INTAPE.EQ.0.OR.INTAPE.EQ.5) GO TO 700	MODES	00338
C	MODES	00339
C MODES ON TAPE. CALL SPECIAL ROUTINE TO HANDLE.	MODES	00340
CALL TAPMOD(NS,NM,DEFSL)	MODES	00341
GO TO 900	MODES	00342
700 CONTINUE	MODES	00343
IF(NS.EQ.2) GO TO 720	MODES	00344
C	MODES	00345
C FIRST PLANFORM	MODES	00346
NCH = MYBW	MODES	00347
NC = 1	MODES	00348
GO TO 725	MODES	00349
720 CONTINUE	MODES	00350
NC = MYBW + 1	MODES	00351
NCH = MYBW + MYBT	MODES	00352
C	MODES	00353
C READ AND STORE DEFLECTIONS	MODES	00354
725 CONTINUE	MODES	00355
DO 750 J=NC,NCH	MODES	00356
IST = IS(J)	MODES	00357
NK = NOC(J) + IST - 1	MODES	00358
JSUM = 0	MODES	00359
ITROW = IST	MODES	00360
IF(.NOT.COPLAN.AND.NS.EQ.2) ITROW = IST + IOVLAP	MODES	00361
DO 730 I=1,ITROW	MODES	00362
730 JSUM = JSUM + JOC(I)	MODES	00363
JSU* = JSUM - JOC(ITROW) + 1	MODES	00364
REA (NT5,9015) (DO6(I),I=IS,NK)	MODES	00365
DO 750 I=IST,NK	MODES	00366
IX = I	MODES	00367
IF(.NOT.COPLAN.AND.NS.EQ.2) IX = I + IOVLAP	MODES	00368
ISUB = JSUM + J - JS(IX) - NC + 1	MODES	00369
DEFSL(1,ISUB) = DO6(I)	MODES	00370

JSUM = JSUM + JOC(IX)	MODES 00371
750 CONTINUE	MODES 00372
C	MODES 00373
C READ AND STORE SLOPES	MODES 00374
DO 775 J=NC,NCH	MODES 00375
IST = IS(J)	MODES 00376
NK = NOC(J) + IST - 1	MODES 00377
JSUM = 0	MODES 00378
ITROW = IST	MODES 00379
IF(.NOT.COPLAN.AND.NS.EQ.2) ITROW = IST + IOVLAP	MODES 00380
DO 770 I=1,ITROW	MODES 00381
770 JSUM = JSUM + JOC(I)	MODES 00382
JSUM = JSUM - JOC(ITROW) + 1	MODES 00383
READ(NT5,9015) (DOS(I),I=IST,NK)	MODES 00384
DO 775 I=IST,NK	MODES 00385
IX = I	MODES 00386
IF(.NOT.COPLAN.AND.NS.EQ.2) IX = I + IOVLAP	MODES 00387
ISUB = JSUM + J - JS(IX) - NC + 1	MODES 00388
DEFSL(2,ISUB) = DOS(I)	MODES 00389
JSUM = JSUM + JOC(IX)	MODES 00390
775 CONTINUE	MODES 00391
900 CONTINUE	MODES 00392
C	MODES 00393
C WRITE THE DEFSL ARRAY ONTO MODESC FILE	MODES 00394
IF(NS.EQ.NSURF) GO TO 925	MODES 00395
WRITE (IFILE) DEFSL	MODES 00396
GO TO 950	MODES 00397
925 CONTINUE	MODES 00398
CALL RDINIT	MODES 00399
ITYPE = SHMIXED	MODES 00400
N = IPNTRM(1,NPNTRS)-1	MODES 00401
CALL WRTEMX(IFILE,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,	MODES 00402
1 DEFSL,ITYPE,2,N,PARM,IRR)	MODES 00403
IF(IRR.NE.0) GO TO 6020	MODES 00404
950 CONTINUE	MODES 00405
IF(NS.NE.NSURF) GO TO 1500	MODES 00406
IF(.NOT.PRMOCS.AND..NOT. PRECOEF) GO TO 1500	MODES 00407
C	MODES 00408
C PRINT MODES, COEFFICIENTS OR BOTH	MODES 00409
C	MODES 00410
WRITE (NT6,9500) TITLE,NH,XMACH	MODES 00411
IF(.NOT.PRECOEF) GO TO 975	MODES 00412
IF(MTYPEW.EQ.3) GO TO 960	MODES 00413
C	MODES 00414
C PRINT COEFFICIENTS	MODES 00415
C	MODES 00416
IF(NSURF.EQ.1) GO TO 960	MODES 00417
CALL RDINIT	MODES 00418
ITYPE = SHMIXED	MODES 00419
CALL READMX(IAICSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,	MODES 00420
1 LRS,XX,M,N,PARM,IRR)	MODES 00421
IF(IRR.NE.0) GO TO 6070	MODES 00422
C	MODES 00423
IPLG = IPARM(4)	MODES 00424
IDG1 = IPARM(3)	MODES 00425
CALL PRECOF(IDG1,XX,IPLG)	MODES 00426
C	MODES 00427

960 CONTINUE	MODES 00428
IF(NS.EQ.1.AND.MTYPEW.EQ.3) GO TO 975	MODES 00429
IF(NS.EQ.2.AND.MTYPET.EQ.3) GO TO 975	MODES 00430
CALL PRECOF(IDEG,A,IFLAG)	MODES 00431
C	MODES 00432
975 CONTINUE	MODES 00433
NRCHS = MXBW	MODES 00434
IF(NSURF.EQ.2) NRCHS = MXBT + IOVLAP	MODES 00435
C	MODES 00436
CALL ROUTINE TO PRINT THE MODE SHAPES	MODES 00437
CALL MODOUT(DEFSL,JS,JOC,NRCHS,NM,IOVLAP)	MODES 00438
C	MODES 00439
1500 CONTINUE	MODES 00440
END FILE IFILE	MODES 00441
M = 1	MODES 00442
N = 400	MODES 00443
CALL WRTEMX(IFILE,MXWRT,RANDOU,NFS,NMS,LS,NMR,LWS,1,ID,	MODES 00444
1 IS,ITYPE,M,N,FARM,IRR)	MODES 00445
IF(IRR.NE.0) GO TO 6040	MODES 00446
C	MODES 00447
END FILE IFILE	MODES 00448
REWIND IFILE	MODES 00449
2000 CONTINUE	MODES 00450
GO TO 3000	MODES 00451
2100 CONTINUE	MODES 00452
C	MODES 00453
C DETERMINE THICKNESS SLOPES	MODES 00454
C	MODES 00455
NBV = IPNTRM(1,NPNTS) - 1	MODES 00456
REWIND ITSLSC	MODES 00457
IF(NTSLOP.NE.0) GO TO 2225	MODES 00458
C	MODES 00459
C WRITE ARRAY OF ONES	MODES 00460
C	MODES 00461
DO 2200 I=1,NBV	MODES 00462
DEFSL(I,I) = 1.0	MODES 00463
2200 CONTINUE	MODES 00464
C	MODES 00465
CALL RDINIT	MODES 00466
ITYPE = SHMIXED	MODES 00467
M = 1	MODES 00468
N = NBV	MODES 00469
CALL WRTEMX(ITSLSC,MXWRT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,	MODES 00470
1 DEFSL,ITYPE,M,N,FARM,IRR)	MODES 00471
IF(IRR.NE.0) GO TO 6080	MODES 00472
C	MODES 00473
END FILE ITSLSC	MODES 00474
REWIND ITSLSC	MODES 00475
GO TO 3000	MODES 00476
C	MODES 00477
2225 CONTINUE	MODES 00478
DO 2600 NS=1,NSURF	MODES 00479
IF(NS.EQ.NSURF) GO TO 2230	MODES 00480
IFILE = IVPSC	MODES 00481
REWIND IVPSC	MODES 00482
GO TO 2240	MODES 00483
2230 CONTINUE	MODES 00484

IFILE = ITSLS	MODES	00485
2240 CONTINUE	MODES	00486
C	MODES	00487
DO 2700 NSL=1,NTSLOP	MODES	00488
C	MODES	00489
C ZERO OUT THE ARRAY	MODES	00490
DO 2250 I=1,500	MODES	00491
DEFSL(1,I) = 0.0	MODES	00492
2250 CONTINUE	MODES	00493
C	MODES	00494
IF(NS.EQ.2) GO TO 2325	MODES	00495
NCH = MYBW	MODES	00496
NC = 1	MODES	00497
GO TO 2350	MODES	00498
2325 CONTINUE	MODES	00499
NC = MYBW + 1	MODES	00500
NCH = MYBW + MYBT	MODES	00501
READ (IVPSC) DEFSL	MODES	00502
2350 CONTINUE	MODES	00503
C	MODES	00504
DO 2500 J=NC,NCH	MODES	00505
IST = IS(J)	MODES	00506
NK = NOC(J) + IST - 1	MODES	00507
JSUM = 0	MODES	00508
ITROW = IST	MODES	00509
IF(.NOT.COPLAN.AND.NS.EQ.2) ITROW = IST + IOVLAP	MODES	00510
DO 2400 I=1,ITROW	MODES	00511
2400 JSUM = JSUM + JOC(I)	MODES	00512
JSUM = JSUM - JOC(ITROW) + 1	MODES	00513
READ (NT5,9015) (DOB(I),I=IST,NK)	MODES	00514
DO 2500 I=IST,NK	MODES	00515
IX = I	MODES	00516
IF(.NOT.COPLAN.AND.NS.EQ.2) IX = I + IOVLAP	MODES	00517
ISUB = JSUM + J - JS(IX) - NC + 1	MODES	00518
DEFSL(1,ISUB) = 1. + GAMC * DOB(I)	MODES	00519
JSUM = JSUM + JOC(IX)	MODES	00520
2500 CONTINUE	MODES	00521
C	MODES	00522
IF(NS.EQ.NSURF) GO TO 2550	MODES	00523
WRITE (IVPSC) DEFSL	MODES	00524
GO TO 2600	MODES	00525
2550 CONTINUE	MODES	00526
CALL RDINIT	MODES	00527
ITYPE = SHIMIXED	MODES	00528
M = 1	MODES	00529
N = NBV	MODES	00530
CALL WRTEX(IFILE,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,1D,	MODES	00531
1 DEFSL, ITYPE,M,N,PARM,IRR)	MODES	00532
IF(IRR.NE.0) GO TO 8080	MODES	00533
C	MODES	00534
2600 CONTINUE	MODES	00535
C	MODES	00536
2700 CONTINUE	MODES	00537
END FILE IFILE	MODES	00538
REWIND IFILE	MODES	00539
2800 CONTINUE	MODES	00540
3000 CONTINUE	MODES	00541

C	8015 FORMAT(7E10.0)	MODES	00542
	RETURN	MODES	00544
C		MODES	00545
C	INPUT DATA ERRORS	MODES	00546
C		MODES	00547
	8000 WRITE (NT6,9000) IDEG	MODES	00548
	GO TO 6199	MODES	00549
	8005 WRITE (NT6,9000) IDEG, NPTS	MODES	00550
	GO TO 6199	MODES	00551
		MODES	00552
C		MODES	00553
C	AN ERROR FROM READING OR WRITING A MATRIX FROM TAPE OR	MODES	00554
C	DISK FILE OCCURRED. PRINT MESSAGES AND FLUSH	MODES	00555
C		MODES	00556
	8010 CONTINUE	MODES	00557
	WRITE (NT6,9010) IGEOSC,IRR	MODES	00558
	WRITE (NT6,9011) MNAME	MODES	00559
	GO TO 6100	MODES	00560
	8020 CONTINUE	MODES	00561
	WRITE (NT6,9020) MODESC,IRR	MODES	00562
	WRITE (NT6,9021) NM	MODES	00563
	GO TO 6100	MODES	00564
C		MODES	00565
	8030 CONTINUE	MODES	00566
	WRITE (NT6,9020) MODESC,IRR	MODES	00567
	WRITE (NT6,9022)	MODES	00568
C		MODES	00569
	GO TO 6100	MODES	00570
	8040 CONTINUE	MODES	00571
	WRITE (NT6,9020) MODESC,IRR	MODES	00572
	WRITE (NT6,9023)	MODES	00573
	GO TO 6100	MODES	00574
C		MODES	00575
	8050 CONTINUE	MODES	00576
	WRITE (NT6,9050) IGEOSC,IRR	MODES	00577
	WRITE (NT6,9051) NM	MODES	00578
	GO TO 6100	MODES	00579
C		MODES	00580
	8060 WRITE (NT6,9050) IALCSC,IRR	MODES	00581
	WRITE (NT6,9051) NM	MODES	00582
	GO TO 6100	MODES	00583
C		MODES	00584
	8070 CONTINUE	MODES	00585
	WRITE (NT6,9070) IALCSC,IRR	MODES	00586
	WRITE (NT6,9071) NM	MODES	00587
	GO TO 6100	MODES	00588
C		MODES	00589
	8080 CONTINUE	MODES	00590
	WRITE (NT6,9080) ITSLS,IRR	MODES	00591
	WRITE (NT6,9081) NSL	MODES	00592
C		MODES	00593
C		MODES	00594
	6100 CONTINUE	MODES	00595
	WRITE (NT6,9101) ID(1),ID(2)	MODES	00596
	WRITE (NT6,9102) PARM,IPARM	MODES	00597
	WRITE (NT6,9103) NFS,NMS	MODES	00598
	WRITE (NT6,9104) ITYPE,M,N	MODES	005

6199 CONTINUE	MODES	00600
WRITE(NT6,9900)	MODES	00601
C	MODES	00602
CALL FLUSH(1)	MODES	00603
C	MODES	00604
7005 FORMAT(*0 PREVIOUS MODES AND GEOMETRY HAVE BEEN SPECIFIED. *)	MODES	00605
7010 FORMAT(86H0*** WARNING - PREVIOUS MODE SHAPES HAVE BEEN SPECIFIED,	MODES	00606
1 BUT GEOMETRY HAS CHANGED. ***)	MODES	00607
7015 FORMAT(80H0*** ERROR - PREVIOUS MODE SHAPES HAVE BEEN SPECIFIED, B	MODES	00608
11(T THE GEOM HAS CHANGED. / 13X,41HPREVIOUS MODE SHAPES WERE AT BOX	MODES	00609
2 CENTERS., 26X,4H ***)	MODES	00610
9000 FORMAT(43H0*** ERROR - SPECIFIED POLYNOMIAL DEGREE OF 15,	MODES	00611
1 22H IS OUTSIDE LIMITS, OR, 16,24H IS TOO MANY POINTS ***)	MODES	00612
9010 FORMAT(53H0*** ERROR - WHILE READING THE GEOMETRY SCRATCH FILE A10	MODES	00613
1, 15H, ERROR CODE = 14,4H ***)	MODES	00614
9011 FORMAT(5X,32HAN ATTEMPT WAS MADE TO READ THE A6, 8H MATRIX.//)	MODES	00615
9020 FORMAT(52H0*** ERROR - WHILE WRITING ON THE MODE SCRATCH FILE A10,	MODES	00616
1 15H, ERROR CODE = 14,4H ***)	MODES	00617
9021 FORMAT(5X,40HAN ATTEMPT WAS MADE TO WRITE MODE SHAPE 13,//)	MODES	00618
9022 FORMAT(5X,47HAN ATTEMPT WAS MADE TO WRITE THE POINTER ARRAY. //)	MODES	00619
9023 FORMAT(5X,41HAN ATTEMPT WAS MADE TO WRITE INDEX ARRAY. //)	MODES	00620
9030 FORMAT(57H0*** ERROR - WHILE WRITING THE COEFFICIENT ARRAY ON FILE	MODES	00621
1 A10,15H, ERROR CODE = 14,4H ***)	MODES	00622
9051 FORMAT(5X,44HAN ATTEMPT WAS MADE TO WRITE FOR MODE SHAPE 14)	MODES	00623
9070 FORMAT(59H0*** ERROR - WHILE READING THE COEFFICIENT ARRAY FROM FI	MODES	00624
1LE A10,15H, ERROR CODE = 14,4H ***)	MODES	00625
9071 FORMAT(5X,43HAN ATTEMPT WAS MADE TO READ FOR MODE SHAPE 14)	MODES	00626
9080 FORMAT(56H0*** ERROR - WHILE WRITING ON THE THICKNESS SLOPE FILE	MODES	00627
1 2X,A10,15H, ERROR CODE = 14,4H ***)	MODES	00628
9081 FORMAT(5X,45HAN ATTEMPT WAS MADE TO WRITE THICKNESS SLOPE 14,//)	MODES	00629
9500 FORMAT(1H1,8A10, // 46X,* MODE SHAPE NUMBER *,13,	MODES	00630
1 / 46X,***CH NUMBER =*,F11.6,/46X,24(1H-). /)	MODES	00631
9101 FORMAT(5X,***MATRIX ID = *, A10, I10)	MODES	00632
9102 FORMAT(5X,***PARAMETERS *,10E11.3, /10X,*(INTEGER)*, 17, 9111)	MODES	00633
9103 FORMAT(5X,***FILE SPACING = *,13,* MATRIX SPACING = *,13)	MODES	00634
9104 FORMAT(5X,***MATRIX TYPE =*,A10,*, DIMENSIONED (*14,2H X,14,1H))	MODES	00635
9900 FORMAT(*0 ERROR OCCURRED IN MODES SECTION (MAIN PROGRAM).*)	MODES	00636
END	MODES	00637

SUBROUTINE TAPMOD(NS,NM,DEFSL)	TAPMOD 00002
DIMENSION DEFSL(2,500), D(500), S(500)	TAPMOD 00003
C	TAPMOD 00004
C THIS PROGRAM WILL READ MODE SHAPES FROM TAPE IN THE SAME	TAPMOD 00005
C FORMAT AS PROGRAM TEV059. THE PROGRAM WILL BE REPLACED BY	TAPMOD 00006
C THE AIR FORCE FOR ITS USE AND BOEING MAY MODIFY OR REPLACE	TAPMOD 00007
C AS INPUT TAPES ARE MODIFIED OR REPLACED.	TAPMOD 00008
C	TAPMOD 00009
C NS - =1, FIRST PLANFORM	TAPMOD 00010
C =2, SECOND PLANFORM	TAPMOD 00011
C NM - MODE SHAPE NUMBER	TAPMOD 00012
C DEFSL - ARRAY WHERE MODE SHAPES FOR ENTIRE PLANFORM(S) IS	TAPMOD 00013
C STORED.	TAPMOD 00014
C	TAPMOD 00015
COMMON /GEOMTY/ COPLAN, NSUBDV, XSUBDV, NSUBD2, NSUBCN, NSURF,	GEOMTY 00002
1 B1, B1BETA, B1S, B1BTAS, WLAX, WLAZ, PSIW,	GEOMTY 00003
2 MXBW, MXBBW, MYBW, MYBBW, MXBSW, MYBSW, MYBDSW,	GEOMTY 00004
3 IXBW, XCENR	GEOMTY 00005
LOGICAL COPLAN	GEOMTY 00006
COMMON /GEOM2 / TLAX, TLAZ, PSIT, MXBT, MYBT, MYBBT, MXBST, MYBST,	GEOM2 00002
1 MYBBST, IXBT, IXBST, CAPL	GEOM2 00003
COMMON /FILES / NT5, NT6, INTAPE, INFSP, NPLAIC, NSFAIC, NOUTP,	FILES 00002
1 IOUFSP, MODESC, IVPSC, IGEOC, IWTFC, IAICSC	FILES 00003
COMMON /TAPEIO/ NFS, NMS, LS, NMR, ID(20), NID, ITYPE, LRS, LWS, M, N,	TAPEIO 00002
1 PARM(10), IRR	TAPEIO 00003
DIMENSION IFARM(10)	TAPEIO 00004
EQUIVALENCE (FARM, IFARM)	TAPEIO 00005
COMMON /INDEX/ IS(100), NOC(100), JS(100), JOC(100)	TAPMOD 00020
C	TAPMOD 00021
LOGICAL MREAD, RANDIN	TAPMOD 00022
MREAD = .FALSE.	TAPMOD 00023
RANDIN = .FALSE.	TAPMOD 00024
C	TAPMOD 00025
CALL RDINIT	TAPMOD 00026
IF(NM.NE.1) GO TO 710	TAPMOD 00027
IF(NS.EQ.1) REWIND INTAPE	TAPMOD 00028
NMS = 2	TAPMOD 00029
NFS = INFSP	TAPMOD 00030
710 CONTINUE	TAPMOD 00031
ITYPE = 5H MIXED	TAPMOD 00032
CALL READMX(INTAPE, MREAD, RANDIN, NFS, NMS, LS, NMR, 1, NID, ID, ITYPE,	TAPMOD 00033
1 LRS, D, M, N, FARM, IRR)	TAPMOD 00034
IF(IRR.NE.0) GO TO 6020	TAPMOD 00035
CALL RDINIT	TAPMOD 00036
ITYPE = 5H MIXED	TAPMOD 00037
CALL READMX(INTAPE, MREAD, RANDIN, NFS, NMS, LS, NMR, 1, NID, ID, ITYPE,	TAPMOD 00038
1 LRS, S, M, N, FARM, IRR)	TAPMOD 00039
IF(IRR.NE.0) GO TO 6020	TAPMOD 00040
C	TAPMOD 00041
IF(NS.EQ.2) GO TO 720	TAPMOD 00042
C	TAPMOD 00043
C 1 RST PLANFORM	TAPMOD 00044
NCH = MYBW	TAPMOD 00045
NC = 1	TAPMOD 00046
GO TO 725	TAPMOD 00047
C	TAPMOD 00048
C SECOND PLANFORM	TAPMOD 00049

720 CONTINUE	TAPMOD 00050
NC = MYBW + 1	TAPMOD 00051
NCH = MYBW + MYBT	TAPMOD 00052
C	TAPMOD 00053
C STORE DEFLECTIONS AND SLOPES	TAPMOD 00054
725 CONTINUE	TAPMOD 00055
ITS = 0	TAPMOD 00056
DO 750 J=NC,NCH	TAPMOD 00057
IST = IS(J)	TAPMOD 00058
NK = NOC(J) + IST - 1	TAPMOD 00059
JSUM = 0	TAPMOD 00060
DO 730 I=1,IST	TAPMOD 00061
730 JSUM = JSUM + JOC(I)	TAPMOD 00062
JSUM = JSUM - JOC(IST) + 1	TAPMOD 00063
DO 750 I=IST,NK	TAPMOD 00064
ITS = ITS + 1	TAPMOD 00065
ISUB = JSUM + J - NC + 1 - JS(I)	TAPMOD 00066
DEFSL(1,ISUB) = D(ITS)	TAPMOD 00067
DEFSL(2,ISUB) = S(ITS)	TAPMOD 00068
JSUM = JSUM + JOC(I)	TAPMOD 00069
750 CONTINUE	TAPMOD 00070
C	TAPMOD 00071
C RETURN	TAPMOD 00072
C	TAPMOD 00073
C AN ERROR DURING READING A MATRIX FROM TAPE OR	TAPMOD 00074
C DISK FILE OCCURRED. PRINT MESSAGES AND FLUSH	TAPMOD 00075
C	TAPMOD 00076
8010 CONTINUE	TAPMOD 00077
8020 CONTINUE	TAPMOD 00078
WRITE (NT6,9020) MODESC,IRR	TAPMOD 00079
WRITE (NT6,9021) NM	TAPMOD 00080
8100 CONTINUE	TAPMOD 00081
WRITE(NT6,9101) ID(1),ID(2)	TAPMOD 00082
WRITE(NT6,9102) FARM,IFARM	TAPMOD 00083
WRITE(NT6,9103) NFS,NMS	TAPMOD 00084
WRITE(NT6,9104) ITYPE,M,N	TAPMOD 00085
WRITE(NT6,9900)	TAPMOD 00086
C	TAPMOD 00087
C CALL FLUSH(1)	TAPMOD 00088
C	TAPMOD 00089
9020 FORMAT(54H0*** ERROR - WHILE READING FROM THE MODE SCRATCH FILE	TAPMOD 00090
1 A10,15H, ERROR CODE = I4,4H ***)	TAPMOD 00091
9021 FORMAT(5X,39HAN ATTEMPT WAS MADE TO READ MODE SHAPE I3,//)	TAPMOD 00092
9101 FORMAT(5X, *MATRIX ID = *, A10, I10)	TAPMOD 00093
9102 FORMAT(5X, *PARAMETERS *,10E11.3, /10X,*(INTEGER)*, I7, 9111)	TAPMOD 00094
9103 FORMAT(5X, *FILE SPACING = *,I3, * MATRIX SPACING = *,I3)	TAPMOD 00095
9104 FORMAT(5X, *MATRIX TYPE -*,A10,*, DIMENSIONED (*I4,2H X,I4,1H))	TAPMOD 00096
9900 FORMAT(40 ERROR OCCURRED IN MODES SECTION (SUBROUTINE TAPMOD). * FTNX1	00044
1)	FTNX1 00045
END	TAPMOD 00098

	SUBROUTINE ROPER	ROPER	00002
C		ROPER	00003
C	SUBROUTINE TO DETERMINE THE BOXES ON EACH ROW THAT SHOULD	ROPER	00004
C	HAVE HOPE SHAPES.	ROPER	00005
C		ROPER	00006
C	IS(J) - ROW INDES OF FIRST PLANFORM BOX FOR CHORD J.	ROPER	00007
C	NOC(J) - NUMBER OF PLANFORM BOXES ON CHORD J	ROPER	00008
C	JS(I) - COL INDEX OF FIRST PLANFORM BOX FOR SPAN I.	ROPER	00009
C	JOC(I) - NUMBER OF BOXES BETWEEN FIRST AND LAST PLANFORM	ROPER	00010
C	BOX ON SPAN I	ROPER	00011
C		ROPER	00012
C		ROPER	00013
C	COMMON /GEOMTY/ COPLAN, NSUBDV, XSUBDV, NSUBD2, NSUBCN, NSURF,	GEOMTY	00002
1	B1, B1BETA, B1S, B1BTAS, WLAX, WLAZ, PSIW,	GEOMTY	00003
2	MXBW, MXBBW, MYBW, MYBBW, MXBSW, MYBSW, MYBBSW,	GEOMTY	00004
3	IXBW, XCENR	GEOMTY	00005
	LOGICAL COPLAN	GEOMTY	00006
	COMMON /GEOM2 / TLAX, TLAZ, PSIT, MXBT, MYBT, MYBBT, MXBST, MYBST,	GEOM2	00002
1	MYBBST, IXBT, IXBST, CARL	GEOM2	00003
	COMMON /INDEX/ IS(100), NOC(100), JS(100), JOC(100)	ROPER	00016
	IF(COPLAN) GO TO 100	ROPER	00017
	NCH = MYBW	ROPER	00018
	MXB = MXBW	ROPER	00019
	GO TO 200	ROPER	00020
100	CONTINUE	ROPER	00021
	NCH = MYBW + MYBT	ROPER	00022
	MXB = MXBT	ROPER	00023
200	CONTINUE	ROPER	00024
	DO 500 I = 1, MXB	ROPER	00025
	JS(I) = 0	ROPER	00026
	JOC(I) = 0	ROPER	00027
	JCUT = 0	ROPER	00028
	DO 400 J = 1, NCH	ROPER	00029
	IF(I.LT.IS(J)) GO TO 400	ROPER	00030
	ILAST = IS(J) + NOC(J) - 1	ROPER	00031
	IF(I.GT.ILAST) GO TO 400	ROPER	00032
	IF(JS(I).NE.0) GO TO 300	ROPER	00033
	JS(I) = J	ROPER	00034
	IF(J.GT.MYBW) JS(I) = J - MYBW	ROPER	00035
300	CONTINUE	ROPER	00036
	IF(JCUT.EQ.1) GO TO 400	ROPER	00037
	JV = J	ROPER	00038
	IF(J.GT.MYBW) JV = J - MYBW	ROPER	00039
	IF(JV.LT.JS(I)) GO TO 350	ROPER	00040
	JOC(I) = JV - JS(I) + 1	ROPER	00041
	GO TO 400	ROPER	00042
350	CONTINUE	ROPER	00043
	JOC(I) = JS(I) - JV + JOC(I)	ROPER	00044
	JS(I) = JV	ROPER	00045
	JCUT = 1	ROPER	00046
400	CONTINUE	ROPER	00047
425	CONTINUE	ROPER	00048
500	CONTINUE	ROPER	00049
C		ROPER	00050
C	CALCULATE FOR SECOND PLANFORM. THIS IS ONLY USED FOR	ROPER	00051
C	NON COPLANAR PLANFORMS.	ROPER	00052
	IF(NSURF.EQ.1) GO TO 1500	ROPER	00053

```

IF(COPLAN:      GO TO 1500
NCH = MYBW + MYBT
JCH = MYBW + 1
IFBT = (IXBT-IXBW)/NSUBDV +1
IOMLAP = 0
IF(IFBT.LE.MXBW) IOMLAP = MXBW - IFBT + 1
DO 1000 I=IFBT,MXBT
IX = I + IOMLAP
JS(IX) = 0
JOC(IX) = 0
DO 800 J=JCH,NCH
IF(I.LT.IS(J)) GO TO 800
ILAST = IS(J) + NOC(J) -1
IF(I.GT.ILAST) GO TO 800
IF(JS(IX).EQ.0) JS(IX) = J-MYBW
JV = J- MYBW
JOC(IX) = JV - JS(IX) + 1
800 CONTINUE
825 CONTINUE
1000 CONTINUE
1500 CONTINUE
RETURN
END

```

```

ROPER 00054
ROPER 00055
ROPER 00056
ROPER 00057
ROPER 00058
ROPER 00059
ROPER 00060
ROPER 00061
ROPER 00062
ROPER 00063
ROPER 00064
ROPER 00065
ROPER 00066
ROPER 00067
ROPER 00068
ROPER 00069
ROPER 00070
ROPER 00071
ROPER 00072
ROPER 00073
ROPER 00074
ROPER 00075
ROPER 00076

```

SUBROUTINE FITTER(M,N,X,Y,Z,C,CN,IDIM)	FITTER 00002
DIMENSION X(100), Y(100), Z(IDIM,100), C(IDIM,66)	FITTER 00003
DIMENSION AI(66), A(66,66), XP(11), YP(11)	FITTER 00004
DIMENSION VS(10)	FITTER 00005
LOGICAL COMPLX	FITTER 00006
C	FITTER 00007
C	FITTER 00008
C	FITTER 00009
C	FITTER 00010
C	FITTER 00011
C	FITTER 00012
C	FITTER 00013
C	FITTER 00014
C	FITTER 00015
C	FITTER 00016
C	FITTER 00017
C	FITTER 00018
C	FITTER 00019
C	FITTER 00020
C	FITTER 00021
C	FITTER 00022
C	FITTER 00023
C	FITTER 00024
C	FITTER 00025
C	FITTER 00026
C	FITTER 00027
C	FITTER 00028
C	FITTER 00029
C	FITTER 00030
C	FITTER 00031
C	FITTER 00032
C	FITTER 00033
C	FITTER 00034
C	FITTER 00035
C	FITTER 00036
C	FITTER 00037
C	FITTER 00038
C	FITTER 00039
C	FITTER 00040
C	FITTER 00041
C	FITTER 00042
C	FITTER 00043
C	FITTER 00044
C	FITTER 00045
C	FITTER 00046
C	FITTER 00047
C	FITTER 00048
C	FITTER 00049
C	FITTER 00050
C	FITTER 00051
C	FITTER 00052
C	FITTER 00053
C	FITTER 00054
C	FITTER 00055
C	FITTER 00056
C	FITTER 00057
C	FITTER 00058

```

SUBROUTINE FITTER(M,N,X,Y,Z,C,CN,IDIM)
  DIMENSION X(100), Y(100), Z(IDIM,100), C(IDIM,66)
  DIMENSION AI(66), A(66,66), XP(11), YP(11)
  DIMENSION VS(10)
  LOGICAL COMPLX

C
C      M - DEGREE OF POLYNOMIAL EQUATION
C      N - NUMBER OF DATA POINTS TO FIT CURVE THROUGH
C      X - X COORDINATE OF DATA POINT
C      Y - Y COORDINATE OF DATA POINT
C      Z - Z COORDINATE OF DATA POINT
C      C - OUTPUT COEFFICIENT ARRAY
C      CN - SCALE FACTOR
C      CN - SCALE FACTOR
C      IDIM - INDICATOR OF REAL OR COMPLEX FUNCTION
C              = 1, FUNCTION IS REAL
C              = 2, FUNCTION IS COMPLEX
C              IF COMPLEX SET DIMENSIONS OF FUNCTION AND COEFFICIENTS
C                  TO (IDIM * --- )
C
C      DETERMINE NUMBER OF COEFFICIENTS
C
  EPS = 1.0E-04
  COMPLX = .FALSE.
  IF(IDIM.EQ.2) COMPLX = .TRUE.

C
C      SCALE DATA TO REDUCE MAGNITUDE OF MATRIX TERMS.
C      SHOULD AVOID BOMB OUTS DUE TO OVERFLOW CONDITIONS.
C
  IF(CN.EQ.0) CN=1.0
  IF(CN.EQ.1.0) GO TO 15
  DO 5 I=1,N
    X(I) = X(I)/CN
    Y(I) = Y(I)/CN
  5 CONTINUE
  15 CONTINUE
  XM = M / 4
  XM2= XM/2.
  NC = XM*XM2 + XM2 + EPS
  IF(NC.LE.0) GO TO 25
  M = M-1
  GO TO 15
  25 CONTINUE

C
  MAC = NC

C
C      DETERMINE THE MAXIMUM DEGREE THAT CAN BE COMPUTED IN
C      EACH DIRECTION AND SET UP ORDER OF SOLUTION.
C
  NDV = 1
  MDX = M
  VS(1) = X(1)
  DO 6 I=1,N
    DO 5 J=1,NDV
      IF(X(I).EQ.VS(J)) GO TO 55
    50 CONTINUE
    NDV = NDV + 1
    VS(NDV) = X(I)

```

IF(MCV-1.EQ.M) GO TO 65	FITTER 00059
55 CONTINUE	FITTER 00060
60 CONTINUE	FITTER 00061
MDX = MCV -1	FITTER 00062
65 CONTINUE	FITTER 00063
C	FITTER 00064
NDV = 1	FITTER 00065
MDY = M	FITTER 00066
VS(1) = Y(1)	FITTER 00067
DO 80 I=1,N	FITTER 00068
DO 70 J=1,NDV	FITTER 00069
IF(Y(I).EQ.VS(J)) GO TO 75	FITTER 00070
70 CONTINUE	FITTER 00071
NDV = NDV + 1	FITTER 00072
VS(NDV) = Y(I)	FITTER 00073
IF(MCV-1.EQ.M) GO TO 85	FITTER 00074
75 CONTINUE	FITTER 00075
80 CONTINUE	FITTER 00076
MDY = MDV - 1	FITTER 00077
85 CONTINUE	FITTER 00078
C	FITTER 00079
ITOT = NC +1	FITTER 00080
ITOT1 = ITOT	FITTER 00081
IF(COMPLX) ITOT = ITOT + 1	FITTER 00082
C	FITTER 00083
C	FITTER 00084
C	FITTER 00085
DO 95 I=1,NC	FITTER 00086
C(1,I) = 0.0	FITTER 00087
IF(.NOT.COMPLX) GO TO 90	FITTER 00088
C(2,I) = 0.0	FITTER 00089
90 CONTINUE	FITTER 00090
DO 95 J=1,ITOT	FITTER 00091
95 A(I,J) = 0.0	FITTER 00092
C	FITTER 00093
C	FITTER 00094
C	FITTER 00095
AI(1) =1.0	FITTER 00096
XP(1) =1.0	FITTER 00097
YP(1) =1.0	FITTER 00098
MM = M + 1	FITTER 00099
DO 200 K=1,N	FITTER 00100
DO 10 L=2,MM	FITTER 00101
XP(L) = XP(L-1)*X(K)	FITTER 00102
YP(L) = YP(L-1)*Y(K)	FITTER 00103
10 CONTINUE	FITTER 00104
C	FITTER 00105
I = 1	FITTER 00106
DO 40 L=2,MM	FITTER 00107
DO 20 LL=1,L	FITTER 00108
IL= L - LL +1	FITTER 00109
IF(LL-1.GT.MDY) GO TO 30	FITTER 00110
IF(IL-1.GT.MDX) GO TO 20	FITTER 00111
I = I + 1	FITTER 00112
AI(I) = XP(IL)*YP(LL)	FITTER 00113
20 CONTINUE	FITTER 00114
30 CONTINUE	FITTER 00115

40 CONTINUE	FITTER 00116
AI(I+1) = Z(1,K)	FITTER 00117
IF(COMPLX) AI(I+2) = Z(2,K)	FITTER 00118
IF(K.GT.1) GO TO 45	FITTER 00119
NC = I	FITTER 00120
ITOT = NC + 1	FITTER 00121
ITOT1 = ITOT	FITTER 00122
IF(COMPLX) ITOT = ITOT + 1	FITTER 00123
45 CONTINUE	FITTER 00124
C	FITTER 00125
DO 1100 I=1,NC	FITTER 00126
DO 1100 J=I,ITOT	FITTER 00127
ASAV = AI(I)*AI(J)	FITTER 00128
A(I,J)=A(I,J)+ASAV	FITTER 00129
1100 CONTINUE	FITTER 00130
200 CONTINUE	FITTER 00131
C	FITTER 00132
C SQUARE ROOT METHOD	FITTER 00133
C INTERMEDIATE MATRIX	FITTER 00134
DO 1200 I=1,NC	FITTER 00135
IMI = I-1	FITTER 00136
TMP=0.0	FITTER 00137
IF(I.EQ.1) GO TO 1150	FITTER 00138
DO 1120 L=1,IMI	FITTER 00139
1120 TMP= TMP+ A(L,I)**2	FITTER 00140
1150 CONTINUE	FITTER 00141
T = A(I,I) - TMP	FITTER 00142
IF(T.GT.EPS) GO TO 4	FITTER 00143
A(I,I) = 0.0	FITTER 00144
GO TO 1200	FITTER 00145
4 CONTINUE	FITTER 00146
A(I,I) = SQRT(T)	FITTER 00147
IF(A(I,I).GT.EPS) GO TO 1155	FITTER 00148
A(I,ITOT) = 0.0	FITTER 00149
GO TO 1200	FITTER 00150
1155 CONTINUE	FITTER 00151
C	FITTER 00152
JS = I+1	FITTER 00153
DO 1180 L = JS,ITOT	FITTER 00154
TMP= 0.0	FITTER 00155
IF(I.EQ.1) GO TO 1175	FITTER 00156
DO 1160 L=1,IMI	FITTER 00157
1160 TMP = TMP + A(L,I)*A(L,J)	FITTER 00158
1175 A(I,J) =(A(I,J)-TMP)/A(I,I)	FITTER 00159
1180 CONTINUE	FITTER 00160
1200 CONTINUE	FITTER 00161
C	FITTER 00162
C	FITTER 00163
C BACK SUBSTITUTE FOR COEFFICIENTS	FITTER 00164
DO 1400 K=1,NC	FITTER 00165
I = NC - K + 1	FITTER 00166
IP1=I 1	FITTER 00167
TMP1 = 0.0	FITTER 00168
TMP2 = 0.0	FITTER 00169
IF(A(I,I).GT.EPS) GO TO 1325	FITTER 00170
C(I,I) = 0.0	FITTER 00171
IF(COMPLX) C(2,I) = 0.0	FITTER 00172

```

      GO TO 1400
1325 CONTINUE
      IF(I.EQ.NC) GO TO 1375
      DO 1350 L=IP1,NC
      TMP1 = TMP1 + A(I,L)*C(1,L)
      IF(.NOT.COMPLX) GO TO 1350
      TMP2 = TMP2 + A(I,L)*C(2,L)
1350 CONTINUE
1375 CONTINUE
      C(1,I) = (A(I,ITOT1)-TMP1)/A(I,I)
      IF(.NOT.COMPLX) GO TO 1400
      C(2,I) = (A(I,ITOT) -TMP2)/A(I,I)
1400 CONTINUE
C
C
C      REORDER THE COEFFICIENTS IN CORRECT POWERS
C      OF X AND Y.
C
      IF(NAC.EQ.NC) GO TO 1475
C
      IZ = 1
      I = 1
      DO 1440 L=2,MN
      DO 1420 LL=1,L
      IL = L -LL +1
      I = I +1
      IF(LL-1.LE.MDY.AND.IL-1.LE.MDX) GO TO 1410
      X(I) = 0.0
      Y(I) = 0.0
      GO TO 1420
1410 CONTINUE
      IZ = IZ + 1
      X(I) = C(1,IZ)
      IF(COMPLX) Y(I) = C(2,IZ)
1420 CONTINUE
1440 CONTINUE
C
      DO 1450 I=2,NAC
      C(1,I) = X(I)
      IF(COMPLX) C(2,I) = Y(I)
1450 CONTINUE
1475 CONTINUE
C
C      ELIMINATE THE SCALE FACTOR FROM THE COEFFICIENTS.
C
      IF(CN.EQ.1.0) GO TO 1700
      I=1
      CP= 1.0/CN
      DO 1600 L1=2,MN
      DO 1500 L2=1,L1
      I = I+1
      C(1,I) = C(1,I)*CP
      C(2,I) = C(2,I)*CP
1500 CONTINUE
      CP= CP/CN
1600 CONTINUE
1700 CONTINUE

```

```

FITTER 00173
FITTER 00174
FITTER 00175
FITTER 00176
FITTER 00177
FITTER 00178
FITTER 00179
FITTER 00180
FITTER 00181
FITTER 00182
FITTER 00183
FITTER 00184
FITTER 00185
FITTER 00186
FITTER 00187
FITTER 00188
FITTER 00189
FITTER 00190
FITTER 00191
FITTER 00192
FITTER 00193
FITTER 00194
FITTER 00195
FITTER 00196
FITTER 00197
FITTER 00198
FITTER 00199
FITTER 00200
FITTER 00201
FITTER 00202
FITTER 00203
FITTER 00204
FITTER 00205
FITTER 00206
FITTER 00207
FITTER 00208
FITTER 00209
FITTER 00210
FITTER 00211
FITTER 00212
FITTER 00213
FITTER 00214
FITTER 00215
FITTER 00216
FITTER 00217
FITTER 00218
FITTER 00219
FITTER 00220
FITTER 00221
FITTER 00222
FITTER 00223
FITTER 00224
FITTER 00225
FITTER 00226
FITTER 00227
FITTER 00228
FITTER 00229

```

C
C THE C ARRAY NOW CONTAINS THE COEFFICIENTS.
C
RETURN
END

FITTER 00230
FITTER 00231
FITTER 00232
FITTER 00233
FITTER 00234

SUBROUTINE MODOUT(DEFSL,JS,JOC,NROWS,NM,IOLAP)	MODOUT	00002
DIMENSION DEFSL(2,500),JS(100),JOC(100)	MODOUT	00003
DIMENSION JPH(15),DS(50)	MODOUT	00004
COMMON /GEOMTY/ COPLAN,NSUBDV,XSUBDV,NSUBD2,NSUBCN,NSURF,	GEOMTY	00002
1 B1,B1BETA,B1S,B1BTAS,WLAX,WLAZ,PSIW,	GEOMTY	00003
2 MXBW,MXBW,MYBW,MYBBW,MXBSW,MYBSW,MYBBSW,	GEOMTY	00004
3 IXBW,XCENTR	GEOMTY	00005
LOGICAL COPLAN	GEOMTY	00006
COMMON /FILES / NTS,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUTP,	FILES	00002
1 IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC	FILES	00003
COMMON /PROBLM/ XHACH,NMODES,NTSLOP,NKVALS,SMOOTH,NDEG,CREDIT,	PROBLM	00002
1 EXAIC,SUBDV,PLYWOOD	PROBLM	00003
LOGICAL SMOOTH,CREDIT,EXAIC,SUBDV,PLYWOOD	PROBLM	00004
DIMENSION SLOMAT(3)	MODOUT	00008
EQUIVALENCE (SLOMAT(2), IMAT)	FTMOL	00046
DATA SLOMAT/10H(13,3X, 10H 0,10HP,15F7.3) /	MODOUT	00009
DATA IMAT1,IMAT2 /10H 0, 10H -0 /	MODOUT	00011
C MODOUT WILL PRINT THE MODE SHAPES OUT IN ROW/COLUMN FORM	MODOUT	00012
C	MODOUT	00013
C	MODOUT	00014
C DEFSL - MOD SHAPES IN INTERNAL STORAGE	MODOUT	00015
C JS(I) - FIRST COLUMN FOR WHICH THERE IS A MODE SHAPE ON ROW I	MODOUT	00016
C JOC(I) - NUMBER OF BOXES BETWEEN FIRST AND LAST PLANFORM	MODOUT	00017
C BOX ON ROW I	MODOUT	00018
C NROWS - NUMBER OF ROWS	MODOUT	00019
C NM - MODE SHAPE NUMBER	MODOUT	00020
C IOLAP - NUMBER OF BOXES OVERLAP BETWEEN PLANFORMS FOR	MODOUT	00021
C NON-COPLANAR SURFACES	MODOUT	00022
C	MODOUT	00023
C IMAT = 10H 0	MODOUT	00024
C NBETS = 0	MODOUT	00025
C DO 90 I=1,NROWS	MODOUT	00026
C JL = JS(I)+JOC(I)-1	MODOUT	00027
C IF(NBETS.LT.JL) NBETS=JL	MODOUT	00028
C 90 CONTINUE	MODOUT	00029
C JBETS = (NBETS-1)/15 + 1	MODOUT	00030
C DO 1000 NP=1,2	MODOUT	00031
C	MODOUT	00032
C FIND LARGEST VALUE	MODOUT	00033
C	MODOUT	00034
C	MODOUT	00035
C VALUE = 0.0	MODOUT	00036
C DO 100 L=1,500	MODOUT	00037
C AVAL = ABS(DEFSL(NP,L))	MODOUT	00038
C IF(AVAL.GT.VALUE) VALUE = AVAL	MODOUT	00039
C 100 CONTINUE	MODOUT	00040
C POW = 1	MODOUT	00041
C TSCALE = 10.	MODOUT	00042
C IF(VALUE.GE.10.) TSCALE = 0.1	MODOUT	00043
C DO 110 N=1,8	MODOUT	00044
C POW = POW *TSCALE	MODOUT	00045
C TTEN = VALUE * POW	MODOUT	00046
C IF(TTEN.GE.10.) GO TO 110	MODOUT	00047
C IF(TTEN.LT.1.) GO TO 110	MODOUT	00048
C NPA = N	MODOUT	00049
C GO TO 115	MODOUT	00050
C 110 CONTINUE	MODOUT	00051
C NPA = 0		

115 CONTINUE	MODOUT 00052
IF(VALUE.LE.1.00) GO TO 120	MODOUT 00053
C	MODOUT 00054
C THE ARRAY MUST BE SCALED DOWN.	MODOUT 00055
C	MODOUT 00056
NPX = -NPA	MODOUT 00057
IMAT = IMAT2	MODOUT 00058
GO TO 122	MODOUT 00059
C	MODOUT 00060
C THE ARRAY MUST BE SCALED UP.	MODOUT 00061
C	MODOUT 00062
120 CONTINUE	MODOUT 00063
NPX = NPA	MODOUT 00064
IMAT = IMAT1	MODOUT 00065
122 CONTINUE	MODOUT 00066
IMAT = IMAT + NPA	MODOUT 00067
IF(NP.EQ.2) GO TO 124	MODOUT 00068
WRITE (NT6,9005) NPX	MODOUT 00069
GO TO 125	MODOUT 00070
124 CONTINUE	MODOUT 00071
WRITE (NT6,9010) NPX	MODOUT 00072
125 CONTINUE	MODOUT 00073
DO 900 JPS=1,JSETS	MODOUT 00074
JBASE = (JPS-1)*15	MODOUT 00075
DO 150 JC=1,15	MODOUT 00076
150 JFH(JC) = JC + JBASE	MODOUT 00077
JL=15	MODOUT 00078
IF(JPS.EQ.JSETS) JL= NSETS - 15*(JSETS-1)	MODOUT 00079
WRITE(NT6,9015) (JFH(J) ,J=1,JL)	MODOUT 00080
IOV = 0	MODOUT 00081
ITOT = 0	MODOUT 00082
DO 800 I=1,NROWS	MODOUT 00083
C	MODOUT 00084
C ZERO OUT PRINT ARRAY	MODOUT 00085
DO 200 J=1,50	MODOUT 00086
200 DS(J) = 0.0	MODOUT 00087
C	MODOUT 00088
C PUT THE VALUES INTO PRINT ARRAY	MODOUT 00089
J1 = JS(I)	MODOUT 00090
IF(J1.EQ.0) GO TO 800	MODOUT 00091
JL = JS(I) +JOC(I) -1	MODOUT 00092
DO 300 J=J1,JL	MODOUT 00093
ITOT = ITOT+1	MODOUT 00094
300 DS(J) = DEFSL(NP,ITOT)	MODOUT 00095
C	MODOUT 00096
C PRINT THE ONES IN THIS SET	MODOUT 00097
JIP=(JPS-1)*15 +1	MODOUT 00098
JIL= JIP +14	MODOUT 00099
IF(JL.LT.JIP) GO TO 800	MODOUT 00100
IF(JI.GT.JIL) GO TO 800	MODOUT 00101
IF(JIL.GT.JL) JIL=JL	MODOUT 00102
M = 1	MODOUT 00103
IF(I /LAP.EQ.0) GO TO 350	MODOUT 00104
IF(I LE.MXBW) GO TO 350	MODOUT 00105
M = I-IOVLAP	MODOUT 00106
IF(IOV.NE.0) GO TO 350	MODOUT 00107
IOV = 1	MODOUT 00108

WRITE(NT6,9015)	MODOUT 00109
350 CONTINUE	MODOUT 00110
WRITE(NT6,SLOMAT) M, (DS(J), J=JIP, JIL)	MODOUT 00111
800 CONTINUE	MODOUT 00112
900 CONTINUE	MODOUT 00113
1000 CONTINUE	MODOUT 00114
RETURN	MODOUT 00115
9005 FORMAT(1H0,46X,*DEFLECTIONS X 1.0E *,I2,/ 47X,21(1H-))	MODOUT 00116
9010 FORMAT(1H0,///47X,* SLOPES X 1.0E *,I2,/ 47X,21(1H-))	MODOUT 00117
9015 FORMAT(1H0,5X,15(14,3X))	MODOUT 00118
9020 FORMAT(13,3X,15F7.3)	MODOUT 00119
END	MODOUT 00120

SUBROUTINE PRECOF(IDEQ,A,IFR)	PRECOF 00002
COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUPT,	FILES 00002
1 IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC	FILES 00003
DIMENSION A(21),BLNK(7),BNK2(6)	PRECOF 00004
EQUIVALENCE (BLNK(2),BNK2(1))	PRECOF 00006
DIMENSION IXP(7),IYP(7)	PRECOF 00007
DATA BLNK / 7* 1H /	FTNX1 00047
C	PRECOF 00008
C THIS SUBROUTINE PRINTS THE COEFFICIENTS USED IN THE	PRECOF 00009
C POLYNOMIAL EQUATION USED IN CALCULATION OF MODE SHAPES	PRECOF 00010
C	PRECOF 00011
C IDEQ - DEGREE OF POLYNOMIAL EQUATION	PRECOF 00012
C A - ARRAY OF COEFFICIENTS	PRECOF 00013
C IFR - FLAG INDICATING HOW COEFFICIENTS ARE OBTAINED	PRECOF 00014
C =1, READ FROM CARDS	PRECOF 00015
C =2, FROM LEAST SQUARES SURFACE FIT	PRECOF 00016
C	PRECOF 00017
IF(IFR.EQ.1) WRITE(NT6,9055) A(1)	PRECOF 00018
IF(IFR.EQ.2) WRITE(NT6,9065) A(1)	PRECOF 00019
C	PRECOF 00020
IF(IDEQ.EQ.0) GO TO 550	PRECOF 00021
IDEX = 2	PRECOF 00022
DO 520 I=1,IDEQ	PRECOF 00023
NCL = I+1	PRECOF 00024
LDEX = IDEX + 1	PRECOF 00025
DO 541 NXP=1,NCL	PRECOF 00026
IXP(NXP) = NCL-NXP	PRECOF 00027
IYP(NXP) = NXP-1	PRECOF 00028
541 CONTINUE	PRECOF 00029
WRITE(NT6,9060) (BLNK(NXP),IXP(NXP),IYP(NXP),NXP=1,NCL)	PRECOF 00030
WRITE(NT6,9061) (BNK2(NXP),NXP=1,NCL)	PRECOF 00031
WRITE(NT6,9062) (A(J),J=IDEX,LDEX)	PRECOF 00032
IDEX = LDEX + 1	PRECOF 00033
520 CONTINUE	PRECOF 00034
550 CONTINUE	PRECOF 00035
RETURN	PRECOF 00036
9055 FORMAT(13X,*(MODAL POLYNOMIAL COEFFICIENTS* 20X,*FROM CARD INPUT*	PRECOF 00037
1 /13X,29(1H-) /*0 CONSTANT*/2X,10(1H-) / E12.4)	PRECOF 00038
9060 FORMAT(1HD,6(A1,4HX ** I1,4HY ** I1,3X))	PRECOF 00039
9061 FORMAT(1X,6(A1,10H-----,3X))	PRECOF 00040
9062 FORMAT(E12.4,5E14.4)	PRECOF 00041
9065 FORMAT(1HD,12X,*(MODAL POLYNOMIAL COEFFICIENTS*/13X,*BY LEAST SQUAR	PRECOF 00042
1ES SURFACE FIT*/13X,29(1H-)/*0 CONSTANT*/2X,10(1H-) / E12.4)	PRECOF 00043
END	PRECOF 00044

ITYPE = SMIXED	VICMAIN 00095
MXARRY = GHATAB	VICMAIN 00096
CALL READMX(NSPAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,500,NID,ID,ITYPE,	VICMAIN 00097
1 LRS, ATAB,M,N,PARM,IRR)	VICMAIN 00098
IF(IRR.NE.0) GO TO 6060	VICMAIN 00099
REWIND NSPAIC	VICMAIN 00100
C	VICMAIN 00101
IF(INV.NE.1) GO TO 555	VICMAIN 00102
C	VICMAIN 00103
C PRINT THE TABLE OF CONTENTS	VICMAIN 00104
WRITE (NT6,9215)	VICMAIN 00105
9215 FORMAT(1H1,14X,*SPATIAL AIC TAPE TABLE OF CONTENTS* /15X,34(1H-),	VICMAIN 00106
1 // 5X,*NO.* 4X,*MACH* 6X,*K1-VALUE* 6X,*ERROR* 4X,*SIZE*,5X,	VICMAIN 00107
2 *YBAR* 5X,*ZBAR* /)	VICMAIN 00108
NDELT = 0	VICMAIN 00109
DO 550 I=1,NKST	VICMAIN 00110
IF(AMACH(I).GT.0) GO TO 525	VICMAIN 00111
NDELT = NDELT +1	VICMAIN 00112
GO TO 550	VICMAIN 00113
525 WRITE (NT6,9020) I, AMACH(I),AKVAL(I),AERR(I),ISIZE(I),YBARS(I),	VICMAIN 00114
1 VERTS(I)	VICMAIN 00115
550 CONTINUE	VICMAIN 00116
WRITE (NT6,9025) NDELT	VICMAIN 00117
C	VICMAIN 00118
C SEARCH FOR MATRICES WITH CORRECT K-VALUE, MACH, ERROR, SIZE,	VICMAIN 00119
C AND YBAR.	VICMAIN 00120
C	VICMAIN 00121
555 CONTINUE	VICMAIN 00122
IVAL = 0	VICMAIN 00123
NSIZ = 0	VICMAIN 00124
DO 600 I=1,NKST	VICMAIN 00125
IF(ABS(AMACH(I)-XMACH).GT.1.0E-05) GO TO 600	VICMAIN 00126
IF(ABS(AKVAL(I)-K1).GT.1.0E-07) GO TO 600	VICMAIN 00127
IF(AERR(I).GT.ERR) GO TO 600	VICMAIN 00128
IF(ABS(VERTS(I)-EL).GT.1.0E-04) GO TO 600	VICMAIN 00129
IF(ABS(YBAR-YBARS(I)).GT.1.0E-04) GO TO 600	VICMAIN 00130
C	VICMAIN 00131
C THERE IS A GOOD MATRIX ON TAPE. DETERMINE IF SIZE IS ADEQUATE	VICMAIN 00132
C	VICMAIN 00133
IF(IVAL.NE.0) GO TO 575	VICMAIN 00134
IF(NROWS.GT.ISIZE(I)+10.AND.ERR.GT.AERR(I)) GO TO 600	VICMAIN 00135
IVAL = I	VICMAIN 00136
NSIZ = ISIZE(I)	VICMAIN 00137
575 CONTINUE	VICMAIN 00138
IF(ISIZE(I).GE.NROWS) GO TO 600	VICMAIN 00139
C	VICMAIN 00140
C THE SIZE IS NOT LARGE ENOUGH. SEE IF THIS IS LARGER THAN ANY	VICMAIN 00141
C PREVIOUS MATRIX.	VICMAIN 00142
C	VICMAIN 00143
IF(ISIZE(I).LE.NSIZ) GO TO 600	VICMAIN 00144
IF(NROWS.GT.ISIZE(I)+10.AND.ERR.GT.AERR(I)) GO TO 600	VICMAIN 00145
IVAL = I	VICMAIN 00146
NSIZ = ISIZE(I)	VICMAIN 00147
600 CONTINUE	VICMAIN 00148
C	VICMAIN 00149
C DETERMINE IF THERE WAS A MATRIX ON TAPE THAT COULD BE ENLARGED	VICMAIN 00150
IF(IVAL.EQ.0) GO TO 25	VICMAIN 00151

C		VICMAIN 00038
C	CALCULATE 2 PLANAR AICS IF SUBDIVISION IS APPLIED.	VICMAIN 00039
	NPK = 1	VICMAIN 00040
	IF (NSUBDV.GT.1) NPK = 2	VICMAIN 00041
	NVCS = NSPATK + NPK	VICMAIN 00042
	DO 1000 NV = 1,NVCS	VICMAIN 00043
C		VICMAIN 00044
C	SET K1 VALUE IF SUBDIVIDED AIC	VICMAIN 00045
C		VICMAIN 00046
	IF (NPK.EQ.2.AND.NV.EQ.NVCS) K1= K1/FLOAT(NSUBDV)	VICMAIN 00047
C		VICMAIN 00048
	LENZ = LKERNL	VICMAIN 00049
	IF (NV.EQ.NVCS.AND.NPK.EQ.2) LENZ = LSKERN	VICMAIN 00050
	DO 100 I=1,LENZ	VICMAIN 00051
	C(I) = (0.,0.)	VICMAIN 00052
	W(I) = (0.,0.)	VICMAIN 00053
	V(I) = (0.,0.)	VICMAIN 00054
	100 CONTINUE	VICMAIN 00055
C		VICMAIN 00056
	110 CONTINUE	VICMAIN 00057
	IF (NV.GT.NVCS-NPK) GO TO 10	VICMAIN 00058
C		VICMAIN 00059
C	READ MUAIC ARRAY FROM IGEOSC. FIRST FILE MUST BE SKIPPED	VICMAIN 00060
C	PRIOR TO FIRST READ.	VICMAIN 00061
C		VICMAIN 00062
	CALL RDINIT	VICMAIN 00063
	IF (NV.NE.1) GO TO 200	VICMAIN 00064
	REWIND IGEOSC	VICMAIN 00065
	NFS = 1	VICMAIN 00066
	200 CONTINUE	VICMAIN 00067
	ITYPE = SHMIXED	VICMAIN 00068
	MXARRY = GHMUAIC	VICMAIN 00069
	CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	VICMAIN 00070
	1 LRS,MUAIC,H,NROWS,FARM,IRR)	VICMAIN 00071
	IF (IRR.NE.0) GO TO 6010	VICMAIN 00072
C		VICMAIN 00073
	YBAR = PARM(4)	VICMAIN 00074
	EL = PARM(5)	VICMAIN 00075
	NN = 1	VICMAIN 00076
C		VICMAIN 00077
C	DETERMINE IF SPATIAL AICS ARE ON TAPE AND GET THEM	VICMAIN 00078
C	IF POSSIBLE.	VICMAIN 00079
C		VICMAIN 00080
	NKST = 0	VICMAIN 00081
	IF (.NOT.OSPAIC) GO TO 25	VICMAIN 00082
C		VICMAIN 00083
C	AICS ARE ON TAPE. GET TABLE OF CONTENTS.	VICMAIN 00084
	REWIND NSPAIC	VICMAIN 00085
	CALL RDINIT	VICMAIN 00086
	NFS = 1	VICMAIN 00087
	ITYPE = SHMIXED	VICMAIN 00088
	MXARRY = GHTAB	VICMAIN 00089
	CALL READMX(NSPAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,500,NID,ID,ITYPE,	VICMAIN 00090
	1 LRS, TAB,NKST,N,FARM,IRR)	VICMAIN 00091
	IF (IRR.NE.0) GO TO 6060	VICMAIN 00092
C		VICMAIN 00093
	CALL RDINIT	VICMAIN 00094

C		VICMAIN 00038
C	CALCULATE 2 PLANAR AICS IF SUBDIVISION IS APPLIED.	VICMAIN 00039
	NPK = 1	VICMAIN 00040
	IF(NSUBDV.GT.1) NPK = 2	VICMAIN 00041
	NVCS = NSPATK + NPK	VICMAIN 00042
	DO 1000 NV = 1,NVCS	VICMAIN 00043
C		VICMAIN 00044
C	SET K1 VALUE IF SUBDIVIDED AIC	VICMAIN 00045
C		VICMAIN 00046
	IF(NPK.EQ.2.AND.NV.EQ.NVCS) K1= K1/FLOAT(NSUBDV)	VICMAIN 00047
C		VICMAIN 00048
	LENZ = LKERNL	VICMAIN 00049
	IF(NV.EQ.NVCS.AND.NPK.EQ.2) LENZ = LSKERN	VICMAIN 00050
	DO 100 I=1,LENZ	VICMAIN 00051
	C(I) = (0.,0.)	VICMAIN 00052
	W(I) = (0.,0.)	VICMAIN 00053
	V(I) = (0.,0.)	VICMAIN 00054
	100 CONTINUE	VICMAIN 00055
C		VICMAIN 00056
	110 CONTINUE	VICMAIN 00057
	IF(NV.GT.NVCS-NPK) GO TO 10	VICMAIN 00058
C		VICMAIN 00059
C	READ MUAIC ARRAY FROM IGEOSC. FIRST FILE MUST BE SKIPPED	VICMAIN 00060
C	PRIOR TO FIRST READ.	VICMAIN 00061
C		VICMAIN 00062
	CALL RDINIT	VICMAIN 00063
	IF(NV.NE.1) GO TO 200	VICMAIN 00064
	REWIND IGEOSC	VICMAIN 00065
	NFS = 1	VICMAIN 00066
	200 CONTINUE	VICMAIN 00067
	ITYPE = SHMIXED	VICMAIN 00068
	MXARRY = 6HMUAIC	VICMAIN 00069
	CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	VICMAIN 00070
	1 LRS,MUAIC,M,NROWS,FARM,IRR)	VICMAIN 00071
	IF(IRR.NE.0) GO TO 6010	VICMAIN 00072
C		VICMAIN 00073
	YBAR = FARM(4)	VICMAIN 00074
	EL = FARM(5)	VICMAIN 00075
	NN = 1	VICMAIN 00076
C		VICMAIN 00077
C	DETERMINE IF SPATIAL AICS ARE ON TAPE AND GET THEM	VICMAIN 00078
C	IF POSSIBLE.	VICMAIN 00079
C		VICMAIN 00080
	NKST = 0	VICMAIN 00081
	IF(.NOT.OSPAIC) GO TO 25	VICMAIN 00082
C		VICMAIN 00083
C	AICS ARE ON TAPE. GET TABLE OF CONTENTS.	VICMAIN 00084
	REWIND NSPAIC	VICMAIN 00085
	CALL RDINIT	VICMAIN 00086
	NFS = 1	VICMAIN 00087
	ITYPE = SHMIXED	VICMAIN 00088
	MXARRY = 6HTAB	VICMAIN 00089
	CALL READMX(NSPAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,500,NID,ID,ITYPE,	VICMAIN 00090
	1 LRS, TAB,NKST,N,FARM,IRR)	VICMAIN 00091
	IF(IRR.NE.0) GO TO 6060	VICMAIN 00092
C		VICMAIN 00093
	CALL RDINIT	VICMAIN 00094

C		VICMAIN 00152
C	THERE IS A MATRIX THAT CAN BE ENLARGED.	VICMAIN 00153
	AMACH(IVAL) = -AMACH(IVAL)	VICMAIN 00154
	WRITE (MT6,9030) IVAL, AERR(IVAL),ISIZE(IVAL),NROWS	VICMAIN 00155
C		VICMAIN 00156
C	SPACE TO CORRECT ARRAY ON TAPE	VICMAIN 00157
	CALL RDINIT	VICMAIN 00158
	NMS = (IVAL-1)*4	VICMAIN 00159
	ITYPE = 5H MIXED	VICMAIN 00160
	MXARRY = 6H MUTWO	VICMAIN 00161
	CALL READMX(NSPAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR, 2 ,NID,ID,ITYPE,	VICMAIN 00162
1	LRS,MUTWO,M,N,FARM,IRR)	VICMAIN 00163
	IF(IRR.NE.0) GO TO 6060	VICMAIN 00164
C		VICMAIN 00165
	CALL RDINIT	VICMAIN 00166
	ITYPE = 5H MIXED	VICMAIN 00167
	MXARRY = 6H C	VICMAIN 00168
	CALL READMX(NSPAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	VICMAIN 00169
1	LRS, C, M,N,FARM,IRR)	VICMAIN 00170
	IF(IRR.NE.0) GO TO 6060	VICMAIN 00171
C		VICMAIN 00172
	CALL RDINIT	VICMAIN 00173
	ITYPE = 5H MIXED	VICMAIN 00174
	MXARRY = 6H W	VICMAIN 00175
	CALL READMX(NSPAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	VICMAIN 00176
1	LRS, W, M,N,FARM,IRR)	VICMAIN 00177
	IF(IRR.NE.0) GO TO 6060	VICMAIN 00178
C		VICMAIN 00179
	CALL RDINIT	VICMAIN 00180
	ITYPE = 5H MIXED	VICMAIN 00181
	MXARRY = 6H V	VICMAIN 00182
	CALL READMX(NSPAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	VICMAIN 00183
1	LRS, V, M,N,FARM,IRR)	VICMAIN 00184
	IF(IRR.NE.0) GO TO 6060	VICMAIN 00185
C		VICMAIN 00186
	GO TO 25	VICMAIN 00187
	800 CONTINUE	VICMAIN 00188
C		VICMAIN 00189
C	THERE IS A GOOD MATRIX ON TAPE. READ THE TAPE, PRINT MESSAGE,	VICMAIN 00190
C	MAIL RESUME ON THIS ONE ANYWAY.	VICMAIN 00191
C		VICMAIN 00192
C	SET ISIZ EQUAL TO NROWS SO THAT MATRIX WILL NOT BE WRITTEN	VICMAIN 00193
C	ON TAPE	VICMAIN 00194
C		VICMAIN 00195
	NSIZ = NROWS	VICMAIN 00196
	CALL RDINIT	VICMAIN 00197
	NMS = (I-1)*4	VICMAIN 00198
	ITYPE = 5H MIXED	VICMAIN 00199
	MXARRY = 6H MUTWO	VICMAIN 00200
	CALL READMX(NSPAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2, NID,ID,ITYPE,	VICMAIN 00201
1	LRS,MUTWO,M,N,FARM,IRR)	VICMAIN 00202
	IF(' ',NE.0) GO TO 6060	VICMAIN 00203
C		VICMAIN 00204
	CALL RDINIT	VICMAIN 00205
	ITYPE = 5H MIXED	VICMAIN 00206
	MXARRY = 6H C	VICMAIN 00207
	CALL READMX(NSPAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	VICMAIN 00208

1	LRS,C,M,N, PARM,IRR)	VICMAIN 00209
	IF(IRR.NE.0) GO TO 6060	VICMAIN 00210
C		VICMAIN 00211
	CALL RDINIT	VICMAIN 00212
	ITYPE = SHMIXED	VICMAIN 00213
	MXARRY = GH W	VICMAIN 00214
	CALL READMX(NSPAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	VICMAIN 00215
1	LRS,W,M,N, PARM,IRR)	VICMAIN 00216
	IF(IRR.NE.0) GO TO 6060	VICMAIN 00217
C		VICMAIN 00218
	CALL RDINIT	VICMAIN 00219
	ITYPE = SHMIXED	VICMAIN 00220
	MXARRY = GH V	VICMAIN 00221
	CALL READMX(NSPAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	VICMAIN 00222
1	LRS,V,M,N, PARM, IRR)	VICMAIN 00223
	IF(IRR.NE.0) GO TO 6060	VICMAIN 00224
	WRITE (NT6,6005) I, AERR(I)	VICMAIN 00225
	GO TO 25	VICMAIN 00226
10	CONTINUE	VICMAIN 00227
	YBAR = 0.0	VICMAIN 00228
	EL = 0.0	VICMAIN 00229
C		VICMAIN 00230
C	DETERMINE THE SIZE AND LOCATIONS OF THE PLANAR AIC ARRAYS.	VICMAIN 00231
C	THE UNSUBDIVIDED WILL BE CALCULATED FIRST AND STORED IN	VICMAIN 00232
C	THE PROPER PLACE IN BLANK COMMON.	VICMAIN 00233
C	THE SUBDIVIDED WILL BE CALCULATED SECOND, OVERLAYING SOME OF	VICMAIN 00234
C	THE UNSUBDIVIDED NUMBERS.	VICMAIN 00235
C		VICMAIN 00236
C	LSKERN = SIZE OF UNSUBDIVIDED OR SUBDIVIDED ARRAY ALONE.	VICMAIN 00237
C	ISUB = NUMBER OF ROWS ON UNSUBDIVIDED AIC THAT SUBDIVIDED	VICMAIN 00238
C	AIC ARRAY WILL OVERLAY	VICMAIN 00239
C	LYOBKN = NUMBER OF BOXES OF UNSUBDIVIDED AIC THAT WILL BE	VICMAIN 00240
C	OVERLAID	VICMAIN 00241
C	IPKERN = SUBSCRIPT OF WHERE FIRST BOX OF UNSUBDIVIDED BOX	VICMAIN 00242
C	WOULD BE IF IT WERE NOT OVERLAID. THIS ALLOWS	VICMAIN 00243
C	PROGRAM TO REFERENCE UNSUBDIVIDED ARRAY WITH PROPER	VICMAIN 00244
C	SUBSCRIPT.	VICMAIN 00245
C	MAXL = LENGTH OF COMPUTED AIC ARRAY	VICMAIN 00246
C		VICMAIN 00247
	XNA = NPLKRN	VICMAIN 00248
	LUKERN = (XNA/2.) * (XNA + 1.) + .001	VICMAIN 00249
	MXSKRN = NPLKRN	VICMAIN 00250
	IST = 0	VICMAIN 00251
	IF(NSUBDV.EQ.1) GO TO 340	VICMAIN 00252
C		VICMAIN 00253
C	IF THE EFFECTIVE AREA WAS INPUT ON CARD USE THAT	VICMAIN 00254
C	FIND IF THE PLANFORM LIMITS THE SIZE OF THE EFFECTIVE AREA.	VICMAIN 00255
C		VICMAIN 00256
	IF(NRWEA.GT.20) NRWEA = 20	VICMAIN 00257
	IF(NRWEA.NE.0) MXSKRN = NSUBDV * NRWEA	VICMAIN 00258
C		VICMAIN 00259
C		VICMAIN 00260
	NBOXES = MYBBSW	VICMAIN 00261
	NBROW = 1	VICMAIN 00262
	NBOXRW = (MYBBSW-1)*2	VICMAIN 00263
300	CONTINUE	VICMAIN 00264
	IF(NBOXES .GT. LSDW) GO TO 323	VICMAIN 00265

NBOXES = NBOXES + NBOXRW	VICMAIN 00266
NBOXRW = NBOXRW - 2	VICMAIN 00267
IF(NBOXRW.LE.0) GO TO 330	VICMAIN 00268
NSROW = NSROW + 2	VICMAIN 00269
GO TO 300	VICMAIN 00270
325 CONTINUE	VICMAIN 00271
NSROW = NSROW - 1	VICMAIN 00272
IF (NBOXES-NBOXRW/2 .GT. LSDW) NSROW = NSROW - 1	VICMAIN 00273
GO TO 335	VICMAIN 00274
330 CONTINUE	VICMAIN 00275
NSROW = MXSKRN	VICMAIN 00276
335 CONTINUE	VICMAIN 00277
XNA = MXSKRN	VICMAIN 00278
LSKERN = XNA*(XNA/2.) + (XNA/2.) + 0.001	VICMAIN 00279
ISUB = MXSKRN/NSUBDV	VICMAIN 00280
IF (NSROW .LT. MXSKRN) ISUB = NSROW/NSUBDV	VICMAIN 00281
SUB = ISUB	VICMAIN 00282
LTCBKN = SUB*(SUB/2.) + (SUB/2.) + 0.001	VICMAIN 00283
IST = LSKERN - LTCBKN	VICMAIN 00284
340 CONTINUE	VICMAIN 00285
IPKERN = IST + 1	VICMAIN 00286
MAXL = IST + LSKERN	VICMAIN 00287
IF(MAXL.LE.LKERNL) GO TO 21	VICMAIN 00288
WRITE (NT6,9305) MAXL,LKERNL	VICMAIN 00289
9305 FORMAY(59H)*** ERROR - THE SIZE OF THE AIC ARRAY FOR THIS PLANFORM	VICMAIN 00290
1 IS,15,29H, THE MAXIMUM SIZE ALLOWED IS,15, 5H. ***)	VICMAIN 00291
CALL FLUSH(1)	VICMAIN 00292
C	VICMAIN 00293
21 CONTINUE	VICMAIN 00294
C	VICMAIN 00295
IF(NFK.EQ.2.AND.MV.EQ.NVCS-1) GO TO 22	VICMAIN 00296
NN = 1	VICMAIN 00297
NROWS = MXSKRN	VICMAIN 00298
GO TO 23	VICMAIN 00299
22 CONTINUE	VICMAIN 00300
NN = IPKERN	VICMAIN 00301
NROWS = NFKERN	VICMAIN 00302
23 CONTINUE	VICMAIN 00303
DO 24 I=1,NROWS	VICMAIN 00304
MUAIC(1,I) = I	VICMAIN 00305
MUAIC(2,I) = I + I - 1	VICMAIN 00306
24 CONTINUE	VICMAIN 00307
C	VICMAIN 00308
C	VICMAIN 00309
C DETERMINE IF KERNELS EXIST ON TAPE AND GET THEM FROM TAPE	VICMAIN 00310
C IF POSSIBLE.	VICMAIN 00311
C	VICMAIN 00312
NKOT = 0	VICMAIN 00313
IF(.NOT.OPLAIC) GO TO 25	VICMAIN 00314
C	VICMAIN 00315
C KERNELS ARE ON TAPE. GET TABLE OF CONTENTS	VICMAIN 00316
REWIND NPLAIC	VICMAIN 00317
CALL RDNIT	VICMAIN 00318
NFS = 1	VICMAIN 00319
ITYPE = 5H MIXED	VICMAIN 00320
MXARRY = 6HTAB	VICMAIN 00321
CALL READMX(NPLAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,500,NID,ID,ITYPE,	VICMAIN 00322

2	31X,1HC,34X,1HW,34X,1HV,3X,2HNU,3X,2HMU,7X,*VELOCITY POTENTIAL	VICMAIN 00437
3	COEFFICIENT* 10X,*UPWASH COEFFICIENT*,15X,*SIDEWASH COEFFICIENT* /	VICMAIN 00438
4	2(3X,2H-),7X, 32(1H-),2X,32(1H-),2X,32(1H-) ///	VICMAIN 00439
C	IF(.NOT.PRNT) GO TO 52	VICMAIN 00440
	K=0	VICMAIN 00441
	KN = 0	VICMAIN 00442
	IF(NPK.EQ.2.AND.NV.EQ.NVCS-1) KN = IPKERN -1	VICMAIN 00443
C		VICMAIN 00444
	DO 50 I=1,NROWS	VICMAIN 00445
	M= I-1	VICMAIN 00446
	MM = M/2	VICMAIN 00447
	JS = 2*(I-1) +1	VICMAIN 00448
	IF(YBAR.NE.0.0) JS = JS+1	VICMAIN 00449
	IF(EL.EQ.0.0) JS = I	VICMAIN 00450
	DO 50 J=1,JS	VICMAIN 00451
	K=K+1	VICMAIN 00452
	KN = KN + 1	VICMAIN 00453
	N = I - J	VICMAIN 00454
	IF(EL.EQ.0.0) N = 1- J	VICMAIN 00455
	IF(YBAR.LT.0.0) N = -N	VICMAIN 00456
	WRITE (NT6,9210) M,N,C(KN),W(K),V(K)	VICMAIN 00457
9210	FORMAT(2I5,5X,6E17.8)	VICMAIN 00458
50	CONTINUE	VICMAIN 00459
	GO TO 53	VICMAIN 00460
C		VICMAIN 00461
52	CONTINUE	VICMAIN 00462
	RN = NROWS	VICMAIN 00463
	K = RN*(RN/2.) + (RN/2.) + 1.0E-05	VICMAIN 00464
	IF(EL.EQ.0.) GO TO 53	VICMAIN 00465
	K = K+K	VICMAIN 00466
	IF(YBAR.EQ.0.) K = K - NROWS	VICMAIN 00467
53	CONTINUE	VICMAIN 00468
C		VICMAIN 00469
C		VICMAIN 00470
	IF (NBPATK .EQ. 0 .OR. NV .GT. NVCS-NPK) GO TO 55	VICMAIN 00471
	IF(NV.EQ.1) REWIND IAICSC	VICMAIN 00472
C		VICMAIN 00473
C	WRITE THE SPATIAL AICS ON A SCRATCH FILE	VICMAIN 00474
C		VICMAIN 00475
	CALL RDINIT	VICMAIN 00476
	PARM(1) = K1	VICMAIN 00477
	PARM(2) = XMACH	VICMAIN 00478
	PARM(4) = YBAR	VICMAIN 00479
	PARM(5) = EL	VICMAIN 00480
	ITYPE = SHMIXED	VICMAIN 00481
	MXARRY = GHMUAIC	VICMAIN 00482
	CALL WRTEMX(IAICSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,	VICMAIN 00483
1	MUAIC,ITYPE,2,NROWS,PARM,IRR)	VICMAIN 00484
	IF(IRR.NE.0) GO TO 6170	VICMAIN 00485
C		VICMAIN 00486
	MXARRY = GH C	VICMAIN 00487
	CALL WRTEMX(IAICSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,	VICMAIN 00488
1	C, ITYPE, 2,K, PARM, IRR)	VICMAIN 00489
	IF(IRR.NE.0) GO TO 6170	VICMAIN 00490
C		VICMAIN 00491
	MXARRY = 3H W	VICMAIN 00492
		VICMAIN 00493

AMACH(IVAL) = -AMACH(IVAL)	VICMAIN 00380
WRITE (NT6,9030) IVAL, AERR(IVAL),ISIZE(IVAL),NPLKRN	VICMAIN 00381
9030 FORMAT(46H)THE PROGRAM IS GOING TO ENLARGE AIC ARRAY NO. I3,	VICMAIN 00382
1 29H, GENERATED AT AN ACCURACY OF F8.5 /19H IT IS NECESSARY TO	VICMAIN 00383
2 29H ENLARGE THE SAVED ARRAY FROM I3, 3H TO, I3)	VICMAIN 00384
C	VICMAIN 00385
C SPACE TO CORRECT ARRAY ON TAPE	VICMAIN 00386
REWIND NPLAIC	VICMAIN 00387
CALL RDINIT	VICMAIN 00388
NMS = IVAL-1	VICMAIN 00389
ITYPE = 5HMIXED	VICMAIN 00390
MXARRY = 6H C	VICMAIN 00391
CALL READMX(NPLAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	VICMAIN 00392
1 LRS,C(NN), M, N, PARM, IRR)	VICMAIN 00393
IF(IRR.NE.0) GO TO 6050	VICMAIN 00394
C	VICMAIN 00395
C SET MUAIC ARRAY FOR EXPANSION AIC CALCULATION	VICMAIN 00396
DO 1700 I=1,NSIZE	VICMAIN 00397
MUAIC(1,I) = 0	VICMAIN 00398
MUAIC(2,I) = 0	VICMAIN 00399
1700 CONTINUE	VICMAIN 00400
GO TO 25	VICMAIN 00401
1800 CONTINUE	VICMAIN 00402
C	VICMAIN 00403
C THERE IS A GOOD MATRIX ON TAPE. READ THE TAPE, PRINT MESSAGE,	VICMAIN 00404
C DO NOT MAIL RESUME.	VICMAIN 00405
C SET NSIZE EQUAL TO NPLKRN SO THAT MATRIX WILL NOT BE WRITTEN	VICMAIN 00406
C ON TAPE.	VICMAIN 00407
C	VICMAIN 00408
NSIZE = NPLKRN	VICMAIN 00409
CALL RDINIT	VICMAIN 00410
REWIND NPLAIC	VICMAIN 00411
NMS = I-1	VICMAIN 00412
ITYPE = 5HMIXED	VICMAIN 00413
MXARRY = 6H C	VICMAIN 00414
CALL READMX(NPLAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2, NID,ID,ITYPE,	VICMAIN 00415
1 LRS,C(NN), M, N, PARM, IRR)	VICMAIN 00416
IF(IRR.NE.0) GO TO 6050	VICMAIN 00417
C	VICMAIN 00418
WRITE (NT6,6005) I, AERR(I)	VICMAIN 00419
GO TO 35	VICMAIN 00420
25 CONTINUE	VICMAIN 00421
C	VICMAIN 00422
CALL KERNEL (XMACH,K1,ERR,C(NN),W,V)	VICMAIN 00423
C	VICMAIN 00424
35 CONTINUE	VICMAIN 00425
PRNT = .FALSE.	VICMAIN 00426
IF (NV.GT.NVCS-NPK) GO TO 40	VICMAIN 00427
IF(PRSAIC) PRNT = .TRUE.	VICMAIN 00428
GO TO 45	VICMAIN 00429
40 CONTINUE	VICMAIN 00430
IF(PRPAIC) PRNT = .TRUE.	VICMAIN 00431
45 CONTINUE	VICMAIN 00432
IF (.PRNT) WRITE (NT6,9005) TITLE,XMACH,K1,ERR,EL,YBAR	VICMAIN 00433
9005 FOR-AT(1H1, 8A10, // 40X, *AIC CALCULATIONS*, ///	VICMAIN 00434
X 17X, *MACH =*, F10.5, 5X, *K1 =*,	VICMAIN 00435
1 F10.7,5X,*ERR =*,E12.5,5X,*EL =*, F6.2,5X,*YBAR =*,F6.2,//	VICMAIN 00436

2	31X,1HC,34X,1HW,34X,1HV,3X,2HNU,3X,2HMU,7X,*VELOCITY POTENTIAL	VICMAIN 00437
	3COEFFICIENT* 10X,*UPWASH COEFFICIENT*,15X,*SIDEWASH COEFFICIENT* /	VICMAIN 00438
4	2(3X,2H-),7X, 32(1H-),2X,32(1H-),2X,32(1H-) //)	VICMAIN 00439
C		VICMAIN 00440
	IF(.NOT.PRINT) GO TO 52	VICMAIN 00441
	K=0	VICMAIN 00442
	KN = 0	VICMAIN 00443
	IF(NPK.EQ.2.AND.NV.EQ.NVCS-1) KN = IPKERN -1	VICMAIN 00444
C		VICMAIN 00445
	DO 50 I=1,NROWS	VICMAIN 00446
	MM = I-1	VICMAIN 00447
	MM = M/2	VICMAIN 00448
	JS = 2*(I-1) +1	VICMAIN 00449
	IF(YBAR.NE.0.0) JS = JS+1	VICMAIN 00450
	IF(EL.EQ.0.0) JS = I	VICMAIN 00451
	DO 50 J=1,JS	VICMAIN 00452
	K=K+1	VICMAIN 00453
	KN = KN + 1	VICMAIN 00454
	N = I - J	VICMAIN 00455
	IF(EL.EQ.0.0) N = 1- J	VICMAIN 00456
	IF(YBAR.LT.0.0) N = -N	VICMAIN 00457
	WRITE (NT6,9210) M,N,C(KN),W(K),V(K)	VICMAIN 00458
9210	FORMAT(2I5,5X,6E17.8)	VICMAIN 00459
	50 CONTINUE	VICMAIN 00460
	GO TO 53	VICMAIN 00461
C		VICMAIN 00462
	52 CONTINUE	VICMAIN 00463
	RN = NROWS	VICMAIN 00464
	K = RN* (RN/2.) + (RN/2.) + 1.0E-05	VICMAIN 00465
	IF(EL.EQ.0.) GO TO 53	VICMAIN 00466
	K = K+K	VICMAIN 00467
	IF(YBAR.EQ.0.) K = K - NROWS	VICMAIN 00468
	53 CONTINUE	VICMAIN 00469
C		VICMAIN 00470
C		VICMAIN 00471
	IF (NSPATK .EQ. 0 .OR. NV .GT. NVCS-NPK) GO TO 55	VICMAIN 00472
	IF(NV.EQ.1) REWIND IAICSC	VICMAIN 00473
C		VICMAIN 00474
C	WRITE THE SPATIAL AICS ON A SCRATCH FILE	VICMAIN 00475
C		VICMAIN 00476
	CALL RDINIT	VICMAIN 00477
	PARM(1) = K1	VICMAIN 00478
	PARM(2) = XMACH	VICMAIN 00479
	PARM(4) = YBAR	VICMAIN 00480
	PARM(5) = EL	VICMAIN 00481
	ITYPE = SHMIXED	VICMAIN 00482
	MXARRY = SHMUAIC	VICMAIN 00483
	CALL WRITEMX(IAICSC,MXWRIT,RANDCU,NFS,NMS,LS,NMR,LWS,2,ID,	VICMAIN 00484
	1 MUAIC,ITYPE,2,NROWS,PARM,IRR)	VICMAIN 00485
	IF(IRR.NE.0) GO TO 6170	VICMAIN 00486
C		VICMAIN 00487
	MXARRY = SH C	VICMAIN 00488
	CALL WRITEMX(IAICSC,MXWRIT,RANDCU,NFS,NMS,LS,NMR,LWS,2,ID,	VICMAIN 00489
	1 C, ITYPE, 2,K, PARM, IRR)	VICMAIN 00490
	IF(IRR.NE.0) GO TO 6170	VICMAIN 00491
C		VICMAIN 00492
	MXARRY = SH W	VICMAIN 00493

CALL WRTEMX(IAICSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,	VICMAIN 00494
1 W, ITYPE, 2,K, PARM, IRR)	VICMAIN 00495
IF(IRR.NE.0) GO TO 6170	VICMAIN 00496
C	VICMAIN 00497
MXARRY = 3H V	VICMAIN 00498
CALL WRTEMX(IAICSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,	VICMAIN 00499
1 V, ITYPE, 2,K, PARM, IRR)	VICMAIN 00500
IF(IRR.NE.0) GO TO 6170	VICMAIN 00501
C	VICMAIN 00502
IF(MV.NE.NVCS-NFK) GO TO 55	VICMAIN 00503
END FILE IAICSC	VICMAIN 00504
REWIND IAICSC	VICMAIN 00505
55 CONTINUE	VICMAIN 00506
IF(NPLAIC.EQ.0.AND.NSFAIC.EQ.0) GO TO 1000	VICMAIN 00507
C	VICMAIN 00508
WRITE THE KERNEL ON TAPE	VICMAIN 00509
IF(MV.LE.NVCS-NFK) GO TO 900	VICMAIN 00510
C	VICMAIN 00511
WRITE ON THE PLANK KERNEL TAPE	VICMAIN 00512
IF(NPLAIC.EQ.0) GO TO 1000	VICMAIN 00513
C	VICMAIN 00514
DETERMINE IF A PREVIOUS MATRIX WAS ON TAPE.	VICMAIN 00515
IF NKOT = 0 IT IS A NEW TAPE AND THERE ARE NO OLD ONES	VICMAIN 00516
IF NSIZE IS LESS THAN NROWS A MATRIX WAS EXPANDED OR THERE	VICMAIN 00517
WAS NONE WITH CORRESPONDING PARAMETERS	VICMAIN 00518
C	VICMAIN 00519
IF(NKOT.EQ.0) GO TO 60	VICMAIN 00520
IF(NSIZE.GE.NROWS) GO TO 1000	VICMAIN 00521
C	VICMAIN 00522
60 CONTINUE	VICMAIN 00523
NKOT = NKOT + 1	VICMAIN 00524
AMACH(NKOT) = XMACH	VICMAIN 00525
AKVAL(NKOT) = K1	VICMAIN 00526
AERR(NKOT) = ERR	VICMAIN 00527
ISIZE(NKOT) = NROWS	VICMAIN 00528
REWIND NPLAIC	VICMAIN 00529
CALL RDINIT	VICMAIN 00530
NMS = NKOT - 1	VICMAIN 00531
ITYPE = SHMIXED	VICMAIN 00532
MXARRY = 6H C	VICMAIN 00533
CALL WRTEMX(NPLAIC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,	VICMAIN 00534
1 C(NN), ITYPE, 2, K, PARM, IRR)	VICMAIN 00535
IF(IRR.NE.0) GO TO 6150	VICMAIN 00536
C	VICMAIN 00537
END FILE NPLAIC	VICMAIN 00538
CALL RDINIT	VICMAIN 00539
ITYPE = SHMIXED	VICMAIN 00540
MXARRY = 6HTAB	VICMAIN 00541
CALL WRTEMX(NPLAIC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,500,ID,	VICMAIN 00542
1 TAB, ITYPE, NKOT, 3, PARM, IRR)	VICMAIN 00543
IF(IRR.NE.0) GO TO 6150	VICMAIN 00544
C	VICMAIN 00545
MXARRY = 6HISIZE	VICMAIN 00546
CALL WRTEMX(NPLAIC,MXWRIT,RANDOU,NFS,NMS,LS,NMR, LWS,1,ID,	VICMAIN 00547
1 ISIZE, ITYPE, 1, NKOT, PARM, IRR)	VICMAIN 00548
IF(IRR.NE.0) GO TO 6150	VICMAIN 00549
END FILE NPLAIC	VICMAIN 00550

```

END FILE NPLAIC
OPLAIC = .TRUE.
GO TO 1000
900 CONTINUE
C
C      THIS AREA WILL WRITE SPATIAL KERNELS ON TAPE
C      THIS TAPE WILL BE USED IN DOWNWASH AND VELOCITY POTENTIAL
C      CALCULATIONS.
C      IN THE DEVELOPMENT STAGE IT WILL BE ASCERTAINED IF A TAPE
C      SHOULD BE SAVED FOR SUBSEQUENT RUNS.
C
      IF(NSPAIC.EQ.0) GO TO 1000
      IF(NKST.NE.0) GO TO 910
C
      THERE WAS NO OLD KERNEL TAPE THEREFORE SKIP TO NV-NRK AND
      ADD TO TABLE OF CONTENTS.
C
      NKST = NV
      GO TO 920
910 CONTINUE
C
      IF(NSIZ IS LESS THAN NROWS A MATRIX WAS EXPANDED OR THERE
      WAS NONE WITH CORRESPONDING PARAMETERS.
C
      IF(NSIZ.GE.NROWS) GO TO 1000
C
      THERE WAS AN OLD KERNEL TAPE THEREFORE SKIP TO NKST+1 AND
      ADD TO TABLE OF CONTENTS
C
      NKST = NKST + 1
920 CONTINUE
      AMACH(NKST) = XMACH
      AKVAL(NKST) = K1
      AERR(NKST) = ERR
      YBARS(NKST) = YBAR
      VERTS(NKST) = EL
      ISIZE(NKST) = NROWS
      REMIND NSPAIC
      CALL RDINIT
      ITYPE = SHMIXED
      NMS = (NKST - 1)*4
      MXARRY = SHMUAIC
      CALL WRTEMX(NSPAIC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
1          MUAIC,ITYPE,2,NROWS,PARM,IRR)
      IF(IRR.NE.0) GO TO 6160
C
      NMS = 0
      ITYPE = SHMIXED
      MXARRY = 6H C
      CALL WRTEMX(NSPAIC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
1          C, ITYPE,2,K,PARM,IRR)
      IF(IRR.NE.0) GO TO 6160
C
      MXARRY = 6H W
      CALL WRTEMX(NSPAIC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
1          W, ITYPE,2,K,PARM,IRR)
      IF(IRR.NE.0) GO TO 6160

```

```

VICMAIN 00551
VICMAIN 00552
VICMAIN 00553
VICMAIN 00554
VICMAIN 00555
VICMAIN 00556
VICMAIN 00557
VICMAIN 00558
VICMAIN 00559
VICMAIN 00560
VICMAIN 00561
VICMAIN 00562
VICMAIN 00563
VICMAIN 00564
VICMAIN 00565
VICMAIN 00566
VICMAIN 00567
VICMAIN 00568
VICMAIN 00569
VICMAIN 00570
VICMAIN 00571
VICMAIN 00572
VICMAIN 00573
VICMAIN 00574
VICMAIN 00575
VICMAIN 00576
VICMAIN 00577
VICMAIN 00578
VICMAIN 00579
VICMAIN 00580
VICMAIN 00581
VICMAIN 00582
VICMAIN 00583
VICMAIN 00584
VICMAIN 00585
VICMAIN 00586
VICMAIN 00587
VICMAIN 00588
VICMAIN 00589
VICMAIN 00590
VICMAIN 00591
VICMAIN 00592
VICMAIN 00593
VICMAIN 00594
VICMAIN 00595
VICMAIN 00596
VICMAIN 00597
VICMAIN 00598
VICMAIN 00599
VICMAIN 00600
VICMAIN 00601
VICMAIN 00602
VICMAIN 00603
VICMAIN 00604
VICMAIN 00605
VICMAIN 00606
VICMAIN 00607

```

C	MXARRY = GH V	VICMAIN 00608
	CALL WRTEMX(NSPAIC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,	VICMAIN 00609
	1 V, ITYPE,2,K,PARM,IRR)	VICMAIN 00610
	IF(IRR.NE.0) GO TO 6160	VICMAIN 00611
C	END FILE NSPAIC	VICMAIN 00612
C	WRITE TABLE OF CONTENTS ARRAYS	VICMAIN 00613
	ITYPE = 5H MIXED	VICMAIN 00614
	MXARRY = GH TAB	VICMAIN 00615
	CALL WRTEMX(NSPAIC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,500,ID,	VICMAIN 00616
	1 TAB,ITYPE,NKST,3,PARM,IRR)	VICMAIN 00617
	IF(IRR.NE.0) GO TO 6160	VICMAIN 00618
C	MXARRY = GH ATAB	VICMAIN 00619
	CALL WRTEMX(NSPAIC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,500,ID,	VICMAIN 00620
	1 ATAB,ITYPE,NKST,3,PARM,IRR)	VICMAIN 00621
	IF(IRR.NE.0) GO TO 6160	VICMAIN 00622
C	END FILE NSPAIC	VICMAIN 00623
	END FILE NSPAIC	VICMAIN 00624
	IF(MV.EQ.MVCS-NPK) OSFAIC = .TRUE.	VICMAIN 00625
1000	CONTINUE	VICMAIN 00626
	RETURN	VICMAIN 00627
C	6010 CONTINUE	VICMAIN 00628
	WRITE (NT6,9010) IGEO SC,IRR	VICMAIN 00629
	WRITE (NT6,9011) MXARRY	VICMAIN 00630
	GO TO 6100	VICMAIN 00631
C	6050 CONTINUE	VICMAIN 00632
	WRITE (NT6,9050) NPLAIC,IRR	VICMAIN 00633
	WRITE (NT6,9011) MXARRY	VICMAIN 00634
	GO TO 6100	VICMAIN 00635
C	6060 CONTINUE	VICMAIN 00636
	WRITE (NT6,9060) NSPAIC,IRR	VICMAIN 00637
	WRITE (NT6,9011) MXARRY	VICMAIN 00638
	GO TO 6100	VICMAIN 00639
C	6150 CONTINUE	VICMAIN 00640
	WRITE (NT6,9150) NPLAIC,IRR	VICMAIN 00641
	WRITE (NT6,9151) MXARRY	VICMAIN 00642
	GO TO 6100	VICMAIN 00643
C	6160 CONTINUE	VICMAIN 00644
	WRITE (NT6,9160) NSPAIC,IRR	VICMAIN 00645
	WRITE (NT6,9151) MXARRY	VICMAIN 00646
	GO TO 6100	VICMAIN 00647
C	6170 CONTINUE	VICMAIN 00648
	WRITE (NT6,9170) IAIC SC,IRR	VICMAIN 00649
	WRITE (NT6,9151) MXARRY	VICMAIN 00650
C	6100 CONTINUE	VICMAIN 00651
	WRITE (NT6,9101) ID(1),ID(2)	VICMAIN 00652
		VICMAIN 00653
		VICMAIN 00654

WRITE (NT6,9102) PARM,IPARM	VICMAIN 00665
WRITE (NT6,9103) NFS,NMS	VICMAIN 00666
WRITE (NT6,9104) ITYPE,M,N	VICMAIN 00667
WRITE (NT6,9900)	VICMAIN 00668
C	VICMAIN 00669
CALL FLUSH(1)	VICMAIN 00670
C	VICMAIN 00671
8005 FORMAT(*OAI ARRAY NO.*,I3,*, GENERATED AT AN ACCURACY OF *,F6.4,	VICMAIN 00672
1 * IS BEING USED. *)	VICMAIN 00673
C	VICMAIN 00674
9010 FORMAT(53H0*** ERROR - WHILE READING THE GEOMETRY SCRATCH FILE A10	VICMAIN 00675
1, 15H, ERROR CODE = I4,4H ***)	VICMAIN 00676
9011 FORMAT(5X,31HAN ATTEMPT WAS MADE TO READ THE A5,8H MATRIX.//)	VICMAIN 00677
9050 FORMAT(46H0*** ERROR - WHILE READING THE PLANAR AIC FILE A10,	VICMAIN 00678
1 15H, ERROR CODE = I4,4H ***)	VICMAIN 00679
9060 FORMAT(47H0*** ERROR - WHILE READING THE SPATIAL AIC FILE A10,	VICMAIN 00680
1 15H, ERROR CODE = I4,4H ***)	VICMAIN 00681
9150 FORMAT(46H0*** ERROR - WHILE WRITING THE PLANAR AIC FILE A10,	VICMAIN 00682
1 15H, ERROR CODE = I4,4H ***)	VICMAIN 00683
9151 FORMAT(5X,32HAN ATTEMPT WAS MADE TO WRITE THE A6,8H MATRIX.//)	VICMAIN 00684
9160 FORMAT(47H0*** ERROR - WHILE WRITING THE SPATIAL AIC FILE A10,	VICMAIN 00685
1 15H, ERROR CODE = I4,4H ***)	VICMAIN 00686
C	VICMAIN 00687
9101 FORMAT(5X,**MATRIX ID = *, A10, I10)	VICMAIN 00688
9102 FORMAT(5X,**PARAMETERS *,10E11.3, /10X,*(INTEGER)*, I7, 9I11)	VICMAIN 00689
9103 FORMAT(5X,**FILE SPACING = *,I3,* MATRIX SPACING = *,I3)	VICMAIN 00690
9104 FORMAT(5X,**MATRIX TYPE -*,A10,*, DIMENSIONED (*I4,2H X,I4,1H))	VICMAIN 00691
C	VICMAIN 00692
9900 FORMAT(*O ERROR OCCURRED IN AIC SECTION (VICMAIN).*)	VICMAIN 00693
END	VICMAIN 00694

SUBROUTINE KERNEL(XMACH,K1,ERR, C, W, V)	KERNEL 00002
COMMON /FILES / NT5,NT6,INTAPE,INFS,NPLAIC,NSPAIC,NOUTP,	FILES 00002
1 IOUFSF,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC	FILES 00003
COMMON /VICPAR/ YBAR,EL,NROWS,MUATC(2,150)	KERNEL 00003
DIMENSION XIL(4),IUC(3)	KERNEL 00004
DIMENSION C(1), W(1), V(1)	KERNEL 00005
COMPLEX C,W,V, CSV,WSV,VSV	KERNEL 00006
DIMENSION CTH(2), WTH(2), VTH(2)	KERNEL 00007
EQUIVALENCE (CSV,CTH), (WSV,WTH), (VSV,VTH)	KERNEL 00008
COMMON /BESFUN/ XIB(5), A(50,5)	BCSAICB 00001
REAL K1,KIBAR	KERNEL 00010
COMPLEX ZERO	KERNEL 00011
DATA EPS / 1.0E-4 /	KERNEL 00012
C	KERNEL 00013
XMACH - MACH NUMBER	KERNEL 00014
C K1 - REDUCED FREQUENCY	KERNEL 00015
C ERR - CONVERGENCE CRITERIA (RELATIVE, NOT ABSOLUTE)	KERNEL 00016
C C - VELOCITY POTENTIAL AERODYNAMIC INFLUENCE COEFFICIENTS	KERNEL 00017
C W - UPWASH AERODYNAMIC INFLUENCE COEFFICIENTS	KERNEL 00018
C V - SIDEWASH AERODYNAMIC INFLUENCE COEFFICIENTS	KERNEL 00019
C	KERNEL 00020
ZERO = (0.,0.)	KERNEL 00021
TMACH = XMACH*XMACH	KERNEL 00022
KIBAR = (K1*TMACH)/(TMACH - 1.0)	KERNEL 00023
EL2 = EL*EL	KERNEL 00024
ITOT = 0	KERNEL 00025
NTP = 0	KERNEL 00026
IF(YBAR.EQ.0.0) NTP = -1	KERNEL 00027
DO 1000 I=1,NROWS	KERNEL 00028
C	KERNEL 00029
NTP = NTP + 2	KERNEL 00030
C	KERNEL 00031
IS = ITOT+1	KERNEL 00032
IF(EL.EQ.0.0) GO TO 50	KERNEL 00033
ITOT = ITOT + NTP	KERNEL 00034
GO TO 75	KERNEL 00035
50 CONTINUE	KERNEL 00036
ITOT = ITOT + 1	KERNEL 00037
C	KERNEL 00038
75 CONTINUE	KERNEL 00039
IF(MUATC(2,I).EQ.0) GO TO 1000	KERNEL 00040
C	KERNEL 00041
C DETERMINE IF THERE ARE ANY BOXES ON THE I-TH ROW CUT BY THE	KERNEL 00042
C MACH HYPERBOLA. (I=1 IS THE FIRST ROW)	KERNEL 00043
VBARU = FLOAT(I) - 0.5	KERNEL 00044
VBARL = VBARU - 1.0	KERNEL 00045
IF (ABS(EL)+EPS .GT. VBARU) GO TO 950	KERNEL 00046
XLOW = VBARL	KERNEL 00047
IF (ABS(EL) .GT. VBARL) XLOW = ABS(EL)	KERNEL 00048
XIB(1) = XLOW	KERNEL 00049
XINC = 0.25 * (VBARU-XLOW)	BCSAICB 00002
DO 115 J=2,5	BCSAICB 00003
XIB(J) = XIB(J-1) + XINC	KERNEL 00052
105 CONTINUE	KERNEL 00053
DO 106 J=1,250	BCSAICB 00004
A(J) = 0.0	KERNEL 00054
106 CONTINUE	KERNEL 00055

IWARN = 0	KERNEL 00057
DO 106 J=1,5	BCSAICB 00005
TAU = SQRT(XIB(J)*XIB(J) - EL2)	KERNEL 00059
TAUKM = (K1BAR/XMACH) * TAU	KERNEL 00060
XIB(J) = TAUKM	KERNEL 00061
CALL RANGE(TAUKM,N)	KERNEL 00062
IF(N.LE.100) GO TO 107	KERNEL 00063
WRITE (NT6,9005) N	KERNEL 00064
9005 FORMAT(99H0*** THE ARGUMENT FOR A BESSEL FUNCTION YIELDS AN ORDER	KERNEL 00065
1 GREATER THAN 100. ORDER REDUCED TO 100. ***)	KERNEL 00066
N = 100	KERNEL 00067
IF(IWARN.EQ.1) GO TO 107	KERNEL 00068
IWARN = 1	KERNEL 00069
107 CONTINUE	KERNEL 00070
CALL BESSEL(TAUKM,A(1,J),N)	KERNEL 00071
108 CONTINUE	KERNEL 00072
C	KERNEL 00073
C THERE ARE BOXES ON THIS ROW. FIND LEFT MOST BOX AND PROCEED	KERNEL 00074
C FROM LEFT TO RIGHT.	KERNEL 00075
ULEFT = SQRT(VBARU*VBARU - EL2)	KERNEL 00076
URIGHT= -ULEFT	KERNEL 00077
IHALF = (NTP+1)/2	KERNEL 00078
REM = ABS(YBAR) - 0.5	KERNEL 00079
IL = ULEFT - REM	KERNEL 00080
IL = IHALF - IL	KERNEL 00081
IR = REM - URIGHT + 1.0	KERNEL 00082
IR = IHALF + IR	KERNEL 00083
IF(IL.LT.MUIC(1,I)) GO TO 110	KERNEL 00084
C	KERNEL 00085
C HYPERBOLA IS LESS THAN ALLOWED, REDUCE LIMITS.	KERNEL 00086
MUIC(1,I) = IL	KERNEL 00087
GO TO 120	KERNEL 00088
C	KERNEL 00089
C HYPERBOLA CROSSED A BOUNDARY, REDUCE CALCULATIONS.	KERNEL 00090
110 IL = MUIC(1,I)	KERNEL 00091
120 CONTINUE	KERNEL 00092
C	KERNEL 00093
C TEST RIGHT SIDE	KERNEL 00094
IF(IR.GT.MUIC(2,I)) GO TO 130	KERNEL 00095
C	KERNEL 00096
C HYPERBOLA IS LESS THAN ALLOWED, REDUCE LIMITS.	KERNEL 00097
MUIC(2,I) = IR	KERNEL 00098
GO TO 140	KERNEL 00099
C	KERNEL 00100
C HYPERBOLA CROSSES A BOUNDARY, REDUCE CALCULATIONS.	KERNEL 00101
130 IR = MUIC(2,I)	KERNEL 00102
140 CONTINUE	KERNEL 00103
C	KERNEL 00104
C DETERMINE INTEGRALS FOR BOXES IL TO IR.	KERNEL 00105
IF (IL .GT. IR) GO TO 950	KERNEL 00106
DO 900 ID=IL,IR	KERNEL 00107
U = IHALF - ID	KERNEL 00108
IU = U	KERNEL 00109
ULEFT = U + 0.5 + ABS(YBAR)	KERNEL 00110
URIGHT= ULEFT - 1.0	KERNEL 00111
YUBAR = ULEFT - 0.5	KERNEL 00112
C	KERNEL 00113

CBARL = SQRT(ULEFT*ULEFT + EL2)	KERNEL 00114
CBARR = SQRT(URIGHT*URIGHT+EL2)	KERNEL 00115
C	KERNEL 00116
IF(CBARL.LE.VBARL.AND.CBARR.LE.VBARL) GO TO 500	KERNEL 00117
C	KERNEL 00118
IF(IU) 300,400,200	KERNEL 00119
C	KERNEL 00120
C BOX IS TO THE LEFT OF THE CENTER LINE OR APEX	KERNEL 00121
200 CONTINUE	KERNEL 00122
IF(CBARL.LT.VBARU) GO TO 220	KERNEL 00123
C	KERNEL 00124
C EDGE BOX, ONLY HAS ONE SEGMENT TO INTEGRATE	KERNEL 00125
NINT = 1	KERNEL 00126
XIL(1) = CBARR	KERNEL 00127
XIL(2) = VBARU	KERNEL 00128
IUC(1) = 1	KERNEL 00129
GO TO 700	KERNEL 00130
C	KERNEL 00131
C DOUBLY CUT BOX, HAS TWO SEGMENTS TO INTEGRATE	KERNEL 00132
220 NINT = 2	KERNEL 00133
XIL(1) = VBARL	KERNEL 00134
IF(CBARR.GT.VBARL) XIL(1) = CBARR	KERNEL 00135
XIL(2) = CBARL	KERNEL 00136
XIL(3) = VBARU	KERNEL 00137
IUC(1) = 1	KERNEL 00138
IUC(2) = 0	KERNEL 00139
GO TO 700	KERNEL 00140
C	KERNEL 00141
C BOX IS ON THE RIGHT OF THE CENTER LINE OR APEX	KERNEL 00142
300 CONTINUE	KERNEL 00143
IF(CBARR.LT.VBARU) GO TO 320	KERNEL 00144
C	KERNEL 00145
C EDGE BOX, HAS ONLY ONE SEGMENT	KERNEL 00146
NINT = 1	KERNEL 00147
XIL(1) = CBARL	KERNEL 00148
XIL(2) = VBARU	KERNEL 00149
IUC(1) = 2	KERNEL 00150
GO TO 700	KERNEL 00151
C	KERNEL 00152
C DOUBLY CUT BOX, HAS TWO SEGMENTS	KERNEL 00153
320 NINT = 2	KERNEL 00154
XIL(1) = VBARL	KERNEL 00155
IF(CBARL.GT.VBARL) XIL(1) = CBARR	KERNEL 00156
XIL(2) = CBARR	KERNEL 00157
XIL(3) = VBARU	KERNEL 00158
IUC(1) = 2	KERNEL 00159
IUC(2) = 0	KERNEL 00160
GO TO 700	KERNEL 00161
C	KERNEL 00162
C CENTER LINE OR APEX BOX	KERNEL 00163
400 CONTINUE	KERNEL 00164
IF(ABS(EL).LT.VBARL) GO TO 475	KERNEL 00165
IF(CBARL.LT.VBARU.OR.CBARR.LT.VBARU) GO TO 420	KERNEL 00166
C	KERNEL 00167
C ONLY BOX ON ROW, ONLY ONE SEGMENT TO INTEGRATE	KERNEL 00168
NINT = 1	KERNEL 00169
XIL(1) = ABS(EL)	KERNEL 00170

```

      XIL(2) = VBARU
      IUC(1) = 3
      GO TO 700
C
420 CONTINUE
      IF(CBARL.LT.VBARU.AND.CBARR.LT.VBARU) GO TO 440
C
      ONLY HAS 2 SEGMENTS
      NENT = 2
      XIL(1) = ABS(EL)
      XIL(2) = CBARR
      XIL(3) = VBARU
      IUC(1)=3
      IUC(2)=1
      GO TO 700
C
440 CONTINUE
C
      UNLESS THE HYPERBOLA CENTER IS ON A BOX SIDE LINE,
C      I.E. YBAR = 0.5, THEN THERE WILL BE 2 SEGMENTS.
C
      IF(ABS(YBAR).NE.0.5) GO TO 445
      XIL(1) = ABS(EL)
      XIL(2) = CBARL
      XIL(3) = VBARU
      IUC(1) = 1
      IUC(2) = 0
      NENT = 2
      GO TO 700
C
445 CONTINUE
C      WILL HAVE THREE SEGMENTS IF YBAR .NE. ZERO
      XIL(1) = ABS(EL)
      XIL(2) = CBARR
      IUC(1)=3
      IF(ABS(YBAR).NE.0.0) GO TO 450
C
      TWO SEGMENTS
      NENT = 2
      XIL(3) = VBARU
      IUC(2)=0
      GO TO 700
C
      THREE SEGMENTS
C450 NENT = 3
      XIL(3) = CBARL
      IUC(2)=1
      XIL(4) = VBARU
      IUC(3)=0
      GO TO 700
C
      CENTER LINE BOX, BUT NOT APEX, HAS THREE SEGMENTS
C475 CONTINUE
      IF(CBARR.LE.VBARL) GO TO 220
      NENT = 3
      XIL(1) = VBARL
      XIL(2) = CBARR

```

```

KERNEL 00171
KERNEL 00172
KERNEL 00173
KERNEL 00174
KERNEL 00175
KERNEL 00176
KERNEL 00177
KERNEL 00178
KERNEL 00179
KERNEL 00180
KERNEL 00181
KERNEL 00182
KERNEL 00183
KERNEL 00184
KERNEL 00185
KERNEL 00186
KERNEL 00187
KERNEL 00188
KERNEL 00189
KERNEL 00190
KERNEL 00191
KERNEL 00192
KERNEL 00193
KERNEL 00194
KERNEL 00195
KERNEL 00196
KERNEL 00197
KERNEL 00198
KERNEL 00199
KERNEL 00200
KERNEL 00201
KERNEL 00202
KERNEL 00203
KERNEL 00204
KERNEL 00205
KERNEL 00206
KERNEL 00207
KERNEL 00208
KERNEL 00209
KERNEL 00210
KERNEL 00211
KERNEL 00212
KERNEL 00213
KERNEL 00214
KERNEL 00215
KERNEL 00216
KERNEL 00217
KERNEL 00218
KERNEL 00219
KERNEL 00220
KERNEL 00221
KERNEL 00222
KERNEL 00223
KERNEL 00224
KERNEL 00225
KERNEL 00226
KERNEL 00227

```


XIL(3) = CBARL	KERNEL 00228
XIL(4) = VBARU	KERNEL 00229
IUC(1) = 3	KERNEL 00230
IUC(2) = 1	KERNEL 00231
IUC(3) = 0	KERNEL 00232
GO TO 700	KERNEL 00233
C	KERNEL 00234
C FULL BOX, ONLY ONE SEGMENT TO INTEGRATE	KERNEL 00235
500 CONTINUE	KERNEL 00236
NINT = 1	KERNEL 00237
XIL(1) = VBARL	KERNEL 00238
XIL(2) = VBARU	KERNEL 00239
IUC(1) = 0	KERNEL 00240
C	KERNEL 00241
C LIMITS AND TYPES FOR ALL SEGMENTS ARE COMPLETED. INTEGRATE.	KERNEL 00242
700 CONTINUE	KERNEL 00243
IDX = ID	KERNEL 00244
IF(EL.EQ.0.0) IDX = ID - I + 1	KERNEL 00245
IX = IS + IDX - 1	KERNEL 00246
IF(C(IX).NE.0) GO TO 900	KERNEL 00247
DO 800 INT=1,NINT	KERNEL 00248
CSV = (0.,0.)	KERNEL 00249
WSV = (0.,0.)	KERNEL 00250
VSV = (0.,0.)	KERNEL 00251
IFLAG=0	KERNEL 00252
C	KERNEL 00253
C CALL ROMBERG INTEGRATION FOR REAL PART	KERNEL 00254
CALL ROMBER(XIL(INT),XIL(INT+1),IUC(INT),ERR,IFLAG,K1BAR,YMUBAR,	KERNEL 00255
1 EL,XMACH,CTM(1),WTM(1),VTM(1))	KERNEL 00256
IF(K1.EQ.0.0) GO TO 750	KERNEL 00257
IFLAG = 1	KERNEL 00258
C	KERNEL 00259
C CALL ROMBERG INTEGRATION FOR IMAGINARY PART	KERNEL 00260
CALL ROMBER(XIL(INT),XIL(INT+1),IUC(INT),ERR,IFLAG,K1BAR,YMUBAR,	KERNEL 00261
1 EL,XMACH,CTM(2),WTM(2),VTM(2))	KERNEL 00262
750 CONTINUE	KERNEL 00263
C(IX) = C(IX) + CSV	KERNEL 00264
W(IX) = W(IX) + WSV	KERNEL 00265
V(IX) = V(IX) + VSV	KERNEL 00266
800 CONTINUE	KERNEL 00267
C	KERNEL 00268
900 CONTINUE	KERNEL 00269
GO TO 1000	KERNEL 00270
C	KERNEL 00271
950 CONTINUE	KERNEL 00272
MUAIC(1,I) = 0	KERNEL 00273
MUAIC(2,I) = 0	KERNEL 00274
C	KERNEL 00275
1000 CONTINUE	KERNEL 00276
RETURN	KERNEL 00277
END	KERNEL 00278

SUBROUTINE ROMBER(XILL,XILU,IUC,ERR,IFLAG,K1BAR,YMUBAR,EL ,XMACH,	ROMBER	00002
1 C, W, V)	ROMBER	00003
DIMENSION XI(512),FXIC(512),FXIW(512),FXIV(512)	ROMBER	00004
DIMENSION A(11,11),AW(11,11),AV(11,11),VT(2)	ROMBER	00005
REAL K1BAR	ROMBER	00006
PIE = 3.141592654	ROMBER	00007
C	ROMBER	00008
C XILL - XI LOWER LIMIT OF INTEGRATION	ROMBER	00009
C XILU - XI UPPER LIMIT OF INTEGRATION	ROMBER	00010
C IUC - FLAG INDICATING TYPE OF BOX OR EDGE CONDITION OF	ROMBER	00011
C INTERVAL TO BE INTEGRATED.	ROMBER	00012
C IUC = 0, FULL BOX	ROMBER	00013
C = 1, LEFT SIDE OF INTERVAL IS EDGE OF MACH HYP.	ROMBER	00014
C = 2, RIGHT SIDE OF INTERVAL IS EDGE OF MACH HYP.	ROMBER	00015
C = 3, BOTH SIDES OF INTERVAL IS EDGE OF MACH HYP.	ROMBER	00016
C ERR - CONVERGENCE TEST CRITERIA	ROMBER	00017
C IFLAG - INDICATOR OF REAL OR IMAGINARY PARTS	ROMBER	00018
C IFLAG = 0, REAL PART	ROMBER	00019
C = 1, IMAGINARY PART	ROMBER	00020
C K1BAR - FUNCTION OF REDUCED FREQUENCY, MACH NUMBER	ROMBER	00021
C YMUBAR - COORDINATE HORIZONTALLY OF PULSE SENDING BOX	ROMBER	00022
C EL - DISTANCE OF RECEIVING BOX ABOVE SENDING PLANE	ROMBER	00023
C XMACH - MACH NUMBER	ROMBER	00024
C C - C COEFFICIENT	ROMBER	00025
C V - V COEFFICIENT	ROMBER	00026
C W - W COEFFICIENT	ROMBER	00027
C	ROMBER	00028
C	ROMBER	00029
C CALCULATE INITIAL VALUES AT END POINTS	ROMBER	00030
C	ROMBER	00031
EL2 = EL*EL	ROMBER	00032
WK = -XMACH/(PIE*K1BAR)	ROMBER	00033
XI(1) = XILL	ROMBER	00034
XI(2) = XILU	ROMBER	00035
CALL FUNCT(2,XI,FXIC,FXIW,IFLAG,K1BAR,EL ,YMUBAR,IUC,XMACH,BESSW)	ROMBER	00036
IF(EL.EQ.0.0) GO TO 101	ROMBER	00037
IF(K1BAR.EQ.0.0) GO TO 101	ROMBER	00038
IF(IUC.EQ.3) GO TO 101	ROMBER	00039
CALL VFUNC (2, XI,FXIV,IFLAG,K1BAR,EL,YMUBAR,IUC,XMACH,1,VT)	ROMBER	00040
101 CONTINUE	ROMBER	00041
TERM1 = FXIC(1)/XI(1)	ROMBER	00042
TERM2 = FXIC(2)/XI(2)	ROMBER	00043
HINT = 0.5*(XILU-XILL)	ROMBER	00044
TC = HINT *(FXIC(1)+FXIC(2))	ROMBER	00045
TW = HINT *(FXIW(1)+FXIW(2))	ROMBER	00046
A(1,1) = TC	ROMBER	00047
AW(1,1) = TW	ROMBER	00048
IF(EL.EQ.0.0) GO TO 102	ROMBER	00049
IF(K1BAR.EQ.0.0) GO TO 102	ROMBER	00050
IF(IUC.EQ.3) GO TO 102	ROMBER	00051
TV = HINT *(FXIV(1)+FXIV(2))	ROMBER	00052
AV(1,1) = TV	ROMBER	00053
102 CONTINUE	ROMBER	00054
C	ROMBER	00055
DO 30 M=2,11	ROMBER	00056
J = 2*(M-1)	ROMBER	00057
H = (XILU - XILL)/I	ROMBER	00058

C		ROMBER	00059
C	DETERMINE XI LOCATIONS TO EVALUATE FUNCTION AT	ROMBER	00060
	DO 3 J=2,1,2	ROMBER	00061
	JJ = J/2	ROMBER	00062
	XI(JJ) = XILL + (J-1)*H	ROMBER	00063
	3 CONTINUE	ROMBER	00064
	CALL FUNCT(JJ,XI,FXIC,FXIW,IFLAG,K1BAR,EL,YMUBAR,IUC,XMACH,DUMMY)	ROMBER	00065
	IF(EL.EQ.0.0) GO TO 103	ROMBER	00066
	IF(K1BAR.EQ.0.0) GO TO 103	ROMBER	00067
	IF(IUC.EQ.3) GO TO 103	ROMBER	00068
	CALL VFUNC(JJ,XI,FXIV,IFLAG,K1BAR,EL,YMUBAR,IUC,XMACH,0,DUMMY)	ROMBER	00069
	103 CONTINUE	ROMBER	00070
C		ROMBER	00071
C	DETERMINE TRAPEZOIDAL AREA WITH THE NEW FUNCTION EVALUATIONS	ROMBER	00072
	TMNC = 0.0	ROMBER	00073
	TMNW = 0.0	ROMBER	00074
	TMNV = 0.0	ROMBER	00075
	DO 5 J=1,JJ	ROMBER	00076
	TMNC = TMNC + FXIC(J)	ROMBER	00077
	TMNW = TMNW + FXIW(J)	ROMBER	00078
	IF(EL.EQ.0.0) GO TO 5	ROMBER	00079
	IF(K1BAR.EQ.0.0) GO TO 5	ROMBER	00080
	IF(IUC.NE.3) TMNV = TMNV + FXIV(J)	ROMBER	00081
	5 CONTINUE	ROMBER	00082
	TC = 0.5*TC + H*TMNC	ROMBER	00083
	TW = 0.5*TW + H*TMNW	ROMBER	00084
C		ROMBER	00085
C	PUT THE NEW AREAS INTO THE ARRAY AND PERFORM EXTRAPOLATION	ROMBER	00086
	A(M,1) = TC	ROMBER	00087
	AW(M,1) = TW	ROMBER	00088
	IF(EL.EQ.0.0) GO TO 104	ROMBER	00089
	IF(K1BAR.EQ.0.0) GO TO 104	ROMBER	00090
	IF(IUC.EQ.3) GO TO 104	ROMBER	00091
	TV = 0.5*TV + H*TMNV	ROMBER	00092
	AV(M,1) = TV	ROMBER	00093
	104 CONTINUE	ROMBER	00094
	DO 10 N=2,M	ROMBER	00095
	A(M,N) = ((4**(N-1))*A(M,N-1)-A(M-1,N-1))/(4**(N-1)-1)	ROMBER	00096
	AW(M,N) = ((4**(N-1))*AW(M,N-1)-AW(M-1,N-1))/(4**(N-1)-1)	ROMBER	00097
	IF(EL.EQ.0.0) GO TO 10	ROMBER	00098
	IF(K1BAR.EQ.0.0) GO TO 10	ROMBER	00099
	IF(IUC.EQ.3) GO TO 10	ROMBER	00100
	AV(M,N) = ((4**(N-1))*AV(M,N-1)-AV(M-1,N-1))/(4**(N-1)-1)	ROMBER	00101
	10 CONTINUE	ROMBER	00102
C		ROMBER	00103
C	DETERMINE IF THE TECHNIQUE HAS REACHED SUFFICIENT CONVERGENCE	ROMBER	00104
	C = A(M,M)	ROMBER	00105
	W = AW(M,M)	ROMBER	00106
	IF(EL.EQ.0.0) GO TO 105	ROMBER	00107
	IF(K1BAR.EQ.0.0) GO TO 105	ROMBER	00108
	IF(IUC.EQ.3) GO TO 105	ROMBER	00109
	V = AV(M,M)	ROMBER	00110
	105 CONTINUE	ROMBER	00111
	REXR = ABS(ERR*C)	ROMBER	00112
	MM1 = M-1	ROMBER	00113
	UCIF = ABS(A(M,MM1)-A(MM1,MM1))	ROMBER	00114
	RDIF = ABS(A(M,M) - A(M,MM1))	ROMBER	00115

DELS = 0.5*(UDIF+RDIF)	ROMBER 00116
IF(DELS.GT.RERR) GO TO 30	ROMBER 00117
C	ROMBER 00118
C HAS CONVERGED, TEST FOR W CONVERGENCE	ROMBER 00119
IF(EL.EQ.0.0) GO TO 50	ROMBER 00120
RERR = ABS(ERR*W)	ROMBER 00121
UDIF = ABS(AW(M,MM1) - AW(MM1,MM1))	ROMBER 00122
RDIF = ABS(AW(M,M) - AW(M,MM1))	ROMBER 00123
DELS = 0.5 *(UDIF + RDIF)	ROMBER 00124
IF(DELS.GT.RERR) GO TO 30	ROMBER 00125
C	ROMBER 00126
C AND W HAVE CONVERGED, TEST FOR V CONVERGENCE	ROMBER 00127
C	ROMBER 00128
IF(K1BAR.EQ.0.0 .OR. IUC.EQ.3) GO TO 50	ROMBER 00129
RERR = ABS(ERR*V)	ROMBER 00130
UDIF = ABS(AV(M,MM1) - AV(MM1,MM1))	ROMBER 00131
RDIF = ABS(AV(M,M) - AV(MM1,MM1))	ROMBER 00132
DELS = 0.5 *(UDIF + RDIF)	ROMBER 00133
IF(DELS.LE.RERR) GO TO 50	ROMBER 00134
C	ROMBER 00135
C HAS NOT CONVERGED MAKE ANOTHER LOOP.	ROMBER 00136
30 CONTINUE	ROMBER 00137
50 CONTINUE	ROMBER 00138
C = -C/PIE	ROMBER 00139
IF(EL.EQ.0.0) GO TO 70	ROMBER 00140
W = (EL/PIE)*(W*TERM2-TERM1 + BESSW)	ROMBER 00141
IF(K1BAR.EQ.0.0) GO TO 65	ROMBER 00142
IF(IUC.EQ.3) GO TO 70	ROMBER 00143
V = VK*(V+VT(2)-VT(1))	ROMBER 00144
GO TO 70	ROMBER 00145
65 CONTINUE	ROMBER 00146
ETAL = YMUBAR - 0.5	ROMBER 00147
IF(IUC.EQ.2) ETAL = - SQRT(XILU*XILU - EL2)	ROMBER 00148
ETAU = YMUBAR + 0.5	ROMBER 00149
IF(IUC.EQ.1) ETAU = SQRT(XILU*XILU - EL2)	ROMBER 00150
XILU2 = XILU * XILU	ROMBER 00151
XILL2 = XILL * XILL	ROMBER 00152
ETAU2 = ETAU * ETAU	ROMBER 00153
ETAL2 = ETAL * ETAL	ROMBER 00154
SU = ETAU2 + EL2	ROMBER 00155
SL = ETAL2 + EL2	ROMBER 00156
S1 = XILU2 - SU	ROMBER 00157
S2 = XILU2 - SL	ROMBER 00158
S3 = XILL2 - SU	ROMBER 00159
S4 = XILL2 - SL	ROMBER 00160
V1 = 0.0	ROMBER 00161
V2 = 0.0	ROMBER 00162
V3 = 0.0	ROMBER 00163
V4 = 0.0	ROMBER 00164
IF(S1.GT.0.0) V1 = ALOG((XILU+SQRT(S1))/SQRT(SU))	ROMBER 00165
IF(S2.GT.0.0) V2 = ALOG((XILU+SQRT(S2))/SQRT(SL))	ROMBER 00166
IF(S3.GT.0.0) V3 = ALOG((XILL+SQRT(S3))/SQRT(SU))	ROMBER 00167
IF(S4.GT.0.0) V4 = ALOG((XILL+SQRT(S4))/SQRT(SL))	ROMBER 00168
V = (-1.0/PIE)*(V1-V2-V3+V4)	ROMBER 00169
70 CONTINUE	ROMBER 00170
RETURN	ROMBER 00171
END	ROMBER 00172

SUBROUTINE FUNCT(K,XI,FXIC,FXIW,IFLAG,K1BAR,EL ,YMUBAR,IUC,	FUNCT	00002
1 XMACH,BESSY)	FUNCT	00003
DIMENSION XI(512),FXIC(512),FXIW(512),A(50)	FUNCT	00004
C	FUNCT	00005
C K - NUMBER OF FUNCTIONS TO EVALUATE	FUNCT	00006
C XI - VARIABLE OF INTEGRATION	FUNCT	00007
C FXIC - FUNCTIONAL VALUE FOR C EQUATION	FUNCT	00008
C FXIW - FUNCTIONAL VALUE FOR W EQUATION	FUNCT	00009
C IFLAG - INDICATOR OF REAL OR IMAGINARY PARTS	FUNCT	00010
C IFLAG = 0, REAL PART	FUNCT	00011
C IFLAG = 1, IMAGINARY PART	FUNCT	00012
C K1BAR - FUNCTION OF REDUCED FREQUENCY AND MACH NUMBER	FUNCT	00013
C EL - DISTANCE OF RECEIVING BOX ABOVE SENDING PLANE	FUNCT	00014
C YMUBAR - COORDINATE HORIZONTALLY OF PULSE SENDING BOX	FUNCT	00015
C IUC - FLAG INDICATING TYPE OF BOX OR EDGE CONDITION OF	FUNCT	00016
C INTERVAL TO BE INTEGRATED.	FUNCT	00017
C XMACH - MACH NUMBER	FUNCT	00018
C BESSY - EVALUATION OF END POINTS FOR W COEFFICIENTS.	FUNCT	00019
C	FUNCT	00020
REAL K1BAR	FUNCT	00021
PIE = 3.141592654	FUNCT	00022
PIE2 = 1.570796327	FUNCT	00023
EL2 = EL*EL	FUNCT	00024
BESSY = 0.0	FUNCT	00025
C	FUNCT	00026
DO 1000 I=1,K	FUNCT	00027
C	FUNCT	00028
C SET UP CONSTANTS	FUNCT	00029
TAU = SQRT(XI(I)*XI(I) - EL2)	FUNCT	00030
TAUKM = (K1BAR/XMACH)*TAU	FUNCT	00031
EPOW = K1BAR*XI(I)	FUNCT	00032
IF(ABS(TAU).LT.1.0E-06) GO TO 25	FUNCT	00033
THETAU = (YMUBAR + 0.5)/TAU	FUNCT	00034
THETAL = (YMUBAR-0.5)/TAU	FUNCT	00035
GO TO 50	FUNCT	00036
25 CONTINUE	FUNCT	00037
THETAL = 0.0	FUNCT	00038
THETAU = 0.0	FUNCT	00039
50 CONTINUE	FUNCT	00040
C	FUNCT	00041
IF(IFLAG.EQ.0) GO TO 100	FUNCT	00042
C	FUNCT	00043
C IMAGINARY PART	FUNCT	00044
EXPW = -SIN(EPOW)	FUNCT	00045
EXPW = (EPOW*COB(EPOW) - SIN(EPOW))/(XI(I)*XI(I))	FUNCT	00046
GO TO 200	FUNCT	00047
C	FUNCT	00048
C REAL PART	FUNCT	00049
100 EXPW = COB(EPOW)	FUNCT	00050
EXPW = (COB(EPOW) + EPOW*SIN(EPOW))/(XI(I)*XI(I))	FUNCT	00051
C	FUNCT	00052
200 CONTINUE	FUNCT	00053
CALL BFUNC(TAUKM,A,N)	FUNCT	00054
IF(EL.EQ.0.0) GO TO 250	FUNCT	00055
IF(I.GT.1) GO TO 250	FUNCT	00056
IF(IUC.NE.3) GO TO 250	FUNCT	00057
EXL = ABS(EL) - XI(I)	FUNCT	00058

IF (ABS (EXL) .GT. 1.0E-05) GO TO 250	FUNCT	00059
BESSY = (EXPX*PIE)/EL	FUNCT	00060
250 CONTINUE	FUNCT	00061
BESSO = A(1)	FUNCT	00062
PTERM = 0.0	FUNCT	00063
IF (IUC.EQ.0.OR.IUC.EQ.2) GO TO 300	FUNCT	00064
C	FUNCT	00065
C LEFT SIDE IS BOUNDARY CONDITION	FUNCT	00066
C1 = PIE2	FUNCT	00067
GO TO 400	FUNCT	00068
C	FUNCT	00069
300 CONTINUE	FUNCT	00070
IF (ABS (THETAU) .GE. 1.0) GO TO 350	FUNCT	00071
C1 = ASIN (THETAU)	FUNCT	00072
GO TO 400	FUNCT	00073
350 CONTINUE	FUNCT	00074
C1 = SIGN (PIE2, THETAU)	FUNCT	00075
C	FUNCT	00076
400 CONTINUE	FUNCT	00077
IF (IUC.LE.1) GO TO 500	FUNCT	00078
C	FUNCT	00079
C RIGHT SIDE IS BOUNDARY CONDITION	FUNCT	00080
C2 = -PIE2	FUNCT	00081
GO TO 600	FUNCT	00082
C	FUNCT	00083
500 CONTINUE	FUNCT	00084
IF (ABS (THETAL) .GE. 1.0) GO TO 550	FUNCT	00085
C2 = ASIN (THETAL)	FUNCT	00086
GO TO 600	FUNCT	00087
550 CONTINUE	FUNCT	00088
C2 = SIGN (PIE2, THETAL)	FUNCT	00089
600 CONTINUE	FUNCT	00090
C	FUNCT	00091
IF (IUC.EQ.3) GO TO 900	FUNCT	00092
IF (N .EQ. 1) GO TO 900	FUNCT	00093
C	FUNCT	00094
SIGNX = -1.0	FTNXL	00049
R = 0	FUNCT	00096
PSIGN = 1.0	FUNCT	00097
N = (N+1)/2	FUNCT	00098
DO 800 IR=2,N	FUNCT	00099
R = R +1	FUNCT	00100
PSIGN = PSIGN * SIGNX	FTNXL	00050
PTERM = PSIGN/R	FUNCT	00102
BTERM = BTERM + PTERM*(IR)*(SIN(2.0*R*C1) - SIN(2.0*R*C2))	FUNCT	00103
800 CONTINUE	FUNCT	00104
C	FUNCT	00105
900 CONTINUE	FUNCT	00106
FXIC(I) = EXPX*(BESSO*(C1-C2) + BTERM)	FUNCT	00107
FXIW(I) = 0.0	FUNCT	00108
IF (EL2.EQ.0.0) GO TO 1000	FUNCT	00109
FXIW(I) = EXPX*(BESSO*(C1-C2) + BTERM)	FUNCT	00110
C	FUNCT	00111
1000 CONTINUE	FUNCT	00112
RETURN	FUNCT	00113
END	FUNCT	00114

SUBROUTINE BESSEL(K12,A,NA)	BESSEL 00002
DIMENSION A(1), AV(150)	BESSEL 00003
REAL K12	BESSEL 00004
C	BESSEL 00005
C K12 - FUNCTION OF XI VALUE, MACH NUMBER AND REDUCED FREQUENCY	BESSEL 00006
C A = EVALUATION OF THE BESSEL FUNCTION	BESSEL 00007
C NA - ORDER OF THE BESSEL FUNCTION TO BE EVALUATED	BESSEL 00008
C	BESSEL 00009
ALPHA = 1.E-25	BESSEL 00010
NT = NA + 1	BESSEL 00011
BETA=.0008	BESSEL 00012
IF(K12-BETA) 76,76,78	BESSEL 00013
78 CONTINUE	BESSEL 00014
IF(K12-ALPHA)76,76,20	BESSEL 00015
76 CONTINUE	BESSEL 00016
(2) = 0.	BESSEL 00017
A(1) = 1.	BESSEL 00018
GO TO 99	BESSEL 00019
20 SUM = 0.	BESSEL 00020
K = 1.5*K12 + 1.	BESSEL 00021
NP = MAX0(K,NT)	BESSEL 00022
I = NP+11	BESSEL 00023
AV(I+2) = 0.0	BESSEL 00024
AV(I+1) = ALPHA	BESSEL 00025
30 AV(I) = AV(I+1)*I*2./K12-AV(I+2)	BESSEL 00026
IF (I-1) 40, 40, 50	BESSEL 00027
50 IF (MOD(I,2)) 60, 70, 60	BESSEL 00028
60 SUM = SUM + AV(I)	BESSEL 00029
70 I = I-1	BESSEL 00030
GO TO 30	BESSEL 00031
40 C = 1./(2.*SUM+AV(1))	BESSEL 00032
I = 1	BESSEL 00033
DO 90 II=1,NA,2	BESSEL 00034
A(II) = AV(II) * C	BESSEL 00035
I = I + 1	BESSEL 00036
IF(I.EQ.50) GO TO 99	BESSEL 00037
90 CONTINUE	BESSEL 00038
99 CONTINUE	BESSEL 00039
RETURN	BESSEL 00040
END	

<pre> SUBROUTINE RANGE(K12,NA) REAL K12 C C CALCULATES THE RANGE ON THE VARIABLE N FOR SUBROUTINE BESSL C C C K12 = FUNCTION OF X VALUE, MACH NUMBER AND REDUCED C FREQUENCY C NA - ORDER OF THE BESSEL FUNCTION TO BE EVALUATED C 400 CONTINUE C IF(K12- 0.01) 101,98,98 98 IF(K12- 3.00) 102,102,99 99 IF(K12-19.00) 103,103,100 100 GO TO 104 C 101 CONTINUE NA=4 RETURN 102 CONTINUE NA= 3.0*K12 +7.0 RETURN 103 CONTINUE NA= 2.0*K12 +7.0 RETURN 104 CONTINUE NA= (10.0/9.)*K12 + 29. RETURN END </pre>	<pre> RANGE 00002 RANGE 00003 RANGE 00004 RANGE 00005 RANGE 00006 RANGE 00007 RANGE 00008 RANGE 00009 RANGE 00010 RANGE 00011 RANGE 00012 RANGE 00013 RANGE 00014 RANGE 00015 RANGE 00016 RANGE 00017 RANGE 00018 RANGE 00019 RANGE 00020 RANGE 00021 RANGE 00022 RANGE 00023 RANGE 00024 RANGE 00025 RANGE 00026 RANGE 00027 RANGE 00028 RANGE 00029 RANGE 00030 RANGE 00031 </pre>

SUBROUTINE VFUNC(K,XI,FXIV,IFLAG,KIBAR,EL,YMUBAR,IUC,XMACH,IND,VT)	VFUNCT	00002
DIMENSION XI(256),FXIV(256),VT(2)	VFUNCT	00003
REAL KIBAR	VFUNCT	00004
C	VFUNCT	00005
C THIS PROGRAM CALCULATES THE FUNCTION VALUES OF INTEGRATION	VFUNCT	00006
C FOR THE V COEFFICIENTS.	VFUNCT	00007
C	VFUNCT	00008
C K - NUMBER OF VALUES TO CALCULATE	VFUNCT	00009
C XI - VARIABLE ARRAY AT WHICH VALUES ARE CALCULATED.	VFUNCT	00010
C FXIV - FUNCTIONAL VALUES	VFUNCT	00011
C IFLAG = FLAG INDICATING REAL OR COMPLEX PART	VFUNCT	00012
C IFLAG = 0, REAL PART	VFUNCT	00013
C = 1, IMAGINARY PART	VFUNCT	00014
C KIBAR - FUNCTION OR REDUCED FREQUENCY AND MACH NUMBER.	VFUNCT	00015
C EL - DISTANCE OF RECEIVING BOX ABOVE SENDING PLANE.	VFUNCT	00016
C YMUBAR - COORDINATE HORIZONTALLY OF PULSE SENDING PLANE.	VFUNCT	00017
C IUC - FLAG INDICATING TYPE OF BOX OR EDGE CONDITION OF	VFUNCT	00018
C INTERVAL TO BE INTEGRATED.	VFUNCT	00019
C XMACH - MACH NUMBER	VFUNCT	00020
C IND - INDICATOR TO CALCULATE VT TERMS	VFUNCT	00021
C = 0, DO NOT CALCULATE	VFUNCT	00022
C = 1, CALCULATE	VFUNCT	00023
C VT - EXTRA TERMS CALCULATE AT THE LIMITS OF INTEGRATION	VFUNCT	00024
C	VFUNCT	00025
EPS = 1.0E-04	VFUNCT	00026
EL2 = EL*EL	VFUNCT	00027
DO 500 I=1,K	VFUNCT	00028
C	VFUNCT	00029
C CALCULATE CONSTANTS	VFUNCT	00030
EPOW = KIBAR*XI(I)	VFUNCT	00031
XI2 = XI(I)*XI(I)	VFUNCT	00032
FREQM = KIBAR/XMACH	VFUNCT	00033
YMUP2 = (YMUBAR + 0.5)*(YMUBAR + 0.5)	VFUNCT	00034
YMU2 = (YMUBAR - 0.5)*(YMUBAR - 0.5)	VFUNCT	00035
C	VFUNCT	00036
IF(IFLAG.EQ.0) GO TO 100	VFUNCT	00037
C	VFUNCT	00038
C IMAGINARY PART	VFUNCT	00039
EXPN = (EPOW*COS(EPOW)-SIN(EPOW))/XI2	BCSAICA	00002
IF(IND.EQ.1) EXPNV = -SIN(EPOW)	BCSAICA	00003
GO TO 200	VFUNCT	00042
C	VFUNCT	00043
C REAL PART	VFUNCT	00044
100 EXPN = (COS(EPOW)+EPOW*SIN(EPOW))/XI2	BCSAICA	00004
IF(IND.EQ.1) EXPNV = COS(EPOW)	BCSAICA	00005
C	VFUNCT	00047
200 CONTINUE	VFUNCT	00048
C1 = 0.0	VFUNCT	00049
C2 = 0.0	VFUNCT	00050
IF(IUC.EQ.1) GO TO 300	VFUNCT	00051
C	VFUNCT	00052
C1R = XI2 - YMUP2 - EL2	VFUNCT	00053
IF(ABS(C1R).LT.EPS) GO TO 300	VFUNCT	00054
C1 = SIN(FREQM*SQRT(C1R))	VFUNCT	00055
C	VFUNCT	00056
300 CONTINUE	VFUNCT	00057
IF(IUC.EQ.2) GO TO 400	VFUNCT	00058

```

C
  C2R = XI2 - YMU2 - EL2
  IF (ABS(C2R).LT.EPS) GO TO 400
  C2 = SIN(FREQMSQRT(C2R))
C
400 CONTINUE
  FXIV(I) = EXPN * (C1-C2)
  IF (IND.EQ.1) VT(I) = (EXPV/XI(I)) * (C1-C2)
500 CONTINUE
  RETURN
  END

```

```

VFUNCT 00059
VFUNCT 00060
VFUNCT 00061
VFUNCT 00062
VFUNCT 00063
VFUNCT 00064
BCSAICA 00066
BCSAICA 00067
VFUNCT 00067
VFUNCT 00068
VFUNCT 00069

```

<pre> SUBROUTINE BFUNC(X,AV,N) C C X - ARGUMENT FO THE BESSEL FUNCTION C AV - EVALUATION OF BESSEL FUNCTION C N - MAX ORDER OF BESSEL FUNCTION C C COMMON /FILES / NT5,NT6,INTAPE,INFSP,NFLAIC,NSPATC,NOUTP, C IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC COMMON /BESFUN/ XIB(5), A(50,5) DIMENSION AV(1) C C FIND INTERVAL X IS IN IF(X.GE.XIB(1).AND.X.LE.XIB(5)) GO TO 50 EPS = 1.0E-04 I = 1 IF (ABS(X-XIB(I)) .LE. EPS*XIB(I)) GO TO 150 I = 5 IF (ABS(X-XIB(I)) .LE. EPS*XIB(I)) GO TO 150 WRITE (NT6,9005) X,XIB(1),XIB(5) 9005 FORMAT(68H*** ERROR - THE ARGUMENT FOR A BESSEL FUNCTION IS OUT 1OF RANGE. ***/ 14HDARGUMENT = F11.6/14H LOWER LIMIT = F11.6 / 2 14H UPPER LIMIT = F11.6) CALL FLUSH(1) 50 CONTINUE I = 2 100 IF(X-XIB(I)) 200,150,125 125 CONTINUE I = I + 1 GO TO 100 150 CONTINUE C C X EQUALS XIB(I) DO NOT INTERPOLATE C C N = 1 160 CONTINUE AV(N) = A(N,I) IF(A(N+1,I).EQ.0) GO TO 400 N = N + 1 GO TO 160 200 CONTINUE DXX = (X-XIB(I-1))/(XIB(I) -XIB(I-1)) N = 1 300 CONTINUE AV(N) = A(N,I-1) + DXX * (A(N,I) - A(N,I-1)) IF(A(N+1,I-1).EQ.0.) GO TO 400 N = N + 1 GO TO 300 400 CONTINUE RETURN END </pre>	<pre> BFUNC 00002 BFUNC 00003 BFUNC 00004 BFUNC 00005 BFUNC 00006 BFUNC 00007 FILES 00002 FILES 00003 BCSAICB 00006 BFUNC 00010 BFUNC 00011 BFUNC 00012 BCSAICB 00007 BFUNC 00014 BFUNC 00015 BFUNC 00016 BCSAICB 00008 BFUNC 00018 BCSAICB 00009 BFUNC 00020 BFUNC 00021 BFUNC 00022 BFUNC 00023 BFUNC 00024 BFUNC 00025 BFUNC 00026 BFUNC 00027 BFUNC 00028 BFUNC 00029 BFUNC 00030 BFUNC 00031 BFUNC 00032 BFUNC 00033 BFUNC 00034 BFUNC 00035 BFUNC 00036 BFUNC 00037 BFUNC 00038 BFUNC 00039 BFUNC 00040 BFUNC 00041 BFUNC 00042 BFUNC 00043 BFUNC 00044 BFUNC 00045 BFUNC 00046 BFUNC 00047 BFUNC 00048 BFUNC 00049 BFUNC 00050 </pre>

```

OVERLAY (AFMBOX,1,5)
PROGRAM NAWPMBX

C
C      THIS SECTION CONTROLS THE COMPUTATION OF BOX NORMAL WASH
C      VALUES AND VELOCITY POTENTIAL DIFFERENCES.  THE NORMAL WASH
C      VALUES MAY BE PRINTED, BUT OTHERWISE ARE NOT SAVED.  THE
C      VELOCITY POTENTIAL DIFFERENCES ARE PLACED ON SCRATCH FILE
C      IVPSC, TWO MATRICES PER MODE SHAPE
C
COMMON PKERNL (1640)
COMPLEX PKERNL
COMMON /CONTRL/ PREVEX,OMACH, TITLE(8), PRVGEOM,PRVMODE,DIHW,DIHT,
1      DEFAULT
LOGICAL PRVGEOM,PRVMODE,DIHW,DIHT,DEFAULT
COMMON /PROBLM/ XMACH,NMODES,NTSLOP,NKVALS,SMOOTH,NDEG,CRDFIT,
1      EXAIC,SUBDV,PLYWOOD
LOGICAL SMOOTH,CRDFIT,EXAIC,SUBDV,PLYWOOD
COMMON /GEOMTY/ COPLAN,NSUBDV,XSUBDV,NSUBD2,NSUBCN,NSURF,
1      B1,B1BETA,B1S,B1BTAS,WLAX,WLAZ,PSIW,
2      MXBW,MXBBW,MYBW,MYBBW,MXBSW,MYBSW,MYBBSW,
3      IXBW,XCENTR
LOGICAL COPLAN
COMMON /GEOM2/ TLAX,TLAZ,PSIT,MXBT,MYBT,MYBBT,MXBST,MYBST,
1      MYBBST,IXBT,IXBST,CAPL
COMMON /KERN/ ERR,MXSKRN,IPKERN,NPKRN,NSPATK,NROEA
COMMON /KVAL/ IKVAL,XKVAL(20),XKS(20)
COMMON /FILES/ NT5,NT6,INTAPE,INFSP,NPLAIC,NSFAIC,NOUTP,
1      IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC
COMMON /IOCONT/ OPLAIC,OSPAIC,WTGEOM,WTGNAF,WTSL,WTBL,PRBOX,
1      PRPAIC,PRSAIC,PRMODS,PRCOEF,PRDW,PRSW,PRVP,
2      PRBL,PRDCP,PRGNAF,PRGNAC,PRSL,PRLW,PRNW,PRCM
EQUIVALENCE (PRUW,PRDW)
LOGICAL OPLAIC,OSPAIC,WTGEOM,WTGNAF,WTSL,WTBL,PRBOX,PRPAIC,
1      PRSAIC,PRMODS,PRCOEF,PRDW,PRSW,PRVP,PRBL,PRSL,PRGNAF,
2      PRDCP,PRGNAC,PRUW,PRLW,PRNW,PRCM
COMMON /TAPEIO/ NFS,NMS,LS,NMR,ID(20),NID,ITYPE,LRS,LWS,M,N,
1      PARM(10),IRR
DIMENSION IPARM(10)
EQUIVALENCE (PARM,IPARM)
COMMON /MODES/ SYM,SYMT,MTYPEW,MTYPEP
COMMON /ARRAYS/ KBXCDW,LBXCDW,LBOXC,KBXCDT,LBXCDT,KJALPH,LJALPH,
1      KALPHA,KKERNL,LKERNL,KPNTRM,LPNTRM,KDEFSL,KELPHI,
2      LMODES,KPNTSD,LPNTSD,KSDW,LSDW,KPNTDW,LPNTDW,
3      KDW,LDW,KTVP,LTVF
COMMON /SAMPLW/ ISMPLW,ICHORD(10),IBOXF(10),IBOXL(10),ZLOC(10)
COMMON /PAICS/ NMAK,NTTK,NRWTN,XLTK,PAIC(4,50)
INTEGER PAIC
DIMENSION NK(4)
EQUIVALENCE (NMAK,NK(1))
COMMON /MUAICS/ YBAR,EL,MUAIC(2,50),NROWS,SURF,
1      YBARL,ELL,MUAICL(2,50),NROWSL,SURFL,PSIDIF
LOGICAL SURF,SURFL
COMMON /AICS/ XKVL, C(1640),W(1640),V(1640)
COMPLEX C, W, V
C      DELPHI(LMODES),TVP(LTVP),TEXLOC(LTVP)
COMMON /DELTAP/ DELPHI(1080),TVP(250),TEXLOC(250),FEXLOC(250),
1      IPNTRM(2,100),NPNTRS,IOVLAP
NAWPMBX 00002
NAWPMBX 00003
NAWPMBX 00004
NAWPMBX 00005
NAWPMBX 00006
NAWPMBX 00007
NAWPMBX 00008
NAWPMBX 00009
NAWPMBX 00010
NAWPMBX 00011
NAWPMBX 00012
CONTRL 00002
CONTRL 00003
CONTRL 00004
PROBLM 00002
PROBLM 00003
PROBLM 00004
GEOMTY 00002
GEOMTY 00003
GEOMTY 00004
GEOMTY 00005
GEOMTY 00006
GEOM2 00002
GEOM2 00003
KERN 00002
KVAL 00002
FILES 00002
FILES 00003
IOCONT 00002
IOCONT 00003
BCSFRB 00001
IOCONT 00005
IOCONT 00006
IOCONT 00007
BCSFRB 00002
TAPEIO 00002
TAPEIO 00003
TAPEIO 00004
TAPEIO 00005
MODCOM 00002
ARRAYS 00002
ARRAYS 00003
ARRAYS 00004
ARRAYS 00005
SAMPLW 00002
PAICS 00002
PAICS 00003
PAICS 00004
PAICS 00005
MUAICS 00002
MUAICS 00003
MUAICS 00004
AICS 00002
AICS 00003
DELTAP 00002
DELTAP 00003
DELTAP 00004

```

	COMPLEX	DELPHI,	TVP		DELTA	00005
C		DEFSL(2,LMODES)			DELTA	00006
	DIMENSION	DEFSL(2,1000)			DELTA	00007
	EQUIVALENCE	(DELPHI(81), DEFSL)			DELTA	00008
C		ARRAYS DELPHI AND DEFSL ARE			DELTA	00009
C		EQUIVALENCE TO GIVE A 2 ROW UN-OVERLAPPED SECTION			DELTA	00010
	DIMENSION	TVPX(500), XINITX(2)			FTNX1	00051
	COMPLEX	XINIT			FTNX1	00052
	EQUIVALENCE	(TVP,TVPX),(XINIT,XINITX)			FTNX1	00053
	COMMON /NWASHES/	IPNTDW(2,100),ENRUS(1275), ENRLS(1275),IOWLAPN			NWASHES	00002
	COMPLEX	ENRUS, ENRLS			NWASHES	00003
	COMMON /SNWASH/	IPNTSD(2,50), ENSUBD(2,600), IPNTIN,IPNTOT,IPNTLS			SNWASH	00002
C		IPNTSD(LPNTSD), ENSUBD(2*LSDW)			SNWASH	00003
	COMPLEX	ENSUBD			SNWASH	00004
C		IBOXW(LBXCOW,LBOXC), WHERE LBOXC = LSCHDS/20			BOXDES	00002
	COMMON /BOXDES/	IBOXW(150,8)			BOXDES	00003
C		IBOXW IS USED FOR BOTH WING AND TAIL BOX CODES			BOXDES	00004
C					NWVPMBX	00033
C		IPNTRM(2,NROWS),IPNTDW(2,NROWS),IPNTSD(2,NROWS)			NWVPMBX	00034
C					NWVPMBX	00035
	COMMON /LROT	/ LROT			NWVPMBX	00036
	COMMON /CHECKFR/	DPRCFR, GEOCFR, MODCFR, AICCFR, NMSCFR, SMCFR, GAFCFR			CHECKFR	00002
	LOGICAL	DPRCFR, GEOCFR, MODCFR, AICCFR, NMSCFR, SMCFR, GAFCFR			CHECKFR	00003
	LOGICAL	CHECKFR			NWVPMBX	00038
	EQUIVALENCE	(CHECKFR, NMSCFR)			NWVPMBX	00039
	DIMENSION	TITL(3)			NWVPMBX	00040
	DIMENSION	PARMW(10),IPARMW(10)			NWVPMBX	00041
	EQUIVALENCE	(PARMW,IPARMW)			NWVPMBX	00042
	LOGICAL	MXWRIT, RANDOU, MXRD, MXWRT			NWVPMBX	00043
	DATA	TVPX / 500* 60000000000200377777B /			FTNX1	00054
	DATA	MXWRIT,RANDOU, MXRD,MXWRT / .F.,.F.,.F.,.F. /			NWVPMBX	00044
	DATA	XINITX / 2* 37704000000000000000B /			FTNX1	00055
C					NWVPMBX	00047
C		PSIDIF = PSIT - PSI'W			NWVPMBX	00048
	IPNTLS = LPNTSD				NWVPMBX	00049
	LINDWS = LSDW				NWVPMBX	00050
	LROT = NSUBDV + NSUBCN				NWVPMBX	00051
	NTVP = (MYBSW + MYBST)				NWVPMBX	00052
	PARMW(2) = B1				NWVPMBX	00053
	PARMW(3) = XMACH				NWVPMBX	00054
	IPARMW(4) = NMOCES				NWVPMBX	00055
C					NWVPMBX	00056
C					NWVPMBX	00057
	REWIND IGEO6C				NWVPMBX	00058
	REWIND MOCESC				NWVPMBX	00059
	REWIND IVPSC				NWVPMBX	00060
C					NWVPMBX	00061
	IKVL = IKVAL(IKVAL)				NWVPMBX	00062
C					NWVPMBX	00063
C		HEAD GEOMETRIC INFORMATION			NWVPMBX	00064
C		WING BOX CODES			NWVPMBX	00065
	MXARRY = 9HBOX CODES				NWVPMBX	00066
	CALL RDINIT				NWVPMBX	00067
	ITYPE = 5H MIXED				NWVPMBX	00068
	K = LBXCOW				NWVPMBX	00069
	CALL READMX (IGEO6C, MXRD, .F., NFS,NMS,LS, NMR, K, NID,ID, ITYPE,				NWVPMBX	00070
					NWVPMBX	00071

1	LRS, IBOXW, M,N, PARM, IRR)	NWVPMBX 00072
	IF (IRR .NE. 0) GO TO 910	NWVPMBX 00073
	MXBBS = M	NWVPMBX 00074
	IF (INSURF .EQ. 1 .OR. COPLAN) GO TO 20	NWVPMBX 00075
C		NWVPMBX 00076
C	TAIL BOX CODES	NWVPMBX 00077
	MXARRY = 10HTAIL CODES	NWVPMBX 00078
	CALL RDINIT	NWVPMBX 00079
	ITYPE = SHMIXED	NWVPMBX 00080
	CALL READMX (IGEOSC, MXRD, .F., NFS,NMS,LS, NMR, K, NID,ID, ITYPE,	NWVPMBX 00081
1	LRS, IBOXW(MXBBS+1,1), M,N, PARM, IRR)	NWVPMBX 00082
	IF (IRR .NE. 0) GO TO 910	NWVPMBX 00083
	ISUBT = MXBBS + 2 - IXBST	NWVPMBX 00084
C	= SUBSCRIPT FOR IBOXW TO GET TAIL CODES	NWVPMBX 00085
C		NWVPMBX 00086
20	CONTINUE	NWVPMBX 00087
	MXARRY 10H FEXLOC	NWVPMBX 00088
	CALL RLINIT	NWVPMBX 00089
	ITYPE = SHMIXED	NWVPMBX 00090
	CALL READMX(IGEOSC, MXRD,.F., NFS,NMS,LS, NMR, 1, NID,ID, ITYPE,	NWVPMBX 00091
1	LRS, FEXLOC, M,N, PARM, IRR)	NWVPMBX 00092
	IF (IRR .NE. 0) GO TO 910	NWVPMBX 00093
C		NWVPMBX 00094
	MXARRY =10H TEXLOC	NWVPMBX 00095
	CALL RDINIT	NWVPMBX 00096
	ITYPE = SHMIXED	NWVPMBX 00097
	CALL READMX(IGEOSC, MXRD,.F., NFS,NMS,LS, NMR, 1, NID,ID, ITYPE,	NWVPMBX 00098
1	LRS, TEXLOC, M,N, PARM, IRR)	NWVPMBX 00099
	IF (IRR .NE. 0) GO TO 910	NWVPMBX 00100
	IF(M .NE. 1) GO TO 930	NWVPMBX 00101
C		NWVPMBX 00102
	IF (PSIW .NE. 0 .AND. DIHW) GO TO 30	NWVPMBX 00103
	IF (ISHPLW .NE. 0) GO TO 30	NWVPMBX 00104
	IF (INSURF .EQ. 1) GO TO 40	NWVPMBX 00105
	IF (CARL .NE. 0) GO TO 30	NWVPMBX 00106
	IF (PSIDIF .NE. 0) GO TO 30	NWVPMBX 00107
	IF (PSIT .NE. 0 .AND. DIHT) GO TO 30	NWVPMBX 00108
	GO TO 40	NWVPMBX 00109
C		NWVPMBX 00110
C	READ THE ARRAY OF AIC TABLE .F CONTENTS	NWVPMBX 00111
30	CONTINUE	NWVPMBX 00112
	MXARRY = 9HSPAT. TOC	NWVPMBX 00113
	CALL RDINIT	NWVPMBX 00114
	NMS = 2	NWVPMBX 00115
	K = 4	NWVPMBX 00116
	CALL READMX (IGEOSC, MXRD, .F., NFS,NMS,LS, NMR, K, NID,ID, ITYPE,	NWVPMBX 00117
1	LRS, PAIC, M,N, PARM, IRR)	NWVPMBX 00118
	IF (IRR .NE. 0) GO TO 910	NWVPMBX 00119
	DO 35 I = 1,4	NWVPMBX 00120
	NK(I) = IPARM(I+2)	NWVPMBX 00121
35	CONTINUE	NWVPMBX 00122
40	CONTINUE	NWVPMBX 00123
	REWRITE IGEOSC	NWVPMBX 00124
C		NWVPMBX 00125
C	GET POINTER ARRAY FOR MODES	NWVPMBX 00126
	CALL RDINIT	NWVPMBX 00127
	ITYPE = SHMIXED	NWVPMBX 00128

CALL READMX(MODESC, MXRD, .F., NFS, NMS, LS, NMR, 2, NID, ID, ITYPE,	NWVPMBX 00129
1 LRS, IPNTRM, M, N, PARM, IRR)	NWVPMBX 00130
IF (IRR .NE. 0) GO TO 912	NWVPMBX 00131
IF (M .NE. 2) GO TO 931	NWVPMBX 00132
NPNTRS = N	NWVPMBX 00133
IOVLAP = IPARM(3)	NWVPMBX 00134
C IOVLAP = NUMBER OF ROWS TO ALLOW FOR TAIL OVERLAP (TAIL ONLY)	NWVPMBX 00135
C NPNTRS = TOTAL NUMBER OF ROWS ON BOTH SURFACES, + 1 .	NWVPMBX 00136
C (INCLUDES OVERLAP IF SPATIAL)	NWVPMBX 00137
C	NWVPMBX 00138
C SET UP POINTER ARRAY FOR UNSUBDIVIDED DOWNWASHES	NWVPMBX 00139
IP = 1	NWVPMBX 00140
MYBB = MYBBW	NWVPMBX 00141
IF (COPLAN) MYBB = MAX(MYBB, MYBBT)	NWVPMBX 00142
MYBBS = MYBB + NSUBDV	NWVPMBX 00143
MXBB = MXBBW	NWVPMBX 00144
IF (COPLAN) MXBB = MXBT	NWVPMBX 00145
CALL POINTR(1, MXBB, MYBB, .F., .T., IBOXW, LBXCOW, LPNTDW, 1,	NWVPMBX 00146
1 , IP, IPNTDW)	NWVPMBX 00147
MXB = MXBW	NWVPMBX 00148
IF (COPLAN) MXB = MXBT	NWVPMBX 00149
MYB = MYBW	NWVPMBX 00150
IF (COPLAN) MYB = MYBT	NWVPMBX 00151
IOVLAPN = 0	NWVPMBX 00152
IF (NSURF .EQ. 1 .OR. COPLAN) GO TO 50	NWVPMBX 00153
MXB = MXBT	NWVPMBX 00154
IXBUT = (IXBT - IXBW) / NSUBDV + 1	NWVPMBX 00155
IP = MXBBW + 1	NWVPMBX 00156
IPNT = IPNTDW(1, IP)	NWVPMBX 00157
CALL POINTR(IXBUT, MXBT - IXBUT + 1, MYBBT, .F., .T., IROWW(ISUBT, 1),	NWVPMBX 00158
1 LBXCOW, LPNTDW, IPNT, IP, IPNTDW)	NWVPMBX 00159
IF (MXBBW .GE. IXBUT) IOVLAPN = MXBBW - IXBUT + 1	NWVPMBX 00160
50 CONTINUE	NWVPMBX 00161
C	NWVPMBX 00162
C LOOP ON MODE SHAPES	NWVPMBX 00163
DO 500 IMODE = 1, NMODES	NWVPMBX 00164
C	NWVPMBX 00165
C ZERO OUT THE DOWNWASH AND VELOCITY POTENTIAL ARRAYS	NWVPMBX 00166
LIM = IPNTDW(1, IP) - 1	NWVPMBX 00167
DO 80 I = 1, LIM	NWVPMBX 00168
ENRUS(I) = XINIT	NWVPMBX 00169
ENRIS(I) = XINIT	NWVPMBX 00170
80 CONTINUE	NWVPMBX 00171
LIM = IPNTRM(1, NPNTRS) - 1	NWVPMBX 00172
DO 85 I = 1, LIM	NWVPMBX 00173
DELPHI(I) = (0., 0.)	NWVPMBX 00174
85 CONTINUE	NWVPMBX 00175
C	NWVPMBX 00176
C	NWVPMBX 00177
C READ IN MODE SHAPE	NWVPMBX 00178
CALL RDINIT	NWVPMBX 00179
MXARRY = IOHMODE SHAPE	NWVPMBX 00180
ITYPE = 4HREAL	NWVPMBX 00181
CALL READMX(MODESC, MXRD, .F., NFS, NMS, LS, NMR, 2, NID, ID, ITYPE,	NWVPMBX 00182
1 LRS, DEFSL, M, N, PARM, IRR)	NWVPMBX 00183
IF (IRR .NE. 0) GO TO 912	NWVPMBX 00184
IF (M .NE. 2) GO TO 931	NWVPMBX 00185

C		NWVPMBX 00186
C	COMPUTE DOWNWASHES AND VELOCITY POTENTIALS FOR ONE MODE	NWVPMBX 00187
	CALL VELPOT (IBOXW,LBXCDW, PKERNL(IPKERN), PKERNL, .T., DIHW)	NWVPMBX 00188
C	OPTIONAL PRINT OF NORMAL WASHES	NWVPMBX 00189
	IF (.NOT. (CHECKPR .OR. PRNW)) GO TO 90	NWVPMBX 00190
	IF (NSUBDV .EQ. 1) GO TO 87	NWVPMBX 00191
	IF (.NOT. CHECKPR) GO TO 87	NWVPMBX 00192
	TITL(1) = 10HEN SUBDIVI	NWVPMBX 00193
	TITL(2) = 10HDED, UPPER	NWVPMBX 00194
	TITL(3) = 10H, PARTIAL	NWVPMBX 00195
	IF (IPNTIN .LT. IPNTOT) GO TO 86	NWVPMBX 00196
	CALL PRINTR(TITL,IMODE,ENSUBD,2, 1, IPNTIN-1, MYBBS,IPNTSD)	NWVPMBX 00197
	GO TO 87	NWVPMBX 00198
86	CONTINUE	NWVPMBX 00199
	CALL PRINTR(TITL,IMODE,ENSUBD,2,IPNTOT,IPNTLS-1,MYBBS,IPNTSD)	NWVPMBX 00200
	CALL PRINTR(TITL,IMODE,ENSUBD,2, 1, IPNTIN-1, MYBBS,IPNTSD)	NWVPMBX 00201
87	CONTINUE	NWVPMBX 00202
	TITL(1) = 10HWING UPPER	NWVPMBX 00203
	TITL(2) = 10H SURFACE N	NWVPMBX 00204
	TITL(3) = 10HORMAL WASH	NWVPMBX 00205
	IF (COPLAN) TITL(1) = 10H WING/TA	NWVPMBX 00206
	IF (COPLAN) TITL(2) = 10HIL UPPER N	NWVPMBX 00207
	CALL PRINTR(TITL,IMODE, ENRUS,1, 1,MYBB,MYBB, IPNTDW)	NWVPMBX 00208
	IF (.N. COPLAN) TITL(1) = 10HWING LOWER	NWVPMBX 00209
	IF (COPLAN) TITL(2) = 10HIL LOWER N	NWVPMBX 00210
	CALL PRINTR(TITL,IMODE, ENRUS,1, 1,MYBB,MYBB, IPNTDW)	NWVPMBX 00211
C		NWVPMBX 00212
90	CONTINUE	NWVPMBX 00213
	IF (NSURF .EQ. 1 .OR. COPLAN) GO TO 140	NWVPMBX 00214
C		NWVPMBX 00215
C	DETERMINE WHICH WING SURFACE CONTRIBUTES TO THE TAIL.	NWVPMBX 00216
	IF (CAPL .GT. 0) GO TO 130	NWVPMBX 00217
	IF (CAPL .EQ. 0 .AND. PSIDIF .GT. 0) GO TO 130	NWVPMBX 00218
C	THE LOWER WING SURFACE CONTRIBUTES TO THE TAIL	NWVPMBX 00219
	LIM = IPNTDW(1,MYBBW+1) - 1	NWVPMBX 00220
	DO 120 I = 1,LIM	NWVPMBX 00221
	ENRUS(I) = ENRUS(I)	NWVPMBX 00222
120	CONTINUE	NWVPMBX 00223
C		NWVPMBX 00224
C	COMPUTE THE TAIL NORMAL WASHES AND VELOCITY POTENTIALS	NWVPMBX 00225
130	CONTINUE	NWVPMBX 00226
	CALL VELPOT (IBOXW(ISURT,1),LBXCDW, PKERNL(IPKERN), PKERNL,	NWVPMBX 00227
	1, .F., DIHT)	NWVPMBX 00228
C	OPTIONAL PRINT OF NORMAL WASHES	NWVPMBX 00229
	IF (.NOT. (CHECKPR .OR. PRNW)) GO TO 135	NWVPMBX 00230
	IF (NSUBDV .EQ. 1 .OR. .NOT. CHECKPR) GO TO 133	NWVPMBX 00231
	TITL(1) = 10HEN SUBDIVI	NWVPMBX 00232
	TITL(2) = 10HDED, UPPER	NWVPMBX 00233
	TITL(3) = 10H, PARTIAL	NWVPMBX 00234
	IF (IPNTIN .LT. IPNTOT) GO TO 131	NWVPMBX 00235
	CALL PRINTR(TITL,IMODE,ENSUBD,2, 1, IPNTIN-1, MYBBS,IPNTSD)	NWVPMBX 00236
	GO TO 133	NWVPMBX 00237
131	CONTINUE	NWVPMBX 00238
	CALL PRINTR(TITL,IMODE,ENSUBD,2,IPNTOT,IPNTLS-1,MYBBS,IPNTSD)	NWVPMBX 00239
	CALL PRINTR(TITL,IMODE,ENSUBD,2, 1, IPNTIN-1, MYBBS,IPNTSD)	NWVPMBX 00240
133	CONTINUE	NWVPMBX 00241
	TITL(1) = 10HTAIL UPPER	NWVPMBX 00242

CALL PRINTR(TITL,IMCDE, ENR/LS,1, IXBUT,MXBT,MYBBT,	NWVPMBX 00243
1 IPNTRM(1,IOVLAPN + 1))	NWVPMBX 00244
TITL(1) = 10HTAIL LOWER	NWVPMBX 00245
CALL PRINTR(TITL,IMCDE, ENR/LS,1, IXBUT,MXBT,MYBBT,	NWVPMBX 00246
1 IPNTRM(1,IOVLAPN + 1))	NWVPMBX 00247
135 CONTINUE	NWVPMBX 00248
C WRITE THE RESULTS ON SCRATCH FILES	NWVPMBX 00249
140 CONTINUE	NWVPMBX 00250
PARM(1) = MXVL	NWVPMBX 00251
DO 210 I = 1,4	NWVPMBX 00252
210 PARM(I) = PARM(I)	NWVPMBX 00253
C	NWVPMBX 00254
C XX WRITE DELTA RHIS, TEMPORARILY AS A REAL MATRIX, WRTEP XX	NWVPMBX 00255
M = 2	NWVPMBX 00256
ITYPE = 4HREAL	NWVPMBX 00257
M = IPNTRM(1,MXB+IOVLAP+1) - 1	NWVPMBX 00258
CALL WRTEMX(IVPSC, .F., .F., NFS,NMS,LS, NMR,LWS, 2, ID, DELPHI,	NWVPMBX 00259
1 ITYPE, M,N,FARM, IRR)	NWVPMBX 00260
IF (IRR .NE. 0) GO TO 920	NWVPMBX 00261
C	NWVPMBX 00262
M = 2	NWVPMBX 00263
M = NTVP	NWVPMBX 00264
CALL WRTEMX(IVPSC, .F., .F., NFS,NMS,LS, NMR, LWS, 2, ID, TVP,	NWVPMBX 00265
1 ITYPE, M,N,FARM, IRR)	NWVPMBX 00266
IF (IRR .NE. 0) GO TO 920	NWVPMBX 00267
C	NWVPMBX 00268
IF (.NOT. PRVP) GO TO 230	NWVPMBX 00269
TITL(1) = 0H WING	NWVPMBX 00270
TITL(2) = 10HVELOCITY P	NWVPMBX 00271
TITL(3) = 10HPOTENTIALS	NWVPMBX 00272
M = MXBW	NWVPMBX 00273
IF (.N. COPLAN) GO TO 220	NWVPMBX 00274
TITL(1) = 10HWING/TAIL	NWVPMBX 00275
M = MPNTRM - 1	NWVPMBX 00276
220 CALL PRINTR(TITL, IMCDE, DELPHI,1, 1,M, MYB, IPNTRM)	NWVPMBX 00277
IF (NSURF .EQ. 1 .OR. COPLAN) GO TO 230	NWVPMBX 00278
TITL(1) = 0H TAIL	NWVPMBX 00279
CALL PRINTR(TITL, IMCDE, DELPHI, 1, IXBUT,MXBT, MYBT,	NWVPMBX 00280
1 IPNTRM(1,IOVLAP+1))	NWVPMBX 00281
230 CONTINUE	NWVPMBX 00282
C ARE SAMPLE WASHES DESIRED -	NWVPMBX 00283
IF (NSURF .EQ. 2 .OR. ISMPLW .EQ. 0) GO TO 500	NWVPMBX 00284
C YES. IS SAMPLE WASH PRINTOUT DESIRED	NWVPMBX 00285
IF (.NOT. (PRDW .OR. PRSW)) GO TO 500	NWVPMBX 00286
C LOOP ON CHORDS FOR WHICH SAMPLE-WASH IS DESIRED	NWVPMBX 00287
DO 300 JCHRD = 1,ISMPLW	NWVPMBX 00288
JT = 10HJCHRD(JCHRD)	NWVPMBX 00289
IFIRST = 10HJCHRD(JCHRD)	NWVPMBX 00290
ILAST = 10HJCHRD(JCHRD)	NWVPMBX 00291
CALL SMPLW(IBOKW,LBXCDW,JCHRD,JT,IFIRST,ILAST)	NWVPMBX 00292
300 CONTINUE	NWVPMBX 00293
C	NWVPMBX 00294
300 CONTINUE	NWVPMBX 00295
C END OF LOOP ON MODE SHAPES, FROM STATEMENT 50*	NWVPMBX 00296
C	NWVPMBX 00297
RETURN	NWVPMBX 00298
	NWVPMBX 00299

C		NWPMBX 00300
C	DIAGNOSTICS - ALL CALL FLUSH	NWPMBX 00301
C		NWPMBX 00302
C	READING FROM SCRATCH FILE	NWPMBX 00303
	910 CONTINUE	NWPMBX 00304
	WRITE (NT6,9100) IGCOSC	NWPMBX 00305
	GO TO 950	NWPMBX 00306
	912 CONTINUE	NWPMBX 00307
	WRITE (NT6,9120) MODESC	NWPMBX 00308
	GO TO 950	NWPMBX 00309
	920 CONTINUE	NWPMBX 00310
C	WRITING ON SCRATCH FILE	NWPMBX 00311
	WRITE (NT6,9200) IVPSC	NWPMBX 00312
	GO TO 952	NWPMBX 00313
C	INCORRECT DIMENSIONS READ	NWPMBX 00314
	930 CONTINUE	NWPMBX 00315
	I = 1	NWPMBX 00316
	GO TO 932	NWPMBX 00317
	931 I = 2	NWPMBX 00318
	932 WRITE (NT6,9300) I	NWPMBX 00319
	IF (MXRD) GO TO 960	NWPMBX 00320
	GO TO 962	NWPMBX 00321
C	ERROR DETECTED READING A MATRIX	NWPMBX 00322
	950 CONTINUE	NWPMBX 00323
	WRITE (NT6,9500) IRR	NWPMBX 00324
	IF (MXRD) GO TO 960	NWPMBX 00325
	GO TO 962	NWPMBX 00326
C	ERROR DETECTED WRITING A MATRIX	NWPMBX 00327
	952 CONTINUE	NWPMBX 00328
	WRITE (NT6,9520) IRR	NWPMBX 00329
	IF (MXWR) GO TO 960	NWPMBX 00330
	WRITE (NT6,9630) MXARRY	NWPMBX 00331
	GO TO 962	NWPMBX 00332
C	MATRIX DESCRIPTION	NWPMBX 00333
	960 CONTINUE	NWPMBX 00334
	WRITE (NT6,9600) (ID(I),I=1,10),(ID(I),I=1,10)	NWPMBX 00335
	WRITE (NT6,9622) PARM,PARM	NWPMBX 00336
	WRITE (NT6,9614) NMR,NMR,LRS,LWS	NWPMBX 00337
	GO TO 964	NWPMBX 00338
	962 WRITE (NT6,9620) ID(1),ID(2)	NWPMBX 00339
	WRITE (NT6,9622) PARM,PARM	NWPMBX 00340
	WRITE (NT6,9624) NFS,NFS	NWPMBX 00341
	964 WRITE (NT6,9640) ITYPE,M,M	NWPMBX 00342
	WRITE (NT6,9630) MXARRY	NWPMBX 00343
	GO TO 990	NWPMBX 00344
C		NWPMBX 00345
	990 CONTINUE	NWPMBX 00346
	WRITE (NT6,9900)	NWPMBX 00347
C		NWPMBX 00348
	CALL FLUSH(1)	NWPMBX 00349
C		NWPMBX 00350
C	DIAGNOSTIC FORMATS	NWPMBX 00351
	9100 FORMAT(47H0*** ERROR WHILE READING GEOMETRY SCRATCH FILE ,A10,	NWPMBX 00352
	1 4H ***)	NWPMBX 00353
	9120 FORMAT(44H0*** ERROR WHILE READING MODES SCRATCH FILE ,A10,	NWPMBX 00354
	1 4H ***)	NWPMBX 00355
	9200 FORMAT(51H0*** ERROR WHILE WRITING VELOCITY POTENTIAL SCRATCH	NWPMBX 00356

1	6H FILE ,A10, 4H ***)	NWPMBX 00357
9300	FORMAT(10H, 48H*** MATRIX READ ERROR. THE M DIMENSION SHOULD	NWPMBX 00358
1	4H BE ,I2, 4H ***)	NWPMBX 00359
9500	FORMAT(16H0 *** ERROR CODE ,I5, 28H WHILE READING THE FOLLOWING	NWPMBX 00360
1	11H MATRIX ***)	NWPMBX 00361
9520	FORMAT(16H0 *** ERROR CODE ,I5, 28H WHILE WRITING THE FOLLOWING	NWPMBX 00362
1	11H MATRIX ***)	NWPMBX 00363
9600	FORMAT(5X,MATRIX ID = *,10A10 / (20X,10A10))	NWPMBX 00364
9614	FORMAT(5X,22HMATRIX INDEX (NAME) = ,I5,2H (A10,1H) /	NWPMBX 00365
1	5X,33HLEVEL NUMBER READ (OR WRITTEN) = Q2,2H (,Q2,1H))	FTNX1 00056
9620	FORMAT(5X,MATRIX ID = *, A10, 110)	NWPMBX 00367
9622	FORMAT(5X,11HPARAMETERS, 10E11.3 /10X, 9H(INTEGER), 17,9I11)	NWPMBX 00368
9624	FORMAT(5X,15HFILE SPACING = ,I3, 19H, MATRIX SPACING = ,I3)	NWPMBX 00369
9630	FORMAT(5X,A10,21H ARRAY WAS BEING USED)	NWPMBX 00370
9640	FORMAT(5X,MATRIX TYPE - *,A10, *, DIMENSIONED (*I4,* X*,I4,*)*)	NWPMBX 00371
9900	FORMAT(54H0*** ERROR OCCURRED DURING VELOCITY POTENTIAL CALCULAT	NWPMBX 00372
1	8HIONS ***)	NWPMBX 00373
C		NWPMBX 00374
	END	NWPMBX 00375

	SUBROUTINE VELPOT (IBOX, LBXCD, PKERNL, SKERNL, WING, DIHS)	VELPOT 00002
C		VELPOT 00003
C	VELPOT IS CALLED ONCE FOR EACH MODE, TO COMPUTE NORMAL WASHES	VELPOT 00004
C	AND VELOCITY POTENTIALS FOR THAT MODE.	VELPOT 00005
C		VELPOT 00006
C	IBOX = BOX CODES FOR THE SURFACE	VELPOT 00007
C	LBXCD = BOX CODE ARRAY ROW DIMENSION	VELPOT 00008
C	PKERNL = ARRAY CONTAINING C(NU,MU,0)	VELPOT 00009
C	SKERNL = ARRAY CONTAINING SUBDIVIDED C(NU,MU,0)	VELPOT 00010
C	WING = LOGICAL, .T. FOR WING OR COPLAAR, .F. FOR TAIL	VELPOT 00011
C	DIHS = LOGICAL, .T. TO INCLUDE DIHEDRAL EFFECTS, .F. TO	VELPOT 00012
C	IGNORE, FOR WING/WING (TAIL/TAIL)	VELPOT 00013
C		VELPOT 00014
	DIMENSION IBOX(LBXCD,1), ICODE(5)	VELPOT 00015
	COMPLEX PKERNL(1), SKERNL(1)	VELPOT 00016
	LOGICAL WING, DIHS	VELPOT 00017
C		VELPOT 00018
C	OUTPUTS -	VELPOT 00019
C	DELPHI = DELTA PHI (VELOCITY POTENTIAL) ARRAY	VELPOT 00020
C		VELPOT 00021
C	COMMON PARAMETERS USED	VELPOT 00022
C	NSUBDV = NUMBER OF SUBDIVISIONS	VELPOT 00023
C	B1 = BOX LENGTH	VELPOT 00024
C	B1S = SUBDIVIDED BOX LENGTH	VELPOT 00025
C		VELPOT 00026
	COMMON /FILES / NT5,NT6,INTAPE,INFSI, MPLAIC, NSPAIC, NOUTP,	FILES 00002
1	IOUFSP, MODESC, IVPSC, IGEOSC, IWTFS, IAICSC	FILES 00003
	COMMON /ARRAYS/ KBXCDW, LBXCDW, IBOXC, KBXCDT, LBXCDT, KJALPH, LJALPH,	ARRAYS 00002
1	KALPHA, KKERNL, LKERNL, KPNTRM, LPNTRM, KDEFSL, KELPHI,	ARRAYS 00003
2	LMODES, KPNTSD, LPNTSD, KSDW, LSDW, KPNTDW, LPNTDW,	ARRAYS 00004
3	KDW, LDW, KTVP, LTVP	ARRAYS 00005
	COMMON /GEOMTY/ COPLAN, NSUBDV, XSUBDV, NSUBD2, NSUBCN, NSURF,	GEOMTY 00002
1	B1, B1BETA, B1S, B1BTAS, WLAX, WLAZ, PSIW,	GEOMTY 00003
2	MYBW, MYBBW, MYBW, MYBBW, MYBSW, MYBSW, MYBBSW,	GEOMTY 00004
3	IXBW, YCENTR	GEOMTY 00005
	LOGICAL COPLAN	GEOMTY 00006
	COMMON /GEOM2 / TLAX, TLAZ, PSIT, MXBT, MYBT, MYBBT, MXBST, MYBST,	GEOM2 00002
1	MYBBST, IXBT, IXBST, CAFL	GEOM2 00003
	COMMON / MODES/ SYM, SYMT, MTFPEW, MTFPEF	MODECOM 00002
C		VELPOT 00032
	COMMON /MUAICS/ YBAR, EL, MUAIC(2,50), NROHS, SURF,	MUAICS 00002
1	YBARL, ELL, MUAICL(2,50), NROWL, SURFL, PSIDIF	MUAICS 00003
	LOGICAL SURF, SURFL	MUAICS 00004
	COMMON /AICS / XKVL, C(1640), W(1640), V(1640)	AICS 00002
	COMPLEX C, W, V	AICS 00003
C	DELPHI(LMODES), TVP(LTVP), TEXLOC(LTVP)	DELTAP 00002
	COMMON /DELTAP/ DELPHI(1000), TVP(250), TEXLOC(250), FEXLOC(250),	DELTAP 00003
1	IPNTRM(2,100), NPNTRS, IOMLAP	DELTAP 00004
	COMPLEX DELPHI, TVP	DELTAP 00005
C	DEFSL(2, LMODES)	DELTAP 00006
	DIMENSION DEFSL(2,1000)	DELTAP 00007
	EQUIVALENCE (DELPHI(81), DEFSL)	DELTAP 00008
C	ARRAYS DELPHI AND DEFSL ARE	DELTAP 00009
C	EQUIVALENCED TO GIVE A 2 ROW UN-OVERLAPPED SECTION	DELTAP 00010
	COMMON /NMASHES/ IPNTDW(2,100), ENRUS(1275), ENRLS(1275), IOMLAPN	NMASHES 00002
	COMPLEX ENRUS, ENRLS	NMASHES 00003
	COMMON /SMASH/ IPNTSD(2,50), ENSUBD(2,600), IPNTIN, IPNTOT, IPNTLS SMASH	00002

C	IPNTSD(L,PNTSD), ENSUBD(2*LSBW)	SMASH	00003
	COMPLEX ENSUBD	SMASH	00004
C	IBOXW(LBXCW,LBOXC), WHERE LBOXC = LSCHDS/20	BXCDES	00002
	COMMON /BXCDES/ IBOXW(150,8)	BXCDES	00003
C	IBOXW IS USED FOR BOTH WING AND TAIL BOX CODES	BXCDES	00004
	COMMON /CHECKPR/ DPPCPR, GEOCPR, MODCPR, AICCPR, NMSCPR, SMCPR, GACPR	CHECKPR	00002
	LOGICAL DPPCPR, GEOCPR, MODCPR, AICCPR, NMSCPR, SMCPR, GACPR	CHECKPR	00003
	EQUIVALENCE (CHECKPR, NMSCPR)	VELPOT	00040
	LOGICAL CHECKPR	VELPOT	00041
C		VELPOT	00042
	COMPLEX AZERO, DELPH, DELPHA, DELPHB, DELPHC, B, SDELPH, COEF	VELPOT	00043
	COMPLEX ENRULU(50), ENRLL(50), ENRURW(50), ENRULW(50), DRHIL(50)	VELPOT	00044
	COMPLEX ENDF, ENSUM, ENSRUS, ENSRLS	VELPOT	00045
	COMPLEX ENSBD(2)	VELPOT	00046
	EQUIVALENCE (ENSRUS, ENSBD(1)), (ENSRLS, ENSBD(2))	VELPOT	00047
C		VELPOT	00048
	LOGICAL CROW, FROW, LROW, CBOX, FBOX, LBOX, SUBOFF	VELPOT	00049
	LOGICAL FULLBX(50)	VELPOT	00050
	INTEGER WW, TT, RWT, LWT	VELPOT	00051
	COMPLEX XINIT	FTNXL	00057
	DIMENSION XINITX(2)	FTNXL	00058
	EQUIVALENCE (XINIT, XINITX)	FTNXL	00059
	DATA WW, TT, RWT, LWT /1,2,3,4/	VELPOT	00052
C	THESE VALUES MAY BE MODIFIED BY ACTUAL PAIC-- ARRAYS READ IN	VELPOT	00053
	DATA XINITX / 2* 37704000000000000000B /	FTNXL	00060
C		VELPOT	00056
C	SET CONSTANTS	VELPOT	00057
C	IXB = SUBDIVIDED SUBSCRIPT OF FIRST PLANFORM CONTROL POINT	VELPOT	00058
C	IXBS = SUBSCRIPT OF FIRST PLANFORM SUBDIVIDED BOX	VELPOT	00059
C	IXBU = UNSUBDIVIDED SUBSCRIPT OF FIRST PLANFORM CONTROL PT.	VELPOT	00060
C	MYBB = NUMBER OF UNSUBDIVIDED CHORDS TO CONSIDER, INCLUDING	VELPOT	00061
C	DIAPHRAGM	VELPOT	00062
C	MYBBS = NUMBER OF SUBDIVIDED CHORDS TO CONSIDER, INCL. DIAPH.	VELPOT	00063
C	MXB = LAST UNSUBDIVIDED ROW TO CONSIDER	VELPOT	00064
C	MYBS = LAST SUBDIVIDED ROW TO CONSIDER (TO LAST CONTROL PNT)	VELPOT	00065
	IF (WING) GO TO 80	VELPOT	00066
	IOVLP = IOVLP	VELPOT	00067
	IOVLPN = IOVLPN	VELPOT	00068
	PSIS = PSIT	VELPOT	00069
	IXB = IXBT	VELPOT	00070
	IXBS = IXBST	VELPOT	00071
	IXBU = (IXBT - IXBW)/NSUBDV + 1	VELPOT	00072
	MYBB = MYBBT	VELPOT	00073
	MYBBS = MYBBST	VELPOT	00074
	SYMTY = SYMT	VELPOT	00075
	GO TO 90	VELPOT	00076
90	CONTINUE	VELPOT	00077
	IOVLP = 0	VELPOT	00078
	IOVLPN = 0	VELPOT	00079
	PSIS = PSIW	VELPOT	00080
	IXB = IXBW	VELPOT	00081
	IXBS = 1	VELPOT	00082
	IXBU = 1	VELPOT	00083
	SYMTY = SYM	VELPOT	00084
	IF (COPLAN .AND. NSURF .EQ. 2) GO TO 85	VELPOT	00085
	MYBB = MYBBW	VELPOT	00086
	MYBBS = MYBBSW	VELPOT	00087

MXB = MXBBW	VEL POT 00088
MXBS = (MXB-1)*NSUBDV + IXBW	VEL POT 00089
MXBS = MAXD(MXBS, MXBSW)	VEL POT 00090
GO TO 100	VEL POT 00091
85 CONTINUE	VEL POT 00092
MYBB = MAXD(MYBBW, MYBBT)	VEL POT 00093
MYBBS = MAXD(MYBBSW, MYBBST)	VEL POT 00094
90 CONTINUE	VEL POT 00095
MXB = MXBT	VEL POT 00096
MXBS = MXBST	VEL POT 00097
C	VEL POT 00098
100 CONTINUE	VEL POT 00099
PSIS2 = 2*PSIS	VEL POT 00100
PSISUM = PSIW + PSIT	VEL POT 00101
CPSIS2 = COS(PSIS2)	VEL POT 00102
SPSIS2 = SIN(PSIS2)	VEL POT 00103
CPSISM = COS(PSISUM)	VEL POT 00104
SPSISM = SIN(PSISUM)	VEL POT 00105
IRFB = IXB - NSUBD2	VEL POT 00106
C = ROW OF FIRST SUBDIVIDED BOX IN THE FIRST ROW OF THE	VEL POT 00107
C UNSUBDIVIDED BOXES	VEL POT 00108
MYBBSX = ((MYBBS+NSUBD2)/NSUBDV)*NSUBDV	VEL POT 00109
IF (NSUBDV .EQ. 1) GO TO 110	VEL POT 00110
SUBOFF = .F.	VEL POT 00111
MYBBP1 = MYBB+1	VEL POT 00112
DO 105 I = 1, MYBBP1	VEL POT 00113
ENRULU(I) = (0., 0.)	VEL POT 00114
ENRLLI(I) = (0., 0.)	VEL POT 00115
ENRURW(I) = (0., 0.)	VEL POT 00116
ENRULW(I) = (0., 0.)	VEL POT 00117
FULLBX(I) = .T.	VEL POT 00118
105 CONTINUE	VEL POT 00119
GO TO 120	VEL POT 00120
110 SUBOFF = .T.	VEL POT 00121
CROW = .T.	VEL POT 00122
UROW = .T.	VEL POT 00123
FBOX = .T.	VEL POT 00124
UBOX = .T.	VEL POT 00125
LBOX = .T.	VEL POT 00126
C	VEL POT 00127
C LOOP ON ALL (SUBDIVIDED) ROWS OF THE SURFACE	VEL POT 00128
C	VEL POT 00129
120 CONTINUE	VEL POT 00130
FLIROW = FLOAT(IXBS) - 1.0	VEL POT 00131
DO 1300 IROW = IXBS, MXBS	VEL POT 00132
FLIROW = FLIROW + 1.0	VEL POT 00133
C	VEL POT 00134
C SET FLAGS FOR FIRST, CENTER AND LAST SUBDIVIDED ROW IN UN-	VEL POT 00135
C SUBDIVIDED ROW	VEL POT 00136
IF (SUBOFF) GO TO 270	VEL POT 00137
FROM = .F.	VEL POT 00138
IF (IROW - IXB) 230, 220, 210	VEL POT 00139
210 IF (MOD(IROW-IXB, NSUBDV) .NE. 0) GO TO 240	VEL POT 00140
220 CROW = .T.	VEL POT 00141
GO TO 250	VEL POT 00142
230 CROW = .F.	VEL POT 00143
IF (IROW .EQ. IXBS .OR. IROW .EQ. IRFB) FROM = .T.	VEL POT 00144

GO TO 250	VEL POT 00145
240 CROW = .F.	VEL POT 00146
I = MOD((IROW*NSUBCN - IXB, NSUBDV)	VEL POT 00147
IF (I - 1) 260,245,250	VEL POT 00148
245 FROM = .T.	VEL POT 00149
250 LROW = .F.	VEL POT 00150
IF (IROW .NE. MXBS) GO TO 270	VEL POT 00151
260 LROW = .T.	VEL POT 00152
C	VEL POT 00153
C GET THE SUBDIVIDED ROW NUMBER FOR THE CENTER OF THE ASSOCIATED	VEL POT 00154
C UNSUBDIVIDED BOX, ICENT	VEL POT 00155
270 CONTINUE	VEL POT 00156
IF (CROW) GO TO 340	VEL POT 00157
IF (IROW - IRFB) 310,320,330	VEL POT 00158
C NO FULL ROW	VEL POT 00159
310 ICENT = 0	VEL POT 00160
IUCENT = 0	VEL POT 00161
NDCEN = 0	VEL POT 00162
CBOX = .F.	VEL POT 00163
LBOX = .F.	VEL POT 00164
GO TO 355	VEL POT 00165
320 ICENT = IXB	VEL POT 00166
GO TO 350	VEL POT 00167
330 ICENT = ((IROW-IRFB)/NSUBDV)*NSUBDV + IXB	VEL POT 00168
IF (ICENT .GT. MXBS) ICENT = ICENT - NSUBDV	VEL POT 00169
GO TO 350	VEL POT 00170
340 ICENT = IROW	VEL POT 00171
C	VEL POT 00172
350 CONTINUE	VEL POT 00173
IUCENT = (ICENT-IXBW)/NSUBDV + 1	VEL POT 00174
355 CONTINUE	VEL POT 00175
JEXLOC = 1	VEL POT 00176
IF (.NOT. WING) JEXLOC = MYBSW + 1	VEL POT 00177
C	VEL POT 00178
C LOOP ON ALL (SUBDIVIDED) CHORDS FOR THE SURFACE AND DIAPHRAGM	VEL POT 00179
C	VEL POT 00180
DO 1200 JCCL = 1,MYBSX	VEL POT 00181
C	VEL POT 00182
C GET THE CURRENT (SUBDIVIDED) BOX CODE	VEL POT 00183
CALL DCODER(IBOX,LBXCD, IROW,JCCL, IROW,JCCL, .T., NCDBOX)	VEL POT 00184
IF (SUBOFF) GO TO 480	VEL POT 00185
C	VEL POT 00186
C - - - - -	VEL POT 00187
C	VEL POT 00188
C GET INFORMATION ABOUT POSITION WITHIN UNSUBDIVIDED BOX	VEL POT 00189
C	VEL POT 00190
C ICENT = I-LOCATION (SUBDIVIDED) OF THE CONTROL POINT	VEL POT 00191
C JCENT = J-LOCATION (SUBDIVIDED) OF THE CONTROL POINT	VEL POT 00192
C IUCENT = UNSUBDIVIDED I-LOCATION OF ASSOCIATED CONTROL POINT	VEL POT 00193
C JUCENT = UNSUBDIVIDED J-LOCATION OF ASSOCIATED CONTROL POINT	VEL POT 00194
C IPCENT = UNSUBDIVIDED I-LOCATION OF THE NEAREST PLANFORM	VEL POT 00195
C CONTROL POINT, IF THE SUBDIVIDED BOX IS ON-PLANFORM	VEL POT 00196
C CBOX = .T., CURRENT BOX IS A CENTER BOX	VEL POT 00197
C LBOX = .T., THIS IS THE LAST BOX ASSOCIATED WITH THE CONTROL	VEL POT 00198
C POINT	VEL POT 00199
C NDCEN = CODE FOR CONTROL POINT	VEL POT 00200
C NCDBOX = CODE FOR THE CURRENT SUBDIVIDED BOX	VEL POT 00201

C	IICENT = I-LOCATION (SUBDIVIDED) FOR THE NEAREST PLANFORM	VELPOT	00202
C	CONTROL POINT	VELPOT	00203
C		VELPOT	00204
	LBOX = .F.	VELPOT	00205
	JUCENT = (JCOL - NSUBCN)/XSUBDV + 1.5	VELPOT	00206
	JCENT = NSUBBV * JUCENT - NSUBD2	VELPOT	00207
	IF (.NOT. CROW) GO TO 410	VELPOT	00208
	IF (JCENT .NE. JCOL) GO TO 410	VELPOT	00209
	CBOX = .T.	VELPOT	00210
	NDCEN = NDCBOX	VELPOT	00211
	IICENT = INOW	VELPOT	00212
	IF (NDCBOX) 450,1100,450	VELPOT	00213
C	GET CENTER BOX CODE, NDCEN	VELPOT	00214
410	CBOX = .F.	VELPOT	00215
	NDCEN = 0	VELPOT	00216
	IF (ICENT .LE. 0) GO TO 414	VELPOT	00217
	CALL DCODER (IBOX, LBXCD, ICENT, JCENT, ICENT, JCENT, .T., NDCEN)	VELPOT	00218
	IICENT = ICENT	VELPOT	00219
	GO TO 418	VELPOT	00220
414	IICENT = IXB - NSUBDV	VELPOT	00221
C		VELPOT	00222
418	CONTINUE	VELPOT	00223
	IF (NDCBOX - 1) 420,424,450	VELPOT	00224
C	SUBDIVIDED BOX IS NOT CONSIDERED (CODE = 0). IF CENTER CODE	VELPOT	00225
C	IS ALSO ZERO, LOOP TO NEXT BOX. OTHERWISE, CHECK FOR LAST BOX	VELPOT	00226
420	IF (NDCEN) 450,1100,450	VELPOT	00227
C		VELPOT	00228
C	SUBDIVIDED BOX CODE = 1. CHECK WHETHER ITS CONTROL PT = 1	VELPOT	00229
424	CONTINUE	VELPOT	00230
	IF (.NOT. CPLAN) GO TO 431	VELPOT	00231
	IF (TEXLOC(JEXLOC) .LT. FLIROW) JEXLOC = JEXLOC + MYBSW	VELPOT	00232
	IF (FEXLOC(JEXLOC) .GT. FLIROW) JEXLOC = JEXLOC - MYBSW	VELPOT	00233
C	DETERMINE WHETHER SUBDIVIDED BOX IS ON SAME PLANFORM AS	VELPOT	00234
C	IICENT (LOCATION OF NEAREST CONTROL POINT)	VELPOT	00235
	IF (JEXLOC .EQ. JCOL) GO TO 428	VELPOT	00236
C	SUBDIVIDED BOX IS ON THE TAIL. IS IICENT ON THE WING -	VELPOT	00237
	IF (FLOAT(IICENT) .LE. TEXLOC(JCENT)) GO TO 432	VELPOT	00238
C	NO. CHECK FOR IICENT OFF-PLANFORM.	VELPOT	00239
	GO TO 431	VELPOT	00240
C	SUBDIVIDED BOX IS ON THE WING. IS IICENT AFT OF THE WING T.E.	VELPOT	00241
428	IF (FLOAT(IICENT) .GT. TEXLOC(JCENT)) GO TO 432	VELPOT	00242
C	NO. CHECK FOR IICENT OFF-PLANFORM (I.E. DIAPHRAGM)	VELPOT	00243
431	CONTINUE	VELPOT	00244
C	CHECK CODE AT IICENT (NEAREST CONTROL POINT)	VELPOT	00245
	IF (NDCEN .EQ. 1) GO TO 450	VELPOT	00246
C	SUBDIVIDED ON-PLANFORM BOX DOES NOT LIE WITHIN AN UNSUBDIVIDED	VELPOT	00247
C	BOX WHOSE CONTROL POINT IS ON PLANFORM. SEARCH FORE AND AFT	VELPOT	00248
C	FOR THE NEAREST CONTROL POINT ON THE SURFACE.	VELPOT	00249
432	CONTINUE	VELPOT	00250
	IFCOMR = IFIX(FEXLOC(JEXLOC)) + 1	VELPOT	00251
	ILCOMR = TEXLOC(JEXLOC)	VELPOT	00252
	IMAX = 2*NSUBDV	VELPOT	00253
	DO 433 I = NSUBDV, IMAX, NSUBDV	VELPOT	00254
	IICENT = ICENT + I	VELPOT	00255
	IF (IICENT .GT. ILCOMR) GO TO 434	VELPOT	00256
	CALL DCODER (IBOX, LBXCD, IICENT, JCENT, IICENT, JCENT, .T., NCD)	VELPOT	00257
	IF (NCD .EQ. 1) GO TO 440	VELPOT	00258

434 CONTINUE	VEL POT 00259
IIICENT = ICENT-1	VEL POT 00260
IF (IIICENT .LT. IFCONR) GO TO 438	VEL POT 00261
CALL DCODER (IBOX, LBXCD, IIICENT, JCENT, IICENT, JCENT, .T., NCD)	VEL POT 00262
IF (NCD .EQ. 1) GO TO 440	VEL POT 00263
438 CONTINUE	VEL POT 00264
C NO CENTER BOX FOUND. A WARNING DIAGNOSTIC WILL BE PRINTED,	VEL POT 00265
C THEN COMPUTATION WILL CONTINUE AT 420	VEL POT 00266
GO TO 3010.	VEL POT 00267
C A BOX WAS FOUND	VEL POT 00268
440 CONTINUE	VEL POT 00269
C	VEL POT 00270
C THE ASSOCIATED CONTROL POINT HAS BEEN FOUND. GET THE UNSUB-	VEL POT 00271
C DIVIDED SUBSCRIPT.	VEL POT 00272
450 CONTINUE	VEL POT 00273
IIICENT = (IIICENT-IXBW)/NSUBDV + 1	VEL POT 00274
IF (LRGW) GO TO 470	VEL POT 00275
IF (.NOT. FROM) GO TO 460	VEL POT 00276
IF (JCCL .NE. JCENT-NSUBD2) GO TO 460	VEL POT 00277
FBOX = .T.	VEL POT 00278
GO TO 500	VEL POT 00279
460 CONTINUE	VEL POT 00280
FBOX = .F.	VEL POT 00281
GO TO 500	VEL POT 00282
470 CONTINUE	VEL POT 00283
IF (JCCL .EQ. JCENT+NSUBD2) LBOX = .T.	VEL POT 00284
FBOX = .F.	VEL POT 00285
GO TO 500	VEL POT 00286
C	VEL POT 00287
C - - - - -	VEL POT 00288
C	VEL POT 00289
C SET UP VALUES FOR AN UNSUBDIVIDED CASE	VEL POT 00290
C TEST FOR NON-ZERO BOX CODE -	VEL POT 00291
480 CONTINUE	VEL POT 00292
IF (NCDBOX .EQ. 0) GO TO 1100	VEL POT 00293
ICENT = IRCW	VEL POT 00294
IIICENT = ICENT	VEL POT 00295
IUCENT = ICENT	VEL POT 00296
JCENT = JCCL	VEL POT 00297
JUCENT = JCCL	VEL POT 00298
CBOK = .T.	VEL POT 00299
LBOX = .T.	VEL POT 00300
NCDCEN = NCDBOX	VEL POT 00301
FBOX = .T.	VEL POT 00302
IIICENT = ICENT	VEL POT 00303
C	VEL POT 00304
500 CONTINUE	VEL POT 00305
IF (NCDBOX .GT. 0) GO TO 510	VEL POT 00306
FULLBX(JUCENT) = .F.	VEL POT 00307
IF (LBOX) GO TO 1040	VEL POT 00308
IF (FBOX) GO TO 515	VEL POT 00309
GO TO 1100	VEL POT 00310
C	VEL POT 00311
C THE BOX IS TO BE CONSIDERED. ARE N-HAT TERMS NECESSARY -	VEL POT 00312
510 CONTINUE	VEL POT 00313
C ARE N-HAT TERMS ALREADY AVAILABLE -	VEL POT 00314
IF (FBOX) GO TO 515	VEL POT 00315

IF (NCD BOX .NE. 1) GO TO 860	VEL POT 00316
IF (.NOT. WING) GO TO 850	VEL POT 00317
IF (IPCENT .NE. IJCENT) GO TO 830	VEL POT 00318
GO TO 850	VEL POT 00319
C	VEL POT 00320
C - - - - -	VEL POT 00321
C	VEL POT 00322
C COMPUTE N-HAT TERMS FOR THIS (UNSUBDIVIDED) BOX	VEL POT 00323
C	VEL POT 00324
515 CONTINUE	VEL POT 00325
ENRULU(JUCENT) = (0.,0.)	VEL POT 00326
ENRLLL(JUCENT) = (0.,0.)	VEL POT 00327
DPHIL(JUCENT) = (0.,0.)	VEL POT 00328
C ARE LEFT SURFACE CONTRIBUTIONS POSSIBLE -	VEL POT 00329
IF (IJCENT-IXBU .LT. JUCENT) GO TO 800	VEL POT 00330
IF (PSIS .EQ. 0 .OR. .NOT. DIHS) GO TO 800	VEL POT 00331
C GET AIC ARRAYS W AND V FOR LEFT SURFACE INFLUENCE ON RT SURFACE	VEL POT 00332
CALL GETAIC(JUCENT,WW, 0, IR)	VEL POT 00333
IF (IR .NE. 0) GO TO 800	VEL POT 00334
NUBMIN = JUCENT	VEL POT 00335
NUBMAX = IJCENT - IXBU	VEL POT 00336
I = IJCENT - JUCENT	VEL POT 00337
YMBAR = COS(2*PSIS)* (JUCENT-.5)	VEL POT 00338
JBAR = YMBAR + 1	VEL POT 00339
C GET REFERENCE LOCATION IN AIC ARRAYS	VEL POT 00340
IF (YBAR) 520,525,530	VEL POT 00341
520 JINCR = 1	VEL POT 00342
GO TO 535	VEL POT 00343
525 IAIC = NUBMIN**2	VEL POT 00344
INCAIC = 2*NUBMIN + 1	VEL POT 00345
JINCR = -1	VEL POT 00346
GO TO 540	VEL POT 00347
530 JINCR = -1	VEL POT 00348
535 IAIC = NUBMIN**2 + NUBMIN	VEL POT 00349
INCAIC = 2*NUBMIN + 2	VEL POT 00350
C	VEL POT 00351
C LOOP FORWARD OF BOX FOR WING/WING (TAIL/TAIL) N-HAT TERMS	VEL POT 00352
540 CONTINUE	VEL POT 00353
DO 590 NUBAR = NUBMIN,NUBMAX	VEL POT 00354
MUAIC1 = MUAIC(1,NUBAR+1)	VEL POT 00355
MUAIC2 = MUAIC(2,NUBAR+1)	VEL POT 00356
IF (MUAIC2 .EQ. 0) GO TO 585	VEL POT 00357
IF (YBAR .GE. 0) GO TO 550	VEL POT 00358
JCOLL = -JBAR - NUBAR + MUAIC1	VEL POT 00359
GO TO 560	VEL POT 00360
550 JCOLL = -JBAR + NUBAR - MUAIC1 + 2	VEL POT 00361
560 CONTINUE	VEL POT 00362
C	VEL POT 00363
C LOOP LEFT OF RECEIVING CHORD TO GET LEFT SURFACE CONTRIBUTIONS	VEL POT 00364
DO 580 MUAI = MUAIC1,MUAIC2	VEL POT 00365
IF (JCOLL .LE. 0) GO TO 570	VEL POT 00366
CALL DCODER(IBOX,LBXCD, I,JCOLL, I,JCOLL, .F., ICD)	VEL POT 00367
IF (ICD .EQ. 0) GO TO 570	VEL POT 00368
C A CONTRIBUTING BOX HAS BEEN FOUND. GET THE AIC LOCATION	VEL POT 00369
KAIC = IAIC + MUAI	VEL POT 00370
C GET LOCATION IN N ARRAYS FOR THE VALUES AT BOX (I,JCOLL)	VEL POT 00371
IDS = LOCSDW(I,JCOLL, IPNTDW,LPNTDW, 1, LPNTDW)	VEL POT 00372

C	COEF = (CPSIS2*W(KAIC) - SPSIS2*V(KAIC)) * SYMTY	VEL POT	00373
	ENRULU(JUCENT) = ENRUS(IDS)*COEF + ENRULU(JUCENT)	VEL POT	00374
	ENRLL(JUCENT) = -ENRLS(IDS)*COEF + ENRLL(JUCENT)	VEL POT	00375
	DRHIL(JUCENT) = (ENRUS(IDS) - ENRLS(IDS)) * C(KAIC) * SYMTY +	VEL POT	00376
	1 DRHIL(JUCENT)	VEL POT	00377
	570 CONTINUE	VEL POT	00378
C	JCOLL = JCOLL + JINCR	VEL POT	00379
	580 CONTINUE	VEL POT	00380
C	END OF LOOP FOR LEFT ROW CONTRIBUTIONS	VEL POT	00381
C	585 CONTINUE	VEL POT	00382
	I = I - 1	VEL POT	00383
	IF (I .LT. IXBU) GO TO 600	VEL POT	00384
	IAIC = INCAIC + IAIC	VEL POT	00385
	INCAIC = INCAIC + 2	VEL POT	00386
	590 CONTINUE	VEL POT	00387
C	END OF LOOP FORWARD ON ROWS, TO COMPUTE LEFT SURFACE OUT-OF-	VEL POT	00388
C	PLANE EFFECTS, FROM 540	VEL POT	00389
C	IF THIS IS AN ON-PLANFORM TAIL BOX, THERE ARE WING-TAIL	VEL POT	00390
C	CONTRIBUTIONS	VEL POT	00391
	600 CONTINUE	VEL POT	00392
	IF (WING) GO TO 830	VEL POT	00393
	ENRURW(JUCENT) = (0.,0.)	VEL POT	00394
	ENRULW(JUCENT) = (0.,0.)	VEL POT	00395
	IF (NGUCEN .NE. 1) GO TO 860	VEL POT	00396
C	COMPUTE THE RIGHT WING CONTRIBUTION TO THE TAIL BOX	VEL POT	00397
	II = 1	VEL POT	00398
	IF (PSIW .EQ. PSIT) II = 2	VEL POT	00399
	CALL GETAIC(JUCENT,RWT, II, IR)	VEL POT	00400
	IF (IR .NE. 0) GO TO 700	VEL POT	00401
	NUBMIN = ABS(EL) + .5	VEL POT	00402
	NUBMAX = IUCENT - 1	VEL POT	00403
	I = IUCENT - NUBMIN	VEL POT	00404
	YUBAR = (JUCENT - .5) * COS(PSIDIF) + CARL * SIN(PSIW)	VEL POT	00405
	JBAR = YUBAR	VEL POT	00406
	IF (YUBAR .GE. 0) JBAR = JBAR + 1	VEL POT	00407
	IF (YBAR) 620,625,630	VEL POT	00408
	620 JINCR = -1	VEL POT	00409
	GO TO 635	VEL POT	00410
	625 IAIC = NUBMIN*2	VEL POT	00411
	INCAIC = 2*NUBMIN + 1	VEL POT	00412
	JYNCR = 1	VEL POT	00413
	GO TO 640	VEL POT	00414
	630 JINCR = 1	VEL POT	00415
	635 IAIC = NUBMIN*2 + NUBMIN	VEL POT	00416
	INCAIC = 2*NUBMIN + 2	VEL POT	00417
	640 CONTINUE	VEL POT	00418
C	LOOP FORWARD OVER THE RIGHT WING	VEL POT	00419
C	DO 690 NUBAR = NUBMIN,NUBMAX	VEL POT	00420
	MUIC1 = MUIC(1,NUBAR+1)	VEL POT	00421
	MUIC2 = MUIC(2,NUBAR+1)	VEL POT	00422
	IF (MUIC2 .EQ. 0) GO TO 685	VEL POT	00423
		VEL POT	00424
		VEL POT	00425
		VEL POT	00426
		VEL POT	00427
		VEL POT	00428
		VEL POT	00429

IF (YBAR .GE. 0) GO TO 650	VELPOT 00430
JCOLR = JBAR + NUBAR - MUAIC2 + 1	VELPOT 00431
GO TO 660	VELPOT 00432
650 JCOLR = JBAR - NUBAR + MUAIC1 - 1	VELPOT 00433
660 CONTINUE	VELPOT 00434
C	VELPOT 00435
C LOOP ON A ROW OF WING BOXES, COMPUTING RIGHT HAND WING-TAIL	VELPOT 00436
C CONTRIBUTIONS	VELPOT 00437
DO 680 MUA1 = MUAIC1, MUAIC2	VELPOT 00438
IF (JCOLR .LE. 0) GO TO 670	VELPOT 00439
CALL DCDER (IBOXW, LBXCDW, I, JCOLR, I, JCOLR, .F., ICD)	VELPOT 00440
IF (ICD .EQ. 0) GO TO 670	VELPOT 00441
C A CONTRIBUTING BOX HAS BEEN FOUND. GET THE AIC LOCATION	VELPOT 00442
KAIC = IAIC + MUA1	VELPOT 00443
C GET THE NORMAL-WASH LOCATION	VELPOT 00444
IDS = LOCSDW (I, JCOLR, IPNTDW, LPNTDW, 1, LPNTDW)	VELPOT 00445
C ADD THIS CONTRIBUTION TO N-HAT	VELPOT 00446
IF (II .EQ. 2) GO TO 665	VELPOT 00447
ENRURW(JUCENT) = (COS(PSIDIF)*W(KAIC) - SIN(PSIDIF)*Y(KAIC))	VELPOT 00448
1 * ENRUS(IDS) + ENRURW(JUCENT)	VELPOT 00449
GO TO 670	VELPOT 00450
665 CONTINUE	VELPOT 00451
ENRURW(JUCENT) = COS(PSIDIF)*W(KAIC) * ENRUS(IDS) +	VELPOT 00452
1 ENRURW(JUCENT)	VELPOT 00453
670 CONTINUE	VELPOT 00454
JCOLR = JCOLR + JINCR	VELPOT 00455
680 CONTINUE	VELPOT 00456
C END OF LOOP FOR RIGHT WING ROW CONTRIBUTIONS	VELPOT 00457
C	VELPOT 00458
685 CONTINUE	VELPOT 00459
I = I - 1	VELPOT 00460
IF (I .LE. 0) GO TO 700	VELPOT 00461
IAIC = IAIC + INCAIC	VELPOT 00462
INCAIC = INCAIC + 2	VELPOT 00463
690 CONTINUE	VELPOT 00464
C END OF LOOP FORWARD ON ROWS TO COMPUTE RIGHT WING OUT-OF-	VELPOT 00465
C PLANE EFFECTS ON THE TAIL, FROM 640	VELPOT 00466
C	VELPOT 00467
C DETERMINE WHETHER LEFT WING INFLUENCE IS TO BE COMPUTED	VELPOT 00468
700 CONTINUE	VELPOT 00469
IF (SYM .EQ. 0) GO TO 800	VELPOT 00470
C GET AIC ARRAYS W AND Y FOR LEFT WING INFLUENCE ON TAIL	VELPOT 00471
II = 1	VELPOT 00472
IF (-PSIW .EQ. PSIT) II = 2	VELPOT 00473
CALL GETAIC(JUCENT, LWT, II, IR)	VELPOT 00474
IF (IR .NE. 0) GO TO 800	VELPOT 00475
NUBIN = ABS(EL) + .5	VELPOT 00476
NUBMAX = JUCENT - 1	VELPOT 00477
I = JUCENT - NUBIN	VELPOT 00478
YHUBAR = - COS(PSIW + PSIT)*(JUCENT-.5) + CAPL*SIN(PSIW)	VELPOT 00479
JBAR = YHUBAR	VELPOT 00480
IF (YHUBAR .GE. 0) JBAR = JBAR + 1	VELPOT 00481
IF (YBAR) 720, 725, 730	VELPOT 00482
720 JINCR = 1	VELPOT 00483
GO TO 735	VELPOT 00484
725 IAIC = NUBIN+2	VELPOT 00485
INCAIC = 2*NUBIN + 1	VELPOT 00486

JINCR = -1	VELPOT 00487
GO TO 740	VELPOT 00488
730 JINCR = -1	VELPOT 00489
735 IAIC = NUBMIN*2 + NUBMIN	VELPOT 00490
INCAIC = 2*NUBMIN + 2	VELPOT 00491
740 CONTINUE	VELPOT 00492
C	VELPOT 00493
C LOOP FORWARD TO GET LEFT WING CONTRIBUTION TO THE TAIL	VELPOT 00494
IF (SYM .EQ. 0) GO TO 800	VELPOT 00495
DO 790 NUBAR = NUBMIN,NUBMAX	VELPOT 00496
MUAIC1 = MUAIC(1,NUBAR+1)	VELPOT 00497
MUAIC2 = MUAIC(2,NUBAR+1)	VELPOT 00498
IF (MUAIC2 .LE. 0) GO TO 785	VELPOT 00499
IF (YBAR .GE. 0) GO TO 750	VELPOT 00500
JCOLL = JBAR - NUBAR + MUAIC1 - 1	VELPOT 00501
GO TO 760	VELPOT 00502
750 JCOLL = JBAR + NUBAR - MUAIC1 + 1	VELPOT 00503
760 CONTINUE	VELPOT 00504
C	VELPOT 00505
C LOOP ON LEFT WING ROW TO GET LEFT WING CONTRIBUTION TO TAIL	VELPOT 00506
DO 780 MUAT = MUAIC1,MUAIC2	VELPOT 00507
IF (JCOLL .LE. 0) GO TO 770	VELPOT 00508
CALL DCDER(1BOXW,LBXCDW, I,JCOLL, I-JCOLL, .F., ICD)	VELPOT 00509
IF (ICD .EQ. 0) GO TO 770	VELPOT 00510
C A CONTRIBUTING BOX HAS BEEN FOUND. GET THE AIC LOCATION	VELPOT 00511
KAIC = IAIC + MUAT	VELPOT 00512
C GET THE NORMAL WASH LOCATION	VELPOT 00513
IDS = LOCSDW(I, JCOLL, IPNTD, LPNTD, 1, LPNTD)	VELPOT 00514
C ADD THIS CONTRIBUTION TO N-HAT	VELPOT 00515
IF (II .EQ. 2) GO TO 765	VELPOT 00516
ENRULW(JUCENT) = (CPSISM*W(KAIC) - SPSISM*V(KAIC))	VELPOT 00517
1 * ENRUS(IDS) + ENRULW(JUCENT)	VELPOT 00518
GO TO 770	VELPOT 00519
765 CONTINUE	VELPOT 00520
ENRULW(JUCENT) = CPSISM*W(KAIC) * ENRUS(IDS) + ENRULW(JUCENT)	VELPOT 00521
770 CONTINUE	VELPOT 00522
JCOLL = JCOLL + JINCR	VELPOT 00523
780 CONTINUE	VELPOT 00524
C END OF LOOP FOR LEFT WING ROW CONTRIBUTIONS	VELPOT 00525
C	VELPOT 00526
785 CONTINUE	VELPOT 00527
I = I-1	VELPOT 00528
IF (I .LE. 0) GO TO 800	VELPOT 00529
IAIC = IAIC + INCAIC	VELPOT 00530
INCAIC = INCAIC + 2	VELPOT 00531
790 CONTINUE	VELPOT 00532
C END OF LOOP FORWARD ON ROWS, TO COMPUTE LEFT WING OUT-OF-PLANE	VELPOT 00533
C EFFECTS ON THE TAIL, FROM 740	VELPOT 00534
C	VELPOT 00535
IF (SYM .LY. 0) ENRULW(JUCENT) = - ENRULW(JUCENT)	VELPOT 00536
800 CONTINUE	VELPOT 00537
C	VELPOT 00538
C - - - - -	VELPOT 00539
C	VELPOT 00540
C COMPUTE THE UNSUBDIVIDED NORMAL WASH VALUES, IF THE BOX IS ON-	VELPOT 00541
C PLATFORM. IF NOT, GET THE VALUE FROM THE INTERFERENCE TERMS	VELPOT 00542
C AND THE CONDITION THAT DELTA-PHI = 0 ON ANY DIAPHRAGM, MODIFI-	VELPOT 00543

C	ED BY WAKE EFFECTS WHERE APPLICABLE.	VELPOT	00544
C		VELPOT	00545
C	- TAIL -	VELPOT	00546
	850 CONTINUE	VELPOT	00547
	IF (NCDBOX .NE. 1) GO TO 860	VELPOT	00548
C	GET DEFLECTION AND SLOPE OF UNSUBDIVIDED TAIL BOX CENTER	VELPOT	00549
	IDS = LOCSDW(IPCENT+IOWLP, JUCENT, IPNTRM, LMODES,1,LMODES)	VELPOT	00550
	DFL = DEFSL(1,IDS)	VELPOT	00551
	SLP = DEFSL(2,IDS)	VELPOT	00552
C	COMPUTE TAIL NORMAL WASH VALUES	VELPOT	00553
	ENDIF = 2.0*(CMPLX(B1*SLP, XXVL*DFL) - ENRURW(JUCENT)	VELPOT	00554
1	- ENRULW(JUCENT)) + ENRLLL(JUCENT) - ENRULU(JUCENT)	VELPOT	00555
	ENSUM = - (ENRLLL(JUCENT) + ENRULU(JUCENT))	VELPOT	00556
C		VELPOT	00557
	IF (NCDCEN .NE. 1) GO TO 852	VELPOT	00558
	LOCNW = LOCSDW(IUCENT+IOWLPN, JUCENT, IPNTDW, LPNTDW,1,LPNTDW)	VELPOT	00559
	ENRUS(LOCNW) = 0.5*(ENSUM + ENDIF)	VELPOT	00560
	ENRLS(LOCNW) = 0.5*(ENSUM - ENDIF)	VELPOT	00561
	GO TO 850	VELPOT	00562
C		VELPOT	00563
C	- WING -	VELPOT	00564
	860 CONTINUE	VELPOT	00565
	IF (NCDBOX .NE. 1) GO TO 860	VELPOT	00566
C	GET DEFLECTION AND SLOPE OF UNSUBDIVIDED WING BOX CENTER	VELPOT	00567
	IDS = LOCSDW(IPCENT,JUCENT, IPNTRM, LMODES,1,LMODES)	VELPOT	00568
	DFL = DEFSL(1,IDS)	VELPOT	00569
	SLP = DEFSL(2,IDS)	VELPOT	00570
C	COMPUTE WING NORMAL WASH VALUES	VELPOT	00571
	ENSUM = -ENRULU(JUCENT) - ENRLLL(JUCENT)	VELPOT	00572
	ENDIF = ENSUM + (CMPLX(B1*SLP, XXVL*DFL) + ENRLLL(JUCENT)) * 2.0	VELPOT	00573
C		VELPOT	00574
	IF (IPCENT .NE. JUCENT) GO TO 852	VELPOT	00575
	LOCNW = LOCSDW(IUCENT,JUCENT, IPNTDW, LPNTDW, 1, LPNTDW)	VELPOT	00576
	ENRUS(LOCNW) = (ENSUM + ENDIF) * 0.5	VELPOT	00577
	ENRLS(LOCNW) = (ENSUM - ENDIF) * 0.5	VELPOT	00578
C		VELPOT	00579
C	NORMAL-WASH IS AVAILABLE IF THE BOX IS ON-PLATFORM	VELPOT	00580
	880 CONTINUE	VELPOT	00581
	IF (J.N. FROM) LOCNW = LOCSDW(IUCENT+IOWLPN, JUCENT, IPNTDW,LPNTDW,	VELPOT	00582
1	1,LPNTDW)	VELPOT	00583
	IF (SUBOFF) GO TO 855	VELPOT	00584
C	GET THE SUBDIVIDED VALUE FOR THE NORMAL WASH TERMS	VELPOT	00585
	DELPHA = ENRUS(LOCNW)	VELPOT	00586
	IF (DELPHA .EQ. XINIT) GO TO 830	VELPOT	00587
	DELPHB = ENRLS(LOCNW)	VELPOT	00588
	GO TO 854	VELPOT	00589
C	THE NEXT 2 STATEMENTS ARE ONLY HIT FOR A SUBDIVIDED PLATFORM	VELPOT	00590
C	BOX WITH NO ASSOCIATED PLATFORM CONTROL POINT	VELPOT	00591
	892 CONTINUE	VELPOT	00592
	FULLBX(JUCENT) = .FALSE.	VELPOT	00593
	DELPHA = (ENSUM + ENDIF) * 0.5	VELPOT	00594
	DELPHB = (ENSUM - ENDIF) * 0.5	VELPOT	00595
	894 CONTINUE	VELPOT	00596
	ENRUS = CMPLX(REAL(DELPHA)/XSUBDV, AIMAG(DELPHA)/XSUBDV +	VELPOT	00597
1	XXVL*(IROW-IICENT)*REAL(DELPHA))	VELPOT	00598
	ENRLS = CMPLX(REAL(DELPHB)/XSUBDV, AIMAG(DELPHB)/XSUBDV +	VELPOT	00599
1	XXVL*(IROW-IICENT)*REAL(DELPHB))	VELPOT	00600

IF (CBOX) GO TO 870	VELPOT 00601
GO TO 1030	VELPOT 00602
855 CONTINUE	VELPOT 00603
ENSRUS = ENSRUS(LOCNM)	VELPOT 00604
ENSRLS = ENSRLS(LOCNM)	VELPOT 00605
GO TO 870	VELPOT 00606
860 CONTINUE	VELPOT 00607
FULLBX(JUCENT) = .F.	VELPOT 00608
C	VELPOT 00609
C - - - - -	VELPOT 00610
C	VELPOT 00611
C DETERMINE THE VELOCITY POTENTIAL CONTRIBUTIONS FROM	VELPOT 00612
C BOXES LYING AHEAD OF THE CURRENT BOX	VELPOT 00613
C	VELPOT 00614
870 CONTINUE	VELPOT 00615
IF (MIS .NE. 0 .AND. DIHS) GO TO 880	VELPOT 00616
C THE SURFACE IS PLANAR. GET FULL SURFACE CONTRIBUTIONS	VELPOT 00617
DELPH = B(IROW,JCOL, PKERNL,SKERNL, IBOX,LBXCD, WING, .F.)	VELPOT 00618
GO TO 890	VELPOT 00619
C	VELPOT 00620
C DIHEDRAL ANGLE IS TO BE ACCOUNTED FOR. GET THE PLANAR	VELPOT 00621
C (SUBDIVIDED) CONTRIBUTION OF THE RIGHT SURFACE	VELPOT 00622
880 CONTINUE	VELPOT 00623
DELPH = B(IROW,JCOL, PKERNL,SKERNL, IBOX,LBXCD, WING, .T.)	VELPOT 00624
C ADD THE SPATIAL LEFT SURFACE CONTRIBUTIONS	VELPOT 00625
DELPH = DELPH + DRHL(JUCENT)	VELPOT 00626
890 CONTINUE	VELPOT 00627
IF (NCDBOX - 2) 910,1000,980	VELPOT 00628
C	VELPOT 00629
C - - - - -	VELPOT 00630
C	VELPOT 00631
C THE BOX IS ON-PLANFORM, CENTER. COMPLETE THE CALCULATION OF	VELPOT 00632
C THE VELOCITY POTENTIAL	VELPOT 00633
C	VELPOT 00634
910 CONTINUE	VELPOT 00635
IDS = LOCENW(JUCENT+IOWLP, JUCENT, IPNTRM,LMODES, 1, LMODES)	VELPOT 00636
IF (SUBOFF) GO TO 915	VELPOT 00637
DELPHI(IDS) = (ENSRUS-ENSRLS) * SKERNL(1) + DELPH	VELPOT 00638
GO TO 920	VELPOT 00639
915 CONTINUE	VELPOT 00640
DELPHI(IDS) = (ENSRUS-ENSRLS) * PKERNL(1) + DELPH	VELPOT 00641
920 CONTINUE	VELPOT 00642
C	VELPOT 00643
C COMPUTE ANY TRAILING EDGE VELOCITY POTENTIALS ASSOCIATED	VELPOT 00644
C WITH THIS UNSUBDIVIDED BOX	VELPOT 00645
C	VELPOT 00646
C IS THIS A TRAILING EDGE BOX -	VELPOT 00647
IF (.NOT. CPLAN) GO TO 930	VELPOT 00648
IF (TEXLOC(JEXLOC) .LT. FLIROW) JEXLOC = JEXLOC + MYBSW	VELPOT 00649
IF (TEXLOC(JEXLOC) .GT. FLIROW) JEXLOC = JEXLOC - MYBSW	VELPOT 00650
930 CONTINUE	VELPOT 00651
JJ = JEXLOC - NSUBD2	VELPOT 00652
TEXMIN = TEXLOC(JJ)	VELPOT 00653
IF (NSUBDV .EQ. 1) GO TO 935	VELPOT 00654
DO 932 J = 2, NSUBDV	VELPOT 00655
JJ = JJ + 1	VELPOT 00656
TEXMIN = AMIN(TEXMIN, TEXLOC(JJ))	VELPOT 00657

932 CONTINUE	VEL POT	00658
933 IF (TEXMIN .GE. FLIRCH + XSUBDV) GO TO 1030	VEL POT	00659
C YES. GET THE BASIC VELOCITY POTENTIAL	VEL POT	00660
DELPHB = DELPHI(1DS)	VEL POT	00661
C TEST UNSUBDIVIDED BOX AHEAD OF CURRENT ONE -	VEL POT	00662
CALL DCODER(1BOX,LBXCD, IROW+NSUBDV,JCOL, IROW+NSUBDV,JCOL,	VEL POT	00663
1 .T., NCBA)	VEL POT	00664
IF (NCBA .NE. 1) GO TO 940	VEL POT	00665
C IT IS ON PLANFORM. TEST FOR TIP CHORD -	VEL POT	00666
IF (JEXLOC .EQ. JCOL .AND. JUCENT .LT. HYBW) GO TO 950	VEL POT	00667
IF (JEXLOC .GT. JCOL .AND. JUCENT .LT. HYBT) GO TO 950	VEL POT	00668
C BOX IS ON THE TIP CHORD. CHECK FOR THIRD TIP BOX -	VEL POT	00669
IR = IROW - 2*NSUBDV	VEL POT	00670
IF (IR .LT. 1XB) GO TO 940	VEL POT	00671
CALL DCODER(1BOX,LBXCD, IR, JCOL, IR,JCOL, .T., NCDC)	VEL POT	00672
IF (NCDC .EQ. 1) GO TO 950	VEL POT	00673
C TRY MACH RAY EXTRAPOLATION. ARE THE 2 RAY BOXES ON-PLANFORM -	VEL POT	00674
940 CONTINUE	VEL POT	00675
JC = JCOL - NSUBDV	VEL POT	00676
CALL DCODER(1BOX,LBXCD, IROW,JC, IROW,JC, .T., NCDD)	VEL POT	00677
IF (NCDD .NE. 1) GO TO 945	VEL POT	00678
JC = JC - NSUBDV	VEL POT	00679
CALL DCODER(1BOX,LBXCD, IROW+NSUBDV,JC, IROW+NSUBDV,JC, .T.,NCDD)	VEL POT	00680
IF (NCDD .NE. 1) GO TO 945	VEL POT	00681
C	VEL POT	00682
C MACH RAY EXTRAPOLATION, FOLLOWED BY CHORDWISE LINEAR EXTRA-	VEL POT	00683
C POLATION	VEL POT	00684
IDPHM2 = LOCSDW(JUCENT-1+IOMLP, JUCENT-2, IPNTRM,LMODES,1,LMODES)	VEL POT	00685
IDPHM1 = LOCSDW(JUCENT+IOMLP, JUCENT-1, IPNTRM,LMODES,1,LMODES)	VEL POT	00686
DELPHM = DELPHI(IDPHM2)	VEL POT	00687
DELPHC = DELPHI(IDPHM1)	VEL POT	00688
SDELPH = (2.0*DELPHC - DELPHM - DELPHB)/XSUBDV	VEL POT	00689
GO TO 955	VEL POT	00690
C	VEL POT	00691
C MACH RAY UNAVAILABLE. ARE THERE 2 BOXES ON THE CHORD -	VEL POT	00692
945 IF (NCBA .NE. 1) GO TO 3020	VEL POT	00693
C	VEL POT	00694
C CHORDWISE LINEAR EXTRAPOLATION	VEL POT	00695
C	VEL POT	00696
950 CONTINUE	VEL POT	00697
IDPHM1 = LOCSDW(JUCENT-1+IOMLP, JUCENT, IPNTRM,LMODES, 1,LMODES)	VEL POT	00698
SDELPH = (DELPHB - DELPHI(IDPHM1))/XSUBDV	VEL POT	00699
C	VEL POT	00700
C LOOP TO COMPUTE AND STORE TRAILING EDGE VELOCITY POTENTIALS	VEL POT	00701
955 CONTINUE	VEL POT	00702
JA = JEXLOC - NSUBD2	VEL POT	00703
JB = JEXLOC + NSUBD2	VEL POT	00704
DO 960 JJ = JA,JB	VEL POT	00705
XINCR = TEXLOC(JJ) - IROW	VEL POT	00706
IF (XINCR .LT. -XSUBDV/2.0) GO TO 960	VEL POT	00707
TVP(JJ) = DELPHB + XINCR*SDELPH	VEL POT	00708
960 CONTINUE	VEL POT	00709
C ALL TRAILING EDGE VALUES HAVE BEEN COMPUTED FOR THIS	VEL POT	00710
C UNSUBDIVIDED BOX.	VEL POT	00711
GO TO 1030	VEL POT	00712
C	VEL POT	00713
C - - - - -	VEL POT	00714

C		VELPOT	00715
C	BOX IS IN THE TRAILING EDGE DIAPHRAGM AREA. COMPUTE THE WAKE	VELPOT	00716
C	VELOCITY POTENTIAL CONTRIBUTION	VELPOT	00717
	980 CONTINUE	VELPOT	00718
	IF (COPLAN) GO TO 983	VELPOT	00719
	JJ = JEXLOC	VELPOT	00720
	GO TO 990	VELPOT	00721
	985 JJ = JCCL	VELPOT	00722
	IF (JJ .GT. MYBST) GO TO 990	VELPOT	00723
	JJT = JJ + MYBSW	VELPOT	00724
	IF (FLOAT(IROW) .GT. TEXLOC(JJT)) JJ = JJT	VELPOT	00725
	990 CONTINUE	VELPOT	00726
C	COMPUTE (X-DISTANCE/B1) * K1	VELPOT	00727
	XDKVL = (FLOAT(IROW)-TEXLOC(JJ)) * XKVL/XSUBDV	VELPOT	00728
	AZERO = TVP(JJ) * CMPLX(COS(XDKVL), -SIN(XDKVL))	VELPOT	00729
	IF (.NOT. CBOX) GO TO 1010	VELPOT	00730
	IF (JUCENT+IOWLP .GT. NPNTS) GO TO 1010	VELPOT	00731
C	SET DELPHI VALUE TO ZERO, TO CLEAN UP LEFT OVER MODE SHAPES	VELPOT	00732
	I = LOCSDW(IUCENT+IOWLP,JUCENT, IPNTRM, LMODES,1,LMODES)	VELPOT	00733
	IF (I .NE. 0) DELPHI(I) = (0.,0.)	VELPOT	00734
	GO TO 1010	VELPOT	00735
C		VELPOT	00736
C	BOX IS IN A LEADING EDGE OR TIP DIAPHRAGM AREA	VELPOT	00737
	1000 CONTINUE	VELPOT	00738
	AZERO = (0.,0.)	VELPOT	00739
C		VELPOT	00740
C	COMPUTE NORMAL-WASH VALUES FOR A (SUBDIVIDED) DIAPHRAGM BOX	VELPOT	00741
C		VELPOT	00742
	1010 CONTINUE	VELPOT	00743
	IF (SUBOFF) GO TO 1015	VELPOT	00744
	ENDIF = (AZERO - DELPHI) / SKENL(1)	VELPOT	00745
	ENSUM = -(ENRULU(JUCENT) + ENRLLL(JUCENT))	VELPOT	00746
	ENSUM = CMPLX(REAL(ENSUM), ATNAG(ENSUM) + XKVL*XSUBDV*(IROW -	VELPOT	00747
	1 JUCENT)*REAL(ENSUM)) / XSUBDV	VELPOT	00748
	GO TO 1020	VELPOT	00749
	1015 CONTINUE	VELPOT	00750
	ENDIF = (AZERO - DELPHI) / PKENL(1)	VELPOT	00751
	ENSUM = -(ENRULU(JUCENT) + ENRLLL(JUCENT))	VELPOT	00752
C		VELPOT	00753
	1020 CONTINUE	VELPOT	00754
	ENSRUS = 0.5*(ENSUM + ENDIF)	VELPOT	00755
	ENSRLS = 0.5*(ENSUM - ENDIF)	VELPOT	00756
	IF (SUBOFF) GO TO 1030	VELPOT	00757
	GO TO 1035	VELPOT	00758
C		VELPOT	00759
C	- - - - -	VELPOT	00760
C		VELPOT	00761
C	STORE THE NORMAL WASH VALUES	VELPOT	00762
C		VELPOT	00763
	1030 CONTINUE	VELPOT	00764
	IF (SUBOFF) GO TO 1100	VELPOT	00765
C		VELPOT	00766
C	STORE THE COMPUTED SUBDIVIDED NORMAL WASHES	VELPOT	00767
	1035 CONTINUE	VELPOT	00768
	CALL STOSDW (IROW,JCCL, ENSBD, IBOX,LBXCD, IXBS,MXBS,MYBBS,IRR)	VELPOT	00769
	IF (IRR .NE. 0) GO TO 3030	VELPOT	00770
	IF (.NOT. LBOX) GO TO 1100	VELPOT	00771

C		VEL POT	00772
C	DETERMINE THE UNSUBDIVIDED NORMAL WASH VALUES	VEL POT	00773
	1040 IF (NDCEN - 1) 1100,1050,1060	VEL POT	00774
C	IF THE BOX IS COMPLETE, THE VALUE IS ALREADY STORED -	VEL POT	00775
	1050 IF (FULLBX(JUCENT)) GO TO 1100	VEL POT	00776
C	THE UNSUBDIVIDED VALUE EQUALS THE AVERAGE OF ALL ITS	VEL POT	00777
C	SUBDIVIDED BOXES	VEL POT	00778
	1060 CONTINUE	VEL POT	00779
	II = ICENT - NSUBD2	VEL POT	00780
	JJJ = JCENT + NSUBD2	VEL POT	00781
	ENSRUS = (0.,0.)	VEL POT	00782
	ENSRLS = (0.,0.)	VEL POT	00783
	DO 1080 I = 1, NSUBDV	VEL POT	00784
	JJ = JCENT - NSUBD2	VEL POT	00785
	CALL DCOTER (IBOX, LBXCD, II, JJ, II, JJJ, .T., ICODE)	VEL POT	00786
	DO 1075 J = 1, NSUBDV	VEL POT	00787
	IF (ICODE(J) .EQ. 0) GO TO 1070	VEL POT	00788
	LOCNW = LOCSDW(II, JJ, IPNTSD, IPNTIN, IPNTOT, IPNTLS)	VEL POT	00789
	ENSRUS = ENSUBD(1, LOCNW) + ENSRUS	VEL POT	00790
	ENSRLS = ENSUBD(2, LOCNW) + ENSRLS	VEL POT	00791
	1070 CONTINUE	VEL POT	00792
	JJ = JJ + 1	VEL POT	00793
	1075 CONTINUE	VEL POT	00794
	II = II + 1	VEL POT	00795
	1080 CONTINUE	VEL POT	00796
	ENSRUE = ENSRUS/XSUBDV	VEL POT	00797
	ENSRLS = ENSRLS/XSUBDV	VEL POT	00798
C		VEL POT	00799
C	RESTORE THE PARTIAL BOX FLAG FOR THE NEXT ROW	VEL POT	00800
	FULLBX(JUCENT) = .T.	VEL POT	00801
C		VEL POT	00802
C	STORE THE UNSUBDIVIDED NORMAL WASHES	VEL POT	00803
	1090 CONTINUE	VEL POT	00804
	LOCNW = LOCSDW(IUCENT+IOMLPN, JUCENT, IPNTDW, LPNTDW, 1, LPNTDW)	VEL POT	00805
	IF (LOCNW .EQ. 0) GO TO 3040	VEL POT	00806
	ENSRUS(LOCNW) = ENSRUS	VEL POT	00807
	IF (NINE) GO TO 1095	VEL POT	00808
	ENSRLS(LOCNW) = ENSRLS	VEL POT	00809
	GO TO 1100	VEL POT	00810
	1095 ENSRLS(LOCNW) = ENSRLS	VEL POT	00811
	1100 CONTINUE	VEL POT	00812
	JEXLOC = JEXLOC + 1	VEL POT	00813
C		VEL POT	00814
C		VEL POT	00815
	1200 CONTINUE	VEL POT	00816
C	END OF LOOP ON (SUBDIVIDED) CHORDS, STARTING AT 355*	VEL POT	00817
C		VEL POT	00818
C		VEL POT	00819
	1300 CONTINUE	VEL POT	00820
C	END OF LOOP ON (SUBDIVIDED) ROWS, STARTING AT 320	VEL POT	00821
C		VEL POT	00822
C		VEL POT	00823
	RETURN	VEL POT	00824
C		VEL POT	00825
C	- - - - -	VEL POT	00826
C		VEL POT	00827
C	DIAGNOSTICS - ALL CALL FLUSH	VEL POT	00828

C		VEL POT	00829
3010	WRITE (NT6,9010)	VEL POT	00830
	WRITE (NT6,9999) IROW,JCQL, IUCENT,JUCENT	VEL POT	00831
	GO TO 420	VEL POT	00832
3020	WRITE (NT6,9020)	VEL POT	00833
	GO TO 3999	VEL POT	00834
3030	WRITE (NT6,9030)	VEL POT	00835
	GO TO 3999	VEL POT	00836
3040	WRITE (NT6,9040)	VEL POT	00837
	GO TO 3999	VEL POT	00838
C		VEL POT	00839
3999	WRITE (NT6,9999) IROW,JCQL, IUCENT,JUCENT	VEL POT	00840
	CALL FLUSH(1)	VEL POT	00841
C		VEL POT	00842
C		VEL POT	00843
9010	FORMAT(56H)*** WARNING - NO PLANFORM CONTROL POINT FOUND FOR SUBDI	VEL POT	00844
	1 52HVIDED BOX DURING VELOCITY POTENTIAL CALCULATIONS ***)	VEL POT	00845
9020	FORMAT(56H)*** ERROR - THE TIP BOX PATTERN DOES NOT ALLOW TRAILING	VEL POT	00846
	1 44H EDGE VELOCITY POTENTIALS TO BE COMPUTED ***)	VEL POT	00847
9030	FORMAT(56H)*** ERROR - FAILURE IN STORING SUBDIVIDED NORMAL-WASHES	VEL POT	00848
	1 4H ***)	VEL POT	00849
9040	FORMAT(53H)*** ERROR - FAILURE IN STORING CONTROL POINT NORMAL-	VEL POT	00850
	1 10H-WASHES ***)	VEL POT	00851
9999	FORMAT(14X,16H)SUBDIVIDED BOX (,I3,1H,I3,19H), CONTROL POINT (,	VEL POT	00852
	1 I2,1H,I2,1H))	VEL POT	00853
C		VEL POT	00854
	END	VEL POT	00855

	COMPLEX FUNCTION B (IROW,JCOL, PKERNL,SKERNL, IBOX,LBXCD,	B	00002
1	WING, DIH)	B	00003
C		B	00004
C	COMPUTES B = SUM OVER NU(SUM OVER MU((DOWNWASH)*(KERNEL)))	B	00005
C	(NU,MU ,NE. 0)	B	00006
C	B IS USED TO COMPUTE VELOCITY POTENTIALS OF ON-	B	00007
C	PLANFORM BOXES, OR DOWNWASHES OF DIAPHRAGM BOXES	B	00008
C		B	00009
C	PARAMETERS -	B	00010
C	IROW = ROW LOCATION OF BOX FOR WHICH B IS TO BE	B	00011
C	COMPUTED	B	00012
C	JCOL = COLUMN LOCATION OF BOX	B	00013
C	PKERNL = PRIMARY KERNEL ARRAY	B	00014
C	SKERNL = SUBDIVIDED KERNEL ARRAY	B	00015
C	IBOX = ARRAY OF BOX CODES	B	00016
C	LBXCD = LENGTH OF BOX CODE ARRAY	B	00017
C	WING = .T., WING. .F., TAIL	B	00018
C	DIH = .T., LEFT SIDE TO BE IGNORED (SURFACE HAS DIHDL	B	00019
C	= .F., LEFT SIDE TO BE INCLUDED	B	00020
C		B	00021
C	VALUES FROM COMMON -	B	00022
C	NSUBDV = NUMBER OF SUBDIVISIONS	B	00023
C	IXB = CENTER OF FIRST UNSUBDIVIDED BOX RELATIVE TO THE	B	00024
C	SUBDIVIDED PATTERN	B	00025
C	MYB = NUMBER OF UNSUBDIVIDED ROWS	B	00026
C	MYBB = NUMBER OF UNSUBDIVIDED CHORDS, INCLUDING DIAPH.	B	00027
C	MYBBS = NUMBER OF SUBDIVIDED CHORDS, INCLUDING DIAPH.	B	00028
C	MXSKRN = SIZE OF SUBDIVIDED KERNEL	B	00029
C	SYM = SYMMETRY INDICATOR	B	00030
C		B	00031
C	LRGT = NSUBDV + NSUBDV/2 + 1	B	00032
C	ENSUBD = SUBDIVIDED NORMAL-WASHES	B	00033
C	ENRUS, ENRLS = UNSUBDIVIDED NORMAL-WASHES	B	00034
C	IPNTDW = POINTER ARRAY FOR UNSUBDIVIDED NORMAL WASHES	B	00035
C	IPNTSD = POINTER ARRAY FOR SUBDIVIDED NORMAL WASHES (END-	B	00036
C	AROUND	B	00037
C	IPNTIN = NEXT AVAILABLE POINTER	B	00038
C	IPNTOT = FIRST POINTER IN USE	B	00039
C	IPNTLS = DIMENSION OF ARRAY IPNTSD	B	00040
C	LIMDNG = DIMENSION OF SUBDIVIDED NORMAL-WASH ARRAYS	B	00041
C		B	00042
	COMMON /GEOMTY/ COPLAN, NSUBDV, XSUBDV, NSUBDD, NSUBCN, NSURF,	GEOMTY	00002
1	B1,B1BETA,B1S,B1BTAS,MLAX,MLAZ,PSIW,	GEOMTY	00003
2	MXBW,MXBW,MYBW,MYBBW,MXBSW,MYBSW,MYBBSW,	GEOMTY	00004
3	IXBW,XCENTR	GEOMTY	00005
	LOGICAL COPLAN	GEOMTY	00006
	COMMON /GEOM2 / TLAX,YLAZ,PSIT,IXBT,MYBT,MYBBT,MYBST,MYBST,	GEOM2	00002
1	MYBBSY,IXGT,IXBOT,CAPL	GEOM2	00003
	COMMON / MODES/ SYM,SYMT,MTYPEW,MTYPE	MODECOM	00002
	COMMON /SNWASH/ IPNTSD(2,50), ENSUBD(2,600), IPNTIN,IPNTOT,IPNTLS	SNWASH	00002
C	IPNTSD(LPNTSD), ENSUBD(2,LSUBD)	SNWASH	00003
	COMMON /ENSUBD	SNWASH	00004
	COMMON /NWASHES/ IPNTDW(2,100),ENRUS(1275), ENRLS(1275),ICOLAPN	NWASHES	00002
	COMPLEX ENRUS, ENRLS	NWASHES	00003
	COMMON / KERN / ERR,MXSKRN,IPKERN,MLKRN,NSPATK,NROEA	KERN	00002
C	DELPHI(LMODES),TVP(LTVP),TEXLOC(LTVP)	DELTA	00002
	COMMON /DELTA/ DELPHI(1080), TVP(250), TEXLOC(250), FEXLOC(250),	DELTA	00003

1	IPNTRH(2,100),NPNTRS, IOMLAP	DELTA	00004
	COMPLEX DELPHI, TVP	DELTA	00005
C	DEFSL(2,LMODES)	DELTA	00006
	DIMENSION DEFSL(2,1000)	DELTA	00007
	EQUIVALENCE (DELPHI(61), DEFSL)	DELTA	00008
C	ARRAYS DELPHI AND DEFSL ARE	DELTA	00009
C	EQUIVALENCE TO GIVE A 2 ROW UN-OVERLAPPED SECTION	DELTA	00010
	COMMON /LROT / LROT	B	00050
C		B	00051
C		B	00052
	COMPLEX KERNL(1),SKERNL(1)	B	00053
	LOGICAL WING,DIM	B	00054
C		B	00055
	DIMENSION IXCD(150)	B	00056
	LOGICAL LEFT, LSIDE	B	00057
C		B	00058
	IA = IROW	B	00059
	B = ID.,0.)	B	00060
C		B	00061
	IF (WING) GO TO 20	B	00062
C		B	00063
	IXB = IXBT	B	00064
	IXBS = IXBST	B	00065
	MXB = MXBT - (IXBT-IXBW)/NSUBDV	B	00066
	MYBBS = MYBBST	B	00067
	IOMLAPN = IOMLAPN	B	00068
	SYMTY = SYMT	B	00069
	GO TO 25	B	00070
20	CONTINUE	B	00071
	IXB = IXBW	B	00072
	IXBS = 1	B	00073
	MXB = MXBBW	B	00074
	IF (COPLAN) MXB = MXBT	B	00075
	MYBBS = MYBBBW	B	00076
	IOMLAPN = 0	B	00077
	SYMTY = SYM	B	00078
25	CONTINUE	B	00079
	LSIDE = SYMTY .NE. 0 .AND. .NOT. DIM	B	00080
C		B	00081
C	IS SUBDIVISION REQUESTED -	B	00082
	IF (NSUBDV .EQ. 1) GO TO 410	B	00083
C		B	00084
	YES. DETERMINE THE NUMBER OF ROWS WHICH CAN BE HANDLED (MXB1)	B	00085
	NSRWMI = IPNTIN - IPNTOT - 1	B	00086
	IF (NSRWMI .LT. 0) NSRWMI = NSRWMI + IPNTLS	B	00087
	NSRWMI = MIN0(NSRWMI,MXSKRN-1)	B	00088
	IF (IA - IXBS .GT. NSRWMI) GO TO 120	B	00089
C	ALL SUBDIVIDED. ALLOW TO GO ONE BEYOND TO TRIGGER RETURN.	B	00090
	MXB1 = IA - IXBS + 1	B	00091
	GO TO 200	B	00092
C	PARTIAL SUBDIVIDED	B	00093
120	CONTINUE	B	00094
	I = IA - NSRWMI - IXB	B	00095
	I = MOD(I,NSUBDV)	B	00096
	I = LROT - I	B	00097
	MXB1 = NSRWMI - MOD(I,NSUBDV)	B	00098
C		B	00099

C	DETERMINE THE CONTRIBUTION TO B FROM A FORWARD CONE OF SUB-	B	00100
C	DIVIDED BOXES	B	00101
C	MU = ROW NUMBER OF CONTRIBUTING BOX RELATIVE TO RECEIVING	B	00102
C	BOX. MU OF RECEIVING BOX = 0.	B	00103
C	IA = ACTUAL ROW LOCATION OF CONTRIBUTING BOX, RELATIVE TO	B	00104
C	SUBDIVIDED GRID.	B	00105
C		B	00106
	200 CONTINUE	B	00107
	IF (MXB1 .EQ. 0) GO TO 310	B	00108
	DO 300 MU = 1, MXB1	B	00109
	IA = IA - 1	B	00110
C	HAS THE FORWARD EDGE OF THE PATTERN BEEN REACHED -	B	00111
	IF (IA .LT. IXBS) GO TO 600	B	00112
C	NO. GET BOX TYPE CODES FOR CURRENT ROW.	B	00113
	LEFT = .T.	B	00114
	IIA = MOD(IA-1, IPNTLS) + 1	B	00115
	IIAPI = MOD(IA, IPNTLS) + 1	B	00116
	IPNTP1 = IPNTSD(1, IIAPI)	B	00117
	IF (IPNTP1 .LE. 1) GO TO 208	B	00118
	NYB = IPNTP1 - IPNTSD(1, IIA) + IPNTSD(2, IIA) - 1	B	00119
	GO TO 210	B	00120
	208 NYB = MYBBS	B	00121
	210 CONTINUE	B	00122
	CALL DDCOR (IBOX, LBXCD, IA, 1, IA, NYB, .T., IBACD)	B	00123
	IF (NYB .MYBBS) GO TO 215	B	00124
	DO 212 I = 1, MYBBS	B	00125
	IF (IBXCD(NYB) .NE. 0) GO TO 215	B	00126
	NYB = NYB - 1	B	00127
	212 CONTINUE	B	00128
	215 CONTINUE	B	00129
C	IBXCD = ROW OF BOX CODES	B	00130
C	NYB = NUMBER FOUND	B	00131
C		B	00132
C	GET LOCATION IN THE SUBDIVIDED DOWNWASH ARRAY FOR BOX(IA, JCOL)	B	00133
	IDW = LOCSDW(IA, JCOL, IPNTSD, IPNTIN, IPNTOT, IPNTLS)	B	00134
	N = (NUM*(NUM+1))/2 + 1	B	00135
C	KERNEL(NU, MU) = SKERNL((NUM*(NUM+1))/2 + ABS(MU) + 1), SO	B	00136
C	N = SUBSCRIPT FOR KERNEL (STARTING WITH NU, 0)	B	00137
C		B	00138
C	CENTER BOX OF ROW IN CONE	B	00139
	IF (JCOL .GT. NYB) GO TO 220	B	00140
	IF (IBXCD(JCOL) .EQ. 0) GO TO 220	B	00141
	B = B + SKERNL(N) * (ENSUBD(1, IDW) - ENSUBD(2, IDW))	B	00142
	220 CONTINUE	B	00143
C		B	00144
C		B	00145
C	GOING OUT FROM CENTER CHORD OF CONE IN BOTH DIRECTIONS	B	00146
C	IDWR, IDWL = POINTERS IN DOWNWASH ARRAY FOR RIGHT, LEFT SIDES	B	00147
C	IBXR, IBXL = POINTERS IN BOX CODES ARRAY, AS ABOVE	B	00148
C	L = LEFT SIDE POINTER INCREMENTER (CHANGES SIGN WHEN	B	00149
C	THE PLATFORM CENTER-LINE IS ENCOUNTERED)	B	00150
C	E = LEFT SIDE MULTIPLIER, USED TO DETERMINE SYM/ANTI	B	00151
C	SYN. AFTER PLATFORM CENTER-LINE ENCOUNTERED.	B	00152
	IDWR = IDW+1	B	00153
	IDWL = IDW-1	B	00154
	IBXR = JCOL+1	B	00155
	IBXL = IBXR-2	B	00156

N = N + 1	B	00157
L = -1	B	00158
E = 1.0	B	00159
C	B	00160
DO 280 MU = 1, NU	B	00161
C	B	00162
RIGHT SIDE	B	00163
IF (IBXR .GT. NYB) GO TO 230	B	00164
IF (IBXCD(IBXR) .EQ. 0) GO TO 230	B	00165
B = B + SKERNL(N) * (ENSUBD(1, IDWR) - ENSUBD(2, IDWR))	B	00166
230 CONTINUE	B	00167
C	B	00168
LEFT SIDE	B	00169
C	B	00170
IF (.NOT. LEFT) GO TO 270	B	00171
HAS PLATFORM CENTER LINE BEEN ENCOUNTERED -	B	00172
IF (IBXL .GT. 0) GO TO 250	B	00173
C	B	00174
YES. SET PARAMETERS TO SWEEP BACK ACROSS RIGHT HALF AS A	B	00175
C	B	00176
SYMMETRIC/ANTISYMMETRIC IMAGE OF THE LEFT SIDE.	B	00177
LEFT = LSIDE	B	00178
IF (.NOT. LEFT) GO TO 270	B	00179
E = SYMTY	B	00180
L = 1	B	00181
IBXL = 1	B	00182
IDWL = IDWL + 1	B	00183
GO TO 260	B	00184
C	B	00185
IF THE CENTER LINE HAS PREVIOUSLY BEEN ENCOUNTERED, IBXL WILL	B	00186
C	B	00187
BE INCREASING. IF IBXL HAS EXCEEDED THE NUMBER OF BOXES ON	B	00188
C	B	00189
THIS ROW, THIS ROW IS COMPLETE, TRANSFER TO LOOP ON NU.	B	00190
250 IF (IBXL .GT. NYB) GO TO 270	B	00191
260 IF (IBXCD(IBXL) .EQ. 0) GO TO 270	B	00192
B = B + SKERNL(N) * (ENSUBD(1, IDWL) - ENSUBD(2, IDWL)) * E	B	00193
270 CONTINUE	B	00194
C	B	00195
SET COUNTERS FOR NEXT STEP OUTWARD	B	00196
IDWR = IDWR + 1	B	00197
IDWL = IDWL + L	B	00198
IBXR = IBXR + 1	B	00199
IBXL = IBXL + L	B	00200
N = N + 1	B	00201
280 CONTINUE	B	00202
C	B	00203
END OF LOOP ON MU (SUBDIVIDED COLUMNS OUTBOARD)	B	00204
C	B	00205
300 CONTINUE	B	00206
END OF LOOP ON NU (SUBDIVIDED ROWS FORWARD) FROM 200	B	00207
C	B	00208
IS THERE AT LEAST ONE FULL UNSUBDIVIDED ROW LEFT AHEAD OF	B	00209
C	B	00210
CURRENT POSITION -	B	00211
310 CONTINUE	B	00212
IF (IA .LT. IXB) GO TO 800	B	00213
C	B	
C	B	
C	B	
UNSUBDIVIDED BOXES	B	
C	B	
C	B	
DETERMINE ROW AND COLUMN NUMBERS IN SUBDIVIDED ARRAYS CORRES-	B	
C	B	
PONDING TO UNSUBDIVIDED BOX CENTERS.	B	
C	B	
IA = ROW LOCATION OF CONTRIBUTING SUBDIVIDED BOX	B	

C	IIA = ROW LOCATION OF UNSUBDIVIDED BOX	B	00214
C	INU = FIRST ROW OF UNSUBDIVIDED BOXES TO USE, COUNTING	B	00215
C	OUTWARD.	B	00216
C	JJJ = UNSUBDIVIDED CHORD NUMBER OF RECEIVING BOX	B	00217
C		B	00218
	IA = IA + NSUBD2	B	00219
	IIA = (IA - IXBW)/NSUBDV + 1	B	00220
	INU = MYB1 /NSUBDV + 1	B	00221
	JJJ = (JCQL-1)/NSUBDV + 1	B	00222
	GO TO 420	B	00223
C		B	00224
C	SET UP POINTERS IF NO SUBDIVISION WAS REQUESTED	B	00225
C		B	00226
	410 CONTINUE	B	00227
C	IA WAS SET TO IROW UPON ENTRY	B	00228
	IIA = IA	B	00229
	INU = 1	B	00230
	JJJ=JCQL	B	00231
C		B	00232
C	DETERMINE THE CONTRIBUTION TO B FROM A FORWARD CONE OF UNSUB-	B	00233
C	DIVIDED BOXES, STARTING WHERE SUBDIVISION LEFT OFF.	B	00234
C	NU = ROW NUMBER OF CONTRIBUTING BOX RELATIVE TO RECEIVING	B	00235
C	BOX.	B	00236
C		B	00237
	420 DO 500 NU = INU,MYB	B	00238
	IA = IA - NSUBDV	B	00239
	IIA = IIA -1	B	00240
C	HAS THE FORWARD EDGE OF THE PATTERN BEEN REACHED -	B	00241
	IF (IIA .LT. IXB) GO TO 600	B	00242
C	NO. GET BOX TYPE CODES FOR CURRENT ROW, UNSUBDIVIDED BOX	B	00243
C	CENTERS ONLY	B	00244
	LEFT = .T.	B	00245
	IIAVLPM = IIA + IOVLPM	B	00246
	MYB = IPNTDW(1,IIAVLPM+1) - IPNTDW(1,IIAVLPM) + IPNTDW(2,IIAVLPM)	B	00247
	1 - 1	B	00248
	CALL DCDER(IBOX,LBXCD, IIA,1, IIA,MYB, .F., IBXCD)	B	00249
C	IBXCD = ROW OF BOX CODES	B	00250
C	MYB = NUMBER FOUND	B	00251
C		B	00252
C	GET LOCATION IN UNSUBDIVIDED DOWNWASH ARRAY FOR BOX(IIA,JJJ)	B	00253
	IDW = LOCSDW(IIAVLPM,JJJ, IPNTDW, IIAVLPM+1,1,IIAVLPM+2)	B	00254
C		B	00255
	N = (NU*(NU+1))/2 + 1	B	00256
C	N = UNSUBDIVIDED KERNEL SUBSCRIPT FOR MU = 0.	B	00257
C		B	00258
C	CENTER BOX	B	00259
	IF (JJJ .GT. MYB) GO TO 425	B	00260
	IF (IBXCD(JJJ) .NE. 0) B = B + PKERN(N)*(ENRUS(IDW)-EDRLS(IDW))	B	00261
	425 CONTINUE	B	00262
C		B	00263
C	GOING OUT FROM CENTER CHORD IN BOTH DIRECTIONS	B	00264
	IDMR = IDW+1	B	00265
	IDML = IDW-1	B	00266
	IDMR = JJJ+1	B	00267
	IDML = IDMR-2	B	00268
	L = -1	B	00269
	E = 1.0	B	00270

N = N + 1	B	00271
C	B	00272
DO 480 MU = 1, MU	B	00273
C	B	00274
C	B	00275
RIGHT SIDE	B	00276
IF (IBXR .GT. MYB) GO TO 430	B	00277
IF (IBXCD(IBXR) .EQ. 0) GO TO 430	B	00278
B = B + PKERNL(N) * (ENRUS(IDWR) - ENRLS(IDWR))	B	00279
430 CONTINUE	B	00280
C	B	00281
C	B	00282
LEFT SIDE	B	00283
IF (.NOT. LEFT) GO TO 470	B	00284
C	B	00285
HAS PLAFORM CENTER LINE BEEN ENCOUNTERED -	B	00286
IF (IBXL .GT. 0) GO TO 450	B	00287
C	B	00288
YES. SET PARAMETERS TO SWEEP BACK ACROSS RIGHT SIDE	B	00289
LEFT = LSIDE	B	00290
IF (.NOT. LEFT) GO TO 470	B	00291
E = SYMTY	B	00292
L = 1	B	00293
IBXL = 1	B	00294
IDWL = IDWL + 1	B	00295
C	B	00296
TEST FOR ROW COMPLETE, AS IN SUBDIVIDED LOGIC	B	00297
450 IF (IBXL .GT. MYB) GO TO 470	B	00298
460 IF (IBXCD(IBXL) .EQ. 0) GO TO 470	B	00299
B = B + PKERNL(N) * (ENRUS(IDWL) - ENRLS(IDWL)) * E	B	00300
470 CONTINUE	B	00301
C	B	00302
SET COUNTERS FOR NEXT STEP OUTWARD	B	00303
IDWR = IDWR+1	B	00304
IDWL = IDWL+L	B	00305
IBXR = IBXR+1	B	00306
IBXL = IBXL+L	B	00307
N = N + 1	B	00308
480 CONTINUE	B	00309
C	B	00310
END OF LOOP ON MU (CHORDS OUTWARD)	B	00311
C	B	00312
500 CONTINUE	B	00313
C		
END OF LOOP ON MU (ROWS FORWARD) FROM 420		
C		
600 RETURN		
C		
END		

	SUBROUTINE GETAIC(JUCENT, ITPE, ICODE, IR)	GETAIC 00002
C		GETAIC 00003
C	GETS DESIRED AIC ARRAYS FROM DISK	GETAIC 00004
C		GETAIC 00005
C	JUCENT = CHORD NUMBER	GETAIC 00006
C	ITPE = 1, WING/WING 3, RIGHT WING/TAIL	GETAIC 00007
C	2, TAIL/TAIL 4, LEFT WING/TAIL	GETAIC 00008
C	ICODE = 0, C,V,W DESIRED	GETAIC 00009
C	1, V,W DESIRED	GETAIC 00010
C	2, W DESIRED	GETAIC 00011
C	IR = ERROR RETURN 0, SUCCESS 2, C,W NOT FOUND	GETAIC 00012
C	1, C NOT FOUND 3, NOTHING FOUND	GETAIC 00013
C		GETAIC 00014
	COMMON /FILES / NT5,NT6,INTAPE,INFSP,NFLAIC,NSPAIC,NOUP,	FILES 00002
1	IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC	FILES 00003
	COMMON /TAPEIO/ NFS,NMS,LS,NMR,ID(20),NID,ITYPE,LRS,LWS,M,N,	TAPEIO 00002
1	PARM(10),IRR	TAPEIO 00003
	DIMENSION IPARM(10)	TAPEIO 00004
	EQUIVALENCE (PARM,IPARM)	TAPEIO 00005
	COMMON /MUAICS/ YBAR,EL,MUAIC(2,50),NROWS,SURF,	MUAICS 00002
1	YBARL,ELL, MUAICL(2,50),NROWSL,SURFL,PSIDIF	MUAICS 00003
	LOGICAL SURF,SURFL	MUAICS 00004
	COMMON /PAICS / NNAK, NNTK, NRWK, NLWK, PAIC(4,50)	PAICS 00002
	INTEGER PAIC	PAICS 00003
	DIMENSION NK(4)	PAICS 00004
	EQUIVALENCE (NNAK,NK(1))	PAICS 00005
	COMMON /AICS / XKVL, C(1640),W(1640),V(1640)	AICS 00002
	COMPLEX C, W, V	AICS 00003
	COMMON /ARRAYS/ KBXCDW,LBXCDW,LBOXC,KBXCDT,LBXCDT,KJALPH,LJALPH,	ARRAYS 00002
1	KALPHA,KKERNL,LKERNL,KPNTRM,LPNTRM,KDEFSL,KELPHI,	ARRAYS 00003
2	LMODES,KPNTSD,LPTSD,KSDW,LSDW,KPNTDW,LPTDW,	ARRAYS 00004
3	KDW,LDW,KTVP,LTPV	ARRAYS 00005
	LOGICAL MXWRIT,RANDOU	GETAIC 00022
	DATA MXWRIT,RANDOU / .F.,.F. /	GETAIC 00023
	DATA IPAIC,IPAICL,IPNT, IFLAG,IFLAGL, ICCDEP / 640 /	FTNXL 00061
C		GETAIC 00024
	IR = 0	GETAIC 00025
	IF (IPNT .NE. 0) GO TO 100	GETAIC 00026
C		GETAIC 00027
C	INITIAL CALL. SET UP FILES AND POINTERS	GETAIC 00028
	REWINO IAICSC	GETAIC 00029
	IPNT = 1	GETAIC 00030
C	EXPAND PAIC ARRAY	GETAIC 00031
	I = 4	GETAIC 00032
	IF (NRWK .EQ. 0) I = 3	GETAIC 00033
	IF (NNTK .EQ. 0) I = I - 1	GETAIC 00034
	IF (NNAK .EQ. 0) I = I - 1	GETAIC 00035
	IF (NLWK .EQ. 0) GO TO 120	GETAIC 00036
	IF (I .EQ. 4) GO TO 140	GETAIC 00037
	DO 110 J = 1,NLWK	GETAIC 00038
	PAIC(4,J) = PAIC(I,J)	GETAIC 00039
	PAIC(I,J) = 0	GETAIC 00040
110	CONTINUE	GETAIC 00041
	I = I - 1	GETAIC 00042
120	CONTINUE	GETAIC 00043
	IF (NRWK .EQ. 0) GO TO 130	GETAIC 00044
	IF (I .EQ. 3) GO TO 140	GETAIC 00045

DO 125 J = 1, NMTK	GETAIC 00046
PAIC(3,J) = PAIC(1,J)	GETAIC 00047
PAIC(1,J) = 0	GETAIC 00048
125 CONTINUE	GETAIC 00049
I = I - 1	GETAIC 00050
130 CONTINUE	GETAIC 00051
IF (NMTK .EQ. 0) GO TO 140	GETAIC 00052
IF (I .EQ. 2) GO TO 140	GETAIC 00053
DO 135 J = 1, NMTK	GETAIC 00054
PAIC(2,J) = PAIC(1,J)	GETAIC 00055
PAIC(1,J) = 0	GETAIC 00056
135 CONTINUE	GETAIC 00057
140 CONTINUE	GETAIC 00058
C ZERO OUT THE AIC ARRAYS	GETAIC 00059
DO 150 I = 1, LKERNL	GETAIC 00060
C(I) = (0.,0.)	GETAIC 00061
W(I) = (0.,0.)	GETAIC 00062
V(I) = (0.,0.)	GETAIC 00063
150 CONTINUE	GETAIC 00064
C	GETAIC 00065
C GET THE AIC LOCATION	GETAIC 00066
160 CONTINUE	GETAIC 00067
IF (NMTK .LT. JUCENT) GO TO 290	GETAIC 00068
ILOC = PAIC(IJTYPE, JUCENT)	GETAIC 00069
C ARE THE DESIRED ARRAYS ALREADY IN CORE -	GETAIC 00070
IF (ILOC .EQ. IPAIC) GO TO 300	GETAIC 00071
IPAIC = ILOC	GETAIC 00072
ILOC = (ILOC-1)*4 + 1	GETAIC 00073
C	GETAIC 00074
C GET THE MUAIC ARRAY FROM THE NON-PLANAR AIC SCRATCH FILE	GETAIC 00075
C	GETAIC 00076
C SPACE AND READ MUAICS	GETAIC 00077
CALL RDINIT	GETAIC 00078
IF (ILOC - IPNT) 200,220,210	GETAIC 00079
C MUAICS ARE BEHIND CURRENT LOCATION	GETAIC 00080
200 REWIND IAICSC	GETAIC 00081
NMB = ILOC - 1	GETAIC 00082
GO TO 220	GETAIC 00083
C	GETAIC 00084
C REQUIRED MUAICS ARE AHEAD OF CURRENT POSITION	GETAIC 00085
210 CONTINUE	GETAIC 00086
NMB = ILOC - IPNT	GETAIC 00087
C	GETAIC 00088
C READ MUAICS FROM IAICSC	GETAIC 00089
220 CONTINUE	GETAIC 00090
IXARRY = 9H MUAIC	GETAIC 00091
K = 2	GETAIC 00092
CALL READMX(IAICSC, MXWRT, RANCOU, NFB, NMB, L8, NMR, K, MID, ID,	GETAIC 00093
1 IJTYPE, LRS, MUAIC, M,N, PARM, IRR)	GETAIC 00094
IF (IRR .NE. 0) GO TO 3000	GETAIC 00095
NROMB = N	GETAIC 00096
IPNT = ILOC + 1	GETAIC 00097
EL = PARM(5)	GETAIC 00098
YBAR = PARM(4)	GETAIC 00099
ICD = PARM(6)	GETAIC 00100
C	GETAIC 00101
CALL RDINIT	GETAIC 00102

C	IS THE C ARRAY DESIRED -	GETAIC	00103
	IF (ICODE .NE. 0) GO TO 250	GETAIC	00104
C	YES. IS IT AVAILABLE -	GETAIC	00105
	IF (ICD .EQ. 0) GO TO 240	GETAIC	00106
C	NO. SET THE ERROR FLAG AND CONTINUE	GETAIC	00107
	IR = 1	GETAIC	00108
	GO TO 250	GETAIC	00109
C	READ THE C ARRAY FROM IAICSC	GETAIC	00110
	240 CONTINUE	GETAIC	00111
	MXARRY = 9HSPATIAL C	GETAIC	00112
	CALL READMX(IAICSC, MXWRIT,RANDOU, NFS,NMS, LS,NMR, K, NID, ID,	GETAIC	00113
	1 ITYPE, LRS, C, M,N, PARM, IRR)	GETAIC	00114
	IF (IRR .NE. 0) GO TO 3000	GETAIC	00115
	CALL RDINIT	GETAIC	00116
	GO TO 260	GETAIC	00117
	250 CONTINUE	GETAIC	00118
	NMS = 1	GETAIC	00119
	260 CONTINUE	GETAIC	00120
	IPNT = IPNT + 1	GETAIC	00121
C	READ THE W ARRAY FROM IAICSC	GETAIC	00122
	MXARRY = 9HSPATIAL W	GETAIC	00123
	CALL READMX(IAICSC, MXWRIT,RANDOU, NFS,NMS, LS,NMR, K, NID, ID,	GETAIC	00124
	1 ITYPE, LRS, W, M,N, PARM, IRR)	GETAIC	00125
	IF (IRR .NE. 0) GO TO 3000	GETAIC	00126
	IPNT = IPNT + 1	GETAIC	00127
	CALL RDINIT	GETAIC	00128
C	IS THE V ARRAY DESIRED -	GETAIC	00129
	IF (ICODE .EQ. 2) GO TO 300	GETAIC	00130
C	YES. IS IT AVAILABLE -	GETAIC	00131
	IF (ICD .NE. 2) GO TO 280	GETAIC	00132
C	NO. SET ERROR FLAG	GETAIC	00133
	IR = IR + 1	GETAIC	00134
	GO TO 300	GETAIC	00135
C	READ THE V ARRAY FROM IAICSC	GETAIC	00136
	280 CONTINUE	GETAIC	00137
	MXARRY = 9HSPATIAL V	GETAIC	00138
	CALL READMX(IAICSC, MXWRIT,RANDOU, NFS,NMS, LS,NMR, K, NID, ID,	GETAIC	00139
	1 ITYPE, LRS, V, M,N, PARM, IRR)	GETAIC	00140
	IF (IRR .NE. 0) GO TO 3000	GETAIC	00141
	IPNT = IPNT + 1	GETAIC	00142
	GO TO 300	GETAIC	00143
C		GETAIC	00144
C	NO AICS CAN BE FOUND OF THE TYPE DESIRED FOR THIS CHORD	GETAIC	00145
	290 IR = 3	GETAIC	00146
C		GETAIC	00147
	300 CONTINUE	GETAIC	00148
	RETURN	GETAIC	00149
C		GETAIC	00150
C	DIAGNOSTIC	GETAIC	00151
C		GETAIC	00152
	3000 CONTINUE	GETAIC	00153
	WRITE (MT6,9000) IAICSC,IRR	GETAIC	00154
	WRITE (MT6,9192) MXARRY, N,N	GETAIC	00155
	CALL FLUSH(1)	GETAIC	00156
	9000 FORMAT(49H*** ERROR - WHILE READING FROM SPATIAL AIC FILE ,A10,	GETAIC	00157
	1 14H, ERROR CODE = 14, 4H ***)	GETAIC	00158
	9192 FORMAT(14X,A10, 20HARRAY, DIMENSIONED (14,1H,14,11H) WAS BEING	GETAIC	00159

1 34 READ 1
END

GETAIC 00160
GETAIC 00161

	SUBROUTINE STOSDW(IROW, JCQL, EN, IBOX, LBXCD, IXB, MXBS, MYB, IRR)	STOSDW 00002
C		STOSDW 00003
C	STORES A COMPUTED DOWNWASH VALUE IN THE END-AROUND SUBDIVIDED	STOSDW 00004
C	DOWNWASH ARRAY, AND UPDATES POINTERS WHEN NECESSARY	STOSDW 00005
C		STOSDW 00006
C	IROW = BOX CHORDWISE LOCATION	STOSDW 00007
C	JCQL = BOX SPANWISE LOCATION	STOSDW 00008
C	EN = COMPLEX NORMAL-WASHES TO BY STORED	STOSDW 00009
C	IBOX = ARRAY OF BOX CODES	STOSDW 00010
C	LBXCD = LENGTH OF BOX CODE ARRAY	STOSDW 00011
C	IXB = FIRST SUBDIVIDED ROW OF THE PLANFORM	STOSDW 00012
C	MXBS = MAXIMUM CHORD LENGTH OF SUBDIVIDED PATTERN	STOSDW 00013
C	MYB = MAXIMUM ROW LENGTH	STOSDW 00014
C	RETURNS -	STOSDW 00015
C	IRR = ERROR RETURN, 0 = SUCCESSFUL	STOSDW 00016
C	= 1, FUNCTION LOCSDW FOUND THE POINTER OUTSIDE	STOSDW 00017
C	THE DEFINED SET OF DOWNWASHES	STOSDW 00018
C	ENSUBD = SUBDIVIDED NORMAL-WASH ARRAY WITH ADDED VALUE	STOSDW 00019
C		STOSDW 00020
C	COMMON PARAMETERS USED	STOSDW 00021
C	MXSKRN = MAXIMUM SIZE OF THE SUBDIVIDED KERNEL	STOSDW 00022
C	LSDW = DIMENSION OF SUBDIVIDED NORMAL WASH ARRAY	STOSDW 00023
C	IPNTSD = POINTER ARRAY FOR SUBDIVIDED NORMAL WASH ARRAY	STOSDW 00024
C	IPNTIN = NEXT AVAILABLE CELL IN IPNTSD	STOSDW 00025
C	IPNTOT = FIRST CURRENTLY VALID CELL IN IPNTSD	STOSDW 00026
C	IPNTLS = DIMENSION OF IPNTSD	STOSDW 00027
C		STOSDW 00028
	COMMON / KERN / ERR, MXSKRN, IPKERN, NFKERN, NEFATK, NRCWEA	KERN 00002
	COMMON /ARRAYS/ KBXCDW, LBXCDW, LBOXC, KBXCDT, LBXCDT, KJALPH, LJALPH,	ARRAYS 00002
1	KALPHA, KKERNL, LKERNL, KPNTRM, LPNTRY, KDEFSL, KELPHI,	ARRAYS 00003
2	LMODES, KPNTSD, LPNTSD, KSDW, LSDW, KPNTDW, LPNTDW,	ARRAYS 00004
3	KDW, LDW, KTVF, LTVF	ARRAYS 00005
	COMMON /SNWASH/ IPNTSD(2,50), ENSUBD(2,600), IPNTIN, IPNTOT, IPNTLS	SNWASH 00002
C	IPNTSD(LPNTSD), ENSUBD(2*LSDW)	SNWASH 00003
	COMPLEX ENSUBD	SNWASH 00004
	COMMON /CHECKFR/ DPPCFR, GEOCFR, MODCFR, AICCFR, NMSCFR, SMCFR, GAFCFR	CHECKFR 00002
	LOGICAL DPPCFR, GEOCFR, MODCFR, AICCFR, NMSCFR, SMCFR, GAFCFR	CHECKFR 00003
	EQUIVALENCE (CHECKFR, NMSCFR)	STOSDW 00033
	LOGICAL CHECKFR	STOSDW 00034
	DIMENSION TITL(3)	STOSDW 00035
C		STOSDW 00036
C		STOSDW 00037
	COMPLEX EN(2)	STOSDW 00038
C		STOSDW 00039
	IRR = 0	STOSDW 00040
C	IS THIS THE INITIAL CALL -	STOSDW 00041
	IF (IROW.EQ. IXB .AND. JCQL.EQ. 1) GO TO 700	STOSDW 00042
C	NO. IS A NEW ROW BEING CONSIDERED -	STOSDW 00043
	IF (IROW.GT. IOROW) GO TO 200	STOSDW 00044
C	NO. GET THE LOCATION FOR THE VALUE IN THE SUBDIVIDED DOWN-	STOSDW 00045
C	WASH ARRAY	STOSDW 00046
	IJ = LOCSDW(IROW, JCQL, IPNTSD, IPNTIN, IPNTOT, IPNTLS)	STOSDW 00047
	IF (IJ) 900,900,550	STOSDW 00048
C		STOSDW 00049
C	MUST UPDATE POINTERS AND ADD A ROW TO THE SUBDIVIDED BOX ARRAY	STOSDW 00050
	200 CONTINUE	STOSDW 00051
	IOROW = IROW	STOSDW 00052

C	SET THE NEXT VALUES OF THE POINTER ARRAY (1 ROW)	STOSDW	00053
C	INCREMENT IPNTIN, ALLOWING FOR END-AROUND INCREMENTAL	STOSDW	00054
	INM1 = IPNTIN	STOSDW	00055
	I1 = IPNTSD(1,IPNTIN)	STOSDW	00056
	CALL POINTR (IROW, 1,MYB, .T...T., IBOX,LBXCD, IPNTLS,	STOSDW	00057
1	I1, IPNTIN,IPNTSD)	STOSDW	00058
C		STOSDW	00059
C	LOOP INCREMENTING IPNTOT, IF OVER-LAP OCCURS.	STOSDW	00060
220	CONTINUE	STOSDW	00061
	IF (IPNTIN-IPNTOT) 230,225,235	STOSDW	00062
225	IPNTOT = MOD(IPNTOT,IPNTLS) + 1	STOSDW	00063
	GO TO 220	STOSDW	00064
230	ISROWS = IPNTIN-IPNTOT+IPNTLS	STOSDW	00065
	GO TO 240	STOSDW	00066
235	ISROWS = IPNTIN-IPNTOT	STOSDW	00067
240	CONTINUE	STOSDW	00068
	IF (ISROWS .GT. MXSKRN) IPNTOT = MOD(IPNTOT+ISROWS-MXSKRN-1,	STOSDW	00069
1	IPNTLS) + 1	STOSDW	00070
C		STOSDW	00071
C	KEEP SUBDIVIDED DOWNWASHES END-AROUND.	STOSDW	00072
C	HAS THE ARRAY LIMIT BEEN EXCEEDED -	STOSDW	00073
	IOLD = IPNTSD(1,INM1)	STOSDW	00074
	IF (IPNTSD(1,IPNTIN) .LE. LSNW + 1) GO TO 405	STOSDW	00075
C	LIMIT EXCEEDED BY CURRENT ROW. PLACE AT BEGINNING OF THE ARRAY	STOSDW	00076
	IF (.NOT. CHECKPR) GO TO 400	STOSDW	00077
	TITL(1) = 10HEN SUBDIVI	STOSDW	00078
	TITL(2) = 10HDED, UPPER	STOSDW	00079
	TITL(3) = 10H, PARTIAL	STOSDW	00080
	IF (IPNTIN .LT. IPNTOT) GO TO 395	STOSDW	00081
	CALL PRINTR(TITL,0,ENSUBD,2,IPNTOT,IPNTIN-1,MYB,IPNTSD)	STOSDW	00082
	GO TO 400	STOSDW	00083
395	CALL PRINTR(TITL,0,ENSUBD,2,IPNTOT,IPNTLS-1, MYB,IPNTSD)	STOSDW	00084
	CALL PRINTR(TITL,0,ENSUBD,2, 1, IPNTIN-1, MYB,IPNTSD)	STOSDW	00085
400	CONTINUE	STOSDW	00086
	IPNTSD(1,IPNTIN) = IPNTSD(1,IPNTIN) - IOLD + 1	STOSDW	00087
	IPNTSD(1,INM1) = 1	STOSDW	00088
C		STOSDW	00089
405	CONTINUE	STOSDW	00090
	IPM = IPNTSD(1,INM1)	STOSDW	00091
	IP1 = IPNTSD(1,IPNTIN)	STOSDW	00092
	IPO = IPNTSD(1,IPNTOT)	STOSDW	00093
C		STOSDW	00094
	IF (IOLD .GT. IPO) GO TO 430	STOSDW	00095
C		STOSDW	00096
C	ARRAY WAS ALREADY END-AROUND PRIOR TO LATEST ADDITION	STOSDW	00097
	IF (IOLD .EQ. IPM) GO TO 440	STOSDW	00098
C	ADDED ROW WENT END-AROUND AS WELL	STOSDW	00099
410	IPNTOT = MOD(IPNTOT,IPNTLS) + 1	STOSDW	00100
	IF (IPNTSD(1,IPNTOT) .NE. 1) GO TO 410	STOSDW	00101
	IPNTOT = MOD(IPNTOT,IPNTLS) + 1	STOSDW	00102
	IPO = IPNTSD(1,IPNTOT)	STOSDW	00103
	GO TO 440	STOSDW	00104
C		STOSDW	00105
C	ARRAY WAS SEQUENTIAL. CHECK WHETHER IT HAS GONE END-AROUND	STOSDW	00106
430	CONTINUE	STOSDW	00107
	IF (IOLD .EQ. IPM) GO TO 500	STOSDW	00108
C	IT HAS GONE END-AROUND	STOSDW	00109

C		STOSDW	00110
	440 CONTINUE	STOSDW	00111
C	HAS THE ARRAY BEEN OVER-WRITTEN IN GOING END-AROUND -	STOSDW	00112
	IF (IPO .GT. IPI) GO TO 500	STOSDW	00113
C	YES. MOVE IPNTOT UNTIL CLEAR.	STOSDW	00114
	IPNTOT = MOD(IPNTOT,IPNTLS) + 1	STOSDW	00115
	IPO = IPNTSD(1,IPNTOT)	STOSDW	00116
	IF (IPO .NE. 1) GO TO 440	STOSDW	00117
C		STOSDW	00118
C	ALL POINTERS HAVE BEEN RESET. GET LOCATION	STOSDW	00119
	500 CONTINUE	STOSDW	00120
	IJ = LOCSDW(IROW,ICOL, IPNTSD,IPNTIN,IPNTOT,IPNTLS)	STOSDW	00121
	IF (IJ .EQ. 0) GO TO 900	STOSDW	00122
C		STOSDW	00123
C	STORE THE DOWNWASH VALUE	STOSDW	00124
	550 CONTINUE	STOSDW	00125
	ENSUBD(1,IJ) = EN(1)	STOSDW	00126
	ENSUBD(2,IJ) = EN(2)	STOSDW	00127
	600 RETURN	STOSDW	00128
C		STOSDW	00129
C		STOSDW	00130
C	INITIAL CALL	STOSDW	00131
	700 CONTINUE	STOSDW	00132
	IPNTOT = MOD(IXB-1,IPNTLS) + 1	STOSDW	00133
	IPNTIN = IPNTOT	STOSDW	00134
	MX = MIND(MXBS-IXB+1, MXSKRN, IPNTLS-1)	STOSDW	00135
C	SET UP POINTER ARRAY FOR FIRST PASS	STOSDW	00136
	CALL POINTR(IXB, MX,MYB, .T., .T., IBOX,LBXCD, IPNTLS,	STOSDW	00137
	1 1, IPNTIN, IPNTSD)	STOSDW	00138
	720 CONTINUE	STOSDW	00139
	IF (IPNTSD(1,IPNTIN) .LE. LSDW) GO TO 730	STOSDW	00140
	IPNTIN = IPNTIN - 1	STOSDW	00141
	MX = MX - 1	STOSDW	00142
	GO TO 720	STOSDW	00143
	730 CONTINUE	STOSDW	00144
	IROW = MX + IXB - 1	STOSDW	00145
C		STOSDW	00146
	ENSUBD(1,1) = EN(1)	STOSDW	00147
	ENSUBD(2,1) = EN(2)	STOSDW	00148
	GO TO 600	STOSDW	00149
C		STOSDW	00150
C	ERROR	STOSDW	00151
	900 IRR = 1	STOSDW	00152
	GO TO 600	STOSDW	00153
C		STOSDW	00154
	END	STOSDW	00155

	SUBROUTINE PRINTR(TITL,IMODE,ARRAY,K,IXB,MYB,IPNTRM)	PRNTNW 00002
C		PRNTNW 00003
C	TITL - TITLE TO PRINT FOR THE ARRAY	PRNTNW 00004
C	IMODE - MODE SHAPE NUMBER	PRNTNW 00005
C	ARRAY - ARRAY TO BE PRINTED	PRNTNW 00006
C		PRNTNW 00007
	DIMENSION ARRAY(K,1), TITL(3)	PRNTNW 00008
	COMPLEX ARRAY	PRNTNW 00009
	DIMENSION IPNTRM(2,50)	PRNTNW 00010
	COMMON /CONTRL/ PRVEX,OMACH, TITL(8), PRVGEOM,PRVMODE,DIHW,DIHT,	CONTRL 00002
1	DEFAULT	CONTRL 00003
	LOGICAL PRVGEOM,PRVMODE,DIHW,DIHT,DEFAULT	CONTRL 00004
	COMMON /PROBLM/ XMACH,NMODES,NTSLOP,NKVALS,SMOOTH,NDEC,CRDFIT,	PROBLM 00002
1	EXAIC,SUBDV,PLYWOOD	PROBLM 00003
	LOGICAL SMOOTH,CRDFIT,EXAIC,SUBDV,PLYWOOD	PROBLM 00004
	COMMON /FILES / NT5,NT6,INTAPE,INFSF,NPLAIC,NSPAIC,NOUTP,	FILES 00002
1	IQUFSP,MODESC,IVFSC,IGEOSC,IWTFSC,IAICSC	FILES 00003
	COMMON /KVAL / IKVAL,XXVAL(20), XKS(20)	KVAL 00002
	DIMENSION PC(2)	PRNTNW 00016
	DIMENSION S(50),D(50)	PRNTNW 00020
	EQUIVALENCE (S(1),BUFF(1)), (D(1),BUFF(1251))	PRNTNW 00021
	REAL K1	PRNTNW 00022
	INTEGER PAGE	PRNTNW 00023
	COMMON /RWBUFF/ BFCODE,IBFCNT, BUFF(3280)	RWBUFF 00002
	DATA PC / 10HFACE CONTI,4HNDED /	FTNDX 00063
	DATA BLANK / 1H /	FTNDX 00064
	DATA XINIT / -1.0 /	FTNDX 00065
	K1 = XXVAL(IKVAL)	PRNTNW 00024
	IF(XKS(IKVAL).NE.XINIT) K1 = XKS(IKVAL)	PRNTNW 00025
C		PRNTNW 00026
C		PRNTNW 00027
	PAGE = 0	PRNTNW 00028
	N = 1	PRNTNW 00029
	M = 4	PRNTNW 00030
	IF(M.GT.MYB) M = MYB	PRNTNW 00031
100	LINE = 100	PRNTNW 00032
200	DO 1400 I = IXB,MYB	PRNTNW 00033
	DO 300 J=N,M	PRNTNW 00034
	S(J) = 0.0	PRNTNW 00035
	D(J) = 0.0	PRNTNW 00036
300	CONTINUE	PRNTNW 00037
	IF(LINE.LE.50) GO TO 900	PRNTNW 00038
	PAGE = PAGE + 1	PRNTNW 00039
	LINE = 4	PRNTNW 00040
	WRITE (NT6,9001) TITL,TITL, XMACH, K1, IMODE	PRNTNW 00041
C		PRNTNW 00042
	IF(PAGE.EQ.1) GO TO 700	PRNTNW 00043
	WRITE (NT6,9005) PC	PRNTNW 00044
	GO TO 600	PRNTNW 00045
700	WRITE(NT6,9005)	PRNTNW 00046
800	CONTINUE	PRNTNW 00047
	WRITE(NT6,6006) (BLANK,J,J=N,M)	PRNTNW 00048
	WRITE(NT6,6007) (BLANK, J=N,M)	PRNTNW 00049
900	CONTINUE	PRNTNW 00050
	JS = IPNTRM(2,I)	PRNTNW 00051
	IF (JS LE. 0) GO TO 1400	PRNTNW 00052
	IDX = IPNTRM(1,I)	PRNTNW 00053

JE = IPNTRM(1,I+1) - IDX + JS -1	PRNTNW 00054
IF (JE.EQ.0) GO TO 1400	PRNTNW 00055
DO 1000 J=JS,JE	PRNTNW 00056
S(J) = REAL (ARRAY(1,IDX))	PRNTNW 00057
D(J) = AIMAG (ARRAY(1,IDX))	PRNTNW 00058
IDX = IDX +1	PRNTNW 00059
1000 CONTINUE	PRNTNW 00060
DO 1200 J=N,M	PRNTNW 00061
IF (S(J)) 1300,1100,1300	PRNTNW 00062
1100 CONTINUE	PRNTNW 00063
IF (D(J)) 1300,1200,1300	PRNTNW 00064
1200 CONTINUE	PRNTNW 00065
GO TO 1400	PRNTNW 00066
1300 WRITE (N16,9013) I, (S(J),D(J),J=N,M)	PRNTNW 00067
LINE = LINE + 1	PRNTNW 00068
1400 CONTINUE	PRNTNW 00069
M = M+4	PRNTNW 00070
N = N+4	PRNTNW 00071
IF (N.GT.MYB) GO TO 1500	PRNTNW 00072
IF (M.GT.MYB) M=MYB	PRNTNW 00073
IF (LINE.GT.45) GO TO 100	PRNTNW 00074
WRITE (N16,6006) (BLANK,J,J=N,M)	PRNTNW 00075
WRITE (N16,6007) (BLANK, J=N,M)	PRNTNW 00076
LINE = LINE+3	PRNTNW 00077
GO TO 200	PRNTNW 00078
1500 CONTINUE	PRNTNW 00079
RETURN	PRNTNW 00080
9001 FORMAT (1H1,20X,8A10/ 50X,3A10/ 46X, 7H(EACH F5.3,5X,10HRED. FREQ.	PRNTNW 00081
1 * =*,F8.5, *)* /52X,#MODE SHAPE* ,I3)	PRNTNW 00082
9005 FORMAT (44X,42 (1H-),20X,A10,A4)	PRNTNW 00083
9006 FORMAT (4H0ROW, A1,14X,5H0ROW,I3, 3(A1,22X,5H0ROW,I3))	PRNTNW 00084
9007 FORMAT (3X, 4(A1,9X,4HREAL,8X,9HIMAGINARY))	PRNTNW 00085
9013 FORMAT (I4,8E16.8)	PRNTNW 00086
END	PRNTNW 00087

SUBROUTINE DCODER(IBOX, LBOX, IA, JA, IL, JL, SUBD, ICODE)	DCODNW 00002
DIMENSION IBOX(LBOX,1), ICODE(1)	DCODNW 00003
C	DCODNW 00004
C IBOX - ARRAY OF BOX CODES IN PACKED WORD FORMAT	DCODNW 00005
C LBOX - ROW DIMENSION OF BOX CODES ARRAY	DCODNW 00006
C IA - I-TH INDEX OF FIRST CODE TO RETRIEVE	DCODNW 00007
C JA - J-TH INDEX OF FIRST CODE TO RETRIEVE	DCODNW 00008
C IL - LAST BOX CODE ON THE JA-TH CHORD TO RETRIEVE	DCODNW 00009
C JL - LAST BOX ON THE IA-TH ROW TO RETRIEVE	DCODNW 00010
C SUBD - .T., SUBDIVIDED BOX CODES DESIRED, .F. UNSUBDIVIDED.	DCODNW 00011
C ICODE - ARRAY INTO WHICH BOX CODE WILL BE STORED.	DCODNW 00012
C	DCODNW 00013
C COMMENT ON USAGE	DCODNW 00014
C BOX CODES CAN BE RETRIEVED FOR ONE BOX, A ROW OR PART OF	DCODNW 00015
C A ROW, OR A COLUMN OR PART OF A COLUMN. A ROW AND COLUMN CAN	DCODNW 00016
C NOT BE RETRIEVED AT THE SAME TIME. IF ONLY 1 BOX IS DESIRED	DCODNW 00017
C SET IL = IA AND JL = JA. IF BOTH IL .NE. IA AND JL .NE.	DCODNW 00018
C JA, ONE ROW WILL BE RETURNED, IL BEING IGNORED.	DCODNW 00019
C	DCODNW 00020
COMMON /GEOMTY/ COPLAN, NSUBDV, XSUBDV, NSUBD2, NSUBCN, NSURF.	GEOMTY 00002
1 B1, B1BETA, B1S, B1BTAS, WLAX, WLAZ, PSIW,	GEOMTY 00003
2 MXBW, MXBBW, MYBW, MYBBW, MXBSW, MYBSW, MYBBSW,	GEOMTY 00004
3 IXBW, XCENR	GEOMTY 00005
LOGICAL COPLAN	GEOMTY 00006
LOGICAL SUBD	DCODNW 00022
INTEGER SHIFT	DCODNW 00023
DATA NEWRD /20/	DCODNW 00024
MASK = 7	DCODNW 00025
IB = 1	DCODNW 00026
IF (SUBD) GO TO 50	DCODNW 00027
I = NSUBDV * (JA-1) + IXBW	DCODNW 00028
J = NSUBDV * (JA-1) + NSUBCN	DCODNW 00029
ISKIP = NSUBDV	DCODNW 00030
IEND = NSUBDV * (IL-1) + IXBW	DCODNW 00031
JEND = NSUBDV * (JL-1) + NSUBCN	DCODNW 00032
GO TO 60	DCODNW 00033
50 CONTINUE	DCODNW 00034
I = IA	DCODNW 00035
J = JA	DCODNW 00036
ISKIP = 1	DCODNW 00037
IEND = IL	DCODNW 00038
JEND = JL	DCODNW 00039
60 CONTINUE	DCODNW 00040
IF (JL .EQ. JA) GO TO 1100	DCODNW 00041
C	DCODNW 00042
C PROGRAM WILL RETRIEVE NJ BOXES FROM ROW I	DCODNW 00043
100 CONTINUE	DCODNW 00044
DO 1000 JJ = J, JEND, ISKIP	DCODNW 00045
JSB = (JJ-1)/NBWD + 1	DCODNW 00046
IJWORD = IBOX(I, JSB)	DCODNW 00047
JR = (NBWD - MOD(JJ, NBWD)) * 3	DCODNW 00048
IF (JB, EQ, 60) JB = 0	DCODNW 00049
C JB = NUMBER OF BITS TO SHIFT LEFT.	DCODNW 00050
IJMASK = SHIFT(MASK, JB)	DCODNW 00051
IJCODE = IJWORD.AND. IJMASK	DCODNW 00052
NJB = -JB	DCODNW 00053
ICODE(IB) = SHIFT(IJCODE, NJB)	DCODNW 00054

```

      IB = IB + 1
1000 CONTINUE
      GO TO 3000
C
C      PROGRAM WILL RETRIEVE NJ BOXES FROM CHORD J
1100 CONTINUE
      JSB = (J-1)/NBWRD + 1
      JB = (NBWRD - MOD(J,NBWRD) ) * 3
      IF(JB.EQ.60) JB = 0
      IJMASK = SHIFT(MASK,JB)
      NJB = -JB
      D= 2000 II = I,IEND,ISKIP
      IJWORD = IBOX(II,JSB)
      IJCODE = IJWORD.AND.IJMASK
      ICODE(IB) = SHIFT(IJCODE,NJB)
      IB = IB + 1
2000 CONTINUE
C
3000 CONTINUE
      RETURN
      END

```

```

DCODNW 00055
DCODNW 00056
DCODNW 00057
DCODNW 00058
DCODNW 00059
DCODNW 00060
DCODNW 00061
DCODNW 00062
DCODNW 00063
DCODNW 00064
DCODNW 00065
DCODNW 00066
DCODNW 00067
DCODNW 00068
DCODNW 00069
DCODNW 00070
DCODNW 00071
DCODNW 00072
DCODNW 00073
DCODNW 00074
DCODNW 00075

```

SUBROUTINE POINTR(IX,MX, MYB,	SUBD, DIAPH, IBOX,LBXCD,	POINTR	00002
1	MXIR, IPOINT,IPNTIN,IPNTR)	POINTR	00003
C		POINTR	00004
C	GENERATES A POINTER ARRAY WHICH SERVES AS AN INDEX TO A	POINTR	00005
C	CONDENSED ARRAY OF BOX VALUES (MOSES, DOWNWASHES, ETC.)	POINTR	00006
C		POINTR	00007
C	IX = CENTER OF FIRST BOX TO USE	POINTR	00008
C	MX = NUMBER OF ROWS TO PROCESS	POINTR	00009
C	MYB = MAXIMUM ROW LENGTH	POINTR	00010
C	SUBD = .T., SUBDIVIDED BOXES TO BE USED	POINTR	00011
C	= .F., UNSUBDIVIDED BOXES TO BE USED	POINTR	00012
C	DIAPH = .T., DIAPHRAGMS TO BE INCLUDED	POINTR	00013
C	= .F., ONLY PLANKFORM BOXES	POINTR	00014
C	IBOX = ARRAY OF BOX CODES	POINTR	00015
C	LBXCD = ROW DIMENSION OF BOX CODES ARRAY	POINTR	00016
C	MXIR = SIZE OF IPNTR ARRAY	POINTR	00017
C	IPOINT = VALUE TO BE USED FOR FIRST POINTER (NORMALLY 1)	POINTR	00018
C	IN/OUT -	POINTR	00019
C	IPNTIN = LOCATION OF NEXT AVAILABLE LOCATION IN IPNTR ARRAY	POINTR	00020
C	(OVERLAP OF TAIL IS ACCOUNTED FOR HERE.)	POINTR	00021
C	OUTPUTS -	POINTR	00022
C	IPNTR = POINTER ARRAY - OUTPUT FROM THE SUBROUTINE	POINTR	00023
C	IPNTR(1,I) = LOCATION OF THE FIRST VALUE FOR ROW I	POINTR	00024
C	IPNTR(2,I) = CHORD LOCATION (SUBSCRIPT J) OF THAT VALUE	POINTR	00025
C		POINTR	00026
C	COMMON VALUES USED -	POINTR	00027
C		POINTR	00028
C		POINTR	00029
C	DIMENSION IBOX(LBXCD,1), IPNTR(2,1), ICODE(150)	POINTR	00030
C	LOGICAL DIAPH,SUBD,WING	POINTR	00031
C		POINTR	00032
C	ITOTBX = IPOINT	POINTR	00033
C	IL = IX + MX - 1	POINTR	00034
C	DO 100 IROW = IX,IL	POINTR	00035
C	IPBSUM=0	POINTR	00036
C	IUISUM=0	POINTR	00037
C	IPBX=1	POINTR	00038
C	IBXSUM = 0	POINTR	00039
C	CALL DDOCSR(IBOX,LBXCD, IROW,1, IROW,MYB, SUBD, ICODE)	POINTR	00040
C		POINTR	00041
C	FIND LAST BOX ON ROW	POINTR	00042
C	M = MYB	POINTR	00043
C	IF (DIAPH) GO TO 20	POINTR	00044
C	ONLY PLANKFORM BOXES DESIRED	POINTR	00045
C	DO 15 JCOL = 1,MYB	POINTR	00046
C	IF (ICODE(M) .EQ. 1) GO TO 30	POINTR	00047
C	M = M - 1	POINTR	00048
C	15 CONTINUE	POINTR	00049
C	GO TO 92	POINTR	00050
C	PLANKFORM AND DIAPHRAGM DESIRED	POINTR	00051
C	20 CONTINUE	POINTR	00052
C	DO 2 JCOL = 1,MYB	POINTR	00053
C	IF (ICODE(M) .NE. 0) GO TO 30	POINTR	00054
C	M = M - 1	POINTR	00055
C	25 CONTINUE	POINTR	00056
C	GO TO 92	POINTR	00057
C		POINTR	00058

```

C      LOOP ON CHORDS IN THE ROW
30 CONTINUE
   DO 90 JCCL = 1,M
   IF (ICODE(JCCL) - 1) 35,40,50
C
C      ICODE = 0
C
C      35 IF (IDISUM.NE.0) GO TO 40
   IPBX = IPBX + 1
   GO TO 90
C
C      ICODE = 1
C      40 IPBSUM = IDISUM + 1
C
C      ICODE = 2 OR 3
C      50 CONTINUE
   IDISUM = IDISUM + 1
   90 CONTINUE
C
C      92 CONTINUE
   IPNTR(1,IPNTIN) = ITOTBX
   IPNTR(2,IPNTIN) = IPBX
   IPNTIN = MOD(IPNTIN,MXIR) + 1
   IF (DIAPH) GO TO 95
   ITOTBX = ITOTBX + IPBSUM
   GO TO 100
C      95 ITOTBX = ITOTBX + IDISUM
100 CONTINUE
   IPNTR(1,IPNTIN) = ITOTBX
   IPNTR(2,IPNTIN) = 0
   RETURN
   END

```

```

POINTR 00059
POINTR 00060
POINTR 00061
POINTR 00062
POINTR 00063
POINTR 00064
POINTR 00065
POINTR 00066
POINTR 00067
POINTR 00068
POINTR 00069
POINTR 00070
POINTR 00071
POINTR 00072
POINTR 00073
POINTR 00074
POINTR 00075
POINTR 00076
POINTR 00077
POINTR 00078
POINTR 00079
POINTR 00080
POINTR 00081
POINTR 00082
POINTR 00083
POINTR 00084
POINTR 00085
POINTR 00086
POINTR 00087
POINTR 00088
POINTR 00089
POINTR 00090

```

FUNCTION LOCSDW(IROW,JCOL,IPNTSD,IPNTIN,IPNTOT,IPNTLS)	LOCSDW 00002
C	LOCSDW 00003
C RETURNS THE LOCATION OF THE WORD IN THE END-AROUND SUBDIVIDED	LOCSDW 00004
C DOWNWASH ARRAY CORRESPONDING TO BOX(IROW,JCOL) OF THE SUB-	LOCSDW 00005
C DIVIDED BOX ARRAY	LOCSDW 00006
C	LOCSDW 00007
C IROW = BOX CHORDWISE LOCATION	LOCSDW 00008
C JCOL = BOX SPANWISE LOCATION	LOCSDW 00009
C IPNTSD = ARRAY OF POINTERS	LOCSDW 00010
C IPNTIN = NEXT AVAILABLE (UNUSED) CELL IN IPNTSD (END-	LOCSDW 00011
C AROUND)	LOCSDW 00012
C IPNTOT = FIRST CURRENTLY AVAILABLE CELL IN IPNTSD	LOCSDW 00013
C IPNTLS = LAST CELL OF IPNTSD (LENGTH OF ARRAY)	LOCSDW 00014
C RETURN -	LOCSDW 00015
C LOCSDW = LOCATION OF DESIRED DOWNWASH, IF SUCCESSFUL	LOCSDW 00016
C = 0, IF LOCNT LIES OUTSIDE THE DEFINED AREA.	LOCSDW 00017
C	LOCSDW 00018
C DIMENSION IPNTSD(2,IPNTLS)	LOCSDW 00019
C	LOCSDW 00020
C LOCNT = MOD(IROW-1,IPNTLS) + 1	LOCSDW 00021
C LOCNT = LOCATION OF CELL IN IPNTSD WHICH WAS OR IS TO BE	LOCSDW 00022
C USED	LOCSDW 00023
C IF(IPNTIN - IPNTOT) 100, 300, 200	LOCSDW 00024
C END AROUND HAS OCCURRED	LOCSDW 00025
C 100 IF (LOCNT - IPNTIN) 400, 300, 150	LOCSDW 00026
C NOT IN UPPER PART. IS LOCNT WITHIN BOTTOM PART -	LOCSDW 00027
C 150 IF (LOCNT - IPNTOT) 300, 400, 400	LOCSDW 00028
C	LOCSDW 00029
C NO END AROUND, NORMAL SEQUENCE	LOCSDW 00030
C 200 IF (LOCNT - IPNTIN) 250, 300, 300	LOCSDW 00031
C LESS THAN UPPER LIMIT. IS LOCNT .GE. LOWER LIMIT -	LOCSDW 00032
C 250 IF (LOCNT .GE. IPNTOT) GO TO 400	LOCSDW 00033
C	LOCSDW 00034
C ERROR OR INITIAL CONDITION ENCOUNTERED (SHOULD NEVER OCCUR)	LOCSDW 00035
C 300 LOCSDW = 0	LOCSDW 00036
C GO TO 500	LOCSDW 00037
C	LOCSDW 00038
C SUCCESSFUL, BOX HAS BEEN DEFINED	LOCSDW 00039
C 400 IFB = IPNTSD(2,LOCNT)	LOCSDW 00040
C IF(JCOL.LT.IFB) GO TO 300	LOCSDW 00041
C LOCSDW = IPNTSD(1,LOCNT) + JCOL-IFB	LOCSDW 00042
C	LOCSDW 00043
C 500 CONTINUE	LOCSDW 00044
C RETURN	LOCSDW 00045
C END	LOCSDW 00046

	SUBROUTINE SMPLW(IBOX,LBXCD, JCHRD, JT, IFRST,ILAST)	SMPLW	00002
C		SMPLW	00003
C	COMPUTES DOWNWASH, SIDEWASH AND VELOCITY POTENTIAL FOR A	SMPLW	00004
C	SAMPLE CHORD LOCATED IN THE WING FLOW FIELD	SMPLW	00005
C		SMPLW	00006
C	IBOX = ARRAY OF BOX CODES FOR THE WING	SMPLW	00007
C	LBXCD = LENGTH OF BOX CODE ARRAY	SMPLW	00008
C	JCHRD = SAMPLE-WASH CHORD NUMBER	SMPLW	00009
C	JT = J-LOCATION OF THE CHORD	SMPLW	00010
C	IFRST = NUMBER OF FIRST SAMPLE BOX	SMPLW	00011
C	ILAST = NUMBER OF LAST SAMPLE BOX	SMPLW	00012
C		SMPLW	00013
	DIMENSION IBOX(LBXCD,1)	SMPLW	00014
	COMMON /CONTRL/ PREVEX,OMACH, TITLE(8), PRVGEOM,PRVMODE,DIHW,DIHT,	CONTRL	00002
1	DEFAULT	CONTRL	00003
	LOGICAL PRVGEOM,PRVMODE,DIHW,DIHT,DEFAULT	CONTRL	00004
	COMMON /PROBLW/ XMACH,NMODES,NTSLOP,PKVALS,SMOOTH,NDEG,CRDFIT,	PROBLW	00002
1	EXAIC,SUBDV,PLYWOOD	PROBLW	00003
	LOGICAL SMOOTH,CRDFIT,EXAIC,SUBDV,PLYWOOD	PROBLW	00004
	COMMON /SNWASH/ IPNTSD(2,50), ENSUBD(2,600), IPNTIN,IPNTOT,IPNTLS	SNWASH	00002
C	IPNTSD(LPNTSD), ENSUBD(2*LSDW)	SNWASH	00003
	COMPLEX ENSUBD	SNWASH	00004
	COMMON /MUAICS/ YBAR,EL,MUAIC(2,50),NROWS,SURF,	MUAICS	00002
1	YBARL,ELL, MUATCL(2,50),NROWSL,SURFL,PSIDIF	MUAICS	00003
	LOGICAL SURF,SURFL	MUAICS	00004
	COMMON /GEOMTY/ COPLAN,NSUBDV,XSUBDV,NSUBD2,NSUBCN,NSURF,	GEOMTY	00002
1	B1,B1BETA,B1S,B1BTAS,WLAX,WLAZ,PSIW,	GEOMTY	00003
2	MXBW,MXBBW,MYBW,MYBBW,MXBSW,MYBSW,MYBBSW,	GEOMTY	00004
3	IXBW,XCENTR	GEOMTY	00005
	LOGICAL COPLAN	GEOMTY	00006
	COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOJTP,	FILES	00002
1	IQUFSP,MODESC,IVPSC,IGECSC,IWTFSC,IAICSC	FILES	00003
	COMMON /IOCONT/ OPLAIC,OSPAIC,WTGEOM,WTGNAF,WTSL,WTBL,PRBOX,	IOCONT	00002
1	PRPAIC,PRSAIC,PRMODS,PRCOEF,PRDW,PRSW,PRVP,	IOCONT	00003
2	PRBL,PRDCP,PRGNAF,PRGNAC,PRSL,PRLW,PRNW,PRCM	BCSFRB	00001
	EQUIVALENCE (PRUW,PRDW)	IOCONT	00005
	LOGICAL OPLAIC,OSPAIC,WTGEOM,WTGNAF,WTSL,WTBL,PRBOX,PRPAIC,	IOCONT	00006
1	PRSAIC,PRMODS,PRCOEF,PRDW,PRSW,PRVP,PRBL,PRSL,PRGNAF,	IOCONT	00007
2	PRDCP,PRGNAC,PRUW,PRLW,PRNW,PRCM	BCSFRB	00002
	COMMON / MODES/ SYM,SYMT,MTYPEW,MTYPET	MODECOM	00002
	COMMON /NWASHES/ IPNTLW(2,100),ENRUS(1275), ENRLS(1275),IOMLAPN	NWASHES	00002
	COMPLEX ENRUS, ENRLS	NWASHES	00003
	COMMON /AICS / XXVL, C(1640),W(1640),V(1640)	AICS	00002
	COMPLEX C, W, V	AICS	00003
	COMPLEX DW(52),SW(50),LW(50),PHI(50)	SMPLW	00025
	EQUIVALENCE (SW,IPNTSD),(LW,ENSUBD),(DW,ENSUBD(1,26))	SMPLW	00026
	EQUIVALENCE (PHI,ENSUBD(1,52))	SMPLW	00027
	CC .EX WBSUR,VSUM,PHISUM, EN	SMPLW	00028
	INTEGER RWT,LWT	SMPLW	00029
	DATA RWT,LWT / 3,4 /	SMPLW	00030
C		SMPLW	00031
C	SET CONSTANTS	SMPLW	00032
	CPBI = COS(PSIW)	SMPLW	00033
	SPBI = SIN(PSIW)	SMPLW	00034
	BINV = 1./B1	SMPLW	00035
C		SMPLW	00036
C	COMPUTE THE RIGHT WING CONTRIBUTION TO THE SAMPLE CHORD	SMPLW	00037

C		SMPLW	00038
C	GET THE NECESSARY AIC ARRAYS	SMPLW	00039
	CALL GETAIC(JOHRD,RWT, 0, IR)	SMPLW	00040
	IF (IR .NE. 0) GO TO 800	SMPLW	00041
	YUBAR = (JT-.5 + EL*SPSI) / CPSI	SMPLW	00042
	JBAR = YUBAR	SMPLW	00043
	IF (YUBAR .GE. 0) JBAR = JBAR + 1	SMPLW	00044
	NUBMIN = ABS(EL) + .5	SMPLW	00045
	IBX = 1	SMPLW	00046
C		SMPLW	00047
C	LOOP ON BOXES ALONG THE SAMPLE CHORD	SMPLW	00048
	DO 200 IBXX = IFRST,ILAST	SMPLW	00049
	NUBMAX = IBXX - 1	SMPLW	00050
	I = IBXX - NUBMIN	SMPLW	00051
C		SMPLW	00052
C	ZERO OUT THE SUMMATION VARIABLES	SMPLW	00053
	WSUM = (0.,0.)	SMPLW	00054
	VSUM = (0.,0.)	SMPLW	00055
	PHISUM = (0.,0.)	SMPLW	00056
C		SMPLW	00057
	IF (YBAR) 120,125,130	SMPLW	00058
120	JINCR = -1	SMPLW	00059
	GO TO 135	SMPLW	00060
125	IAIC = NUBMIN+2	SMPLW	00061
	INCAIC = 2*NUBMIN + 1	SMPLW	00062
	JINCR = 1	SMPLW	00063
	GO TO 140	SMPLW	00064
130	JINCR = 1	SMPLW	00065
135	IAIC = NUBMIN+2 + NUBMIN	SMPLW	00066
	INCAIC = 2*NUBMIN + 2	SMPLW	00067
140	CONTINUE	SMPLW	00068
C		SMPLW	00069
C	LOOP FORWARD OVER THE RIGHT WING	SMPLW	00070
	DO 190 NUBAR = NUBMIN,NUBMAX	SMPLW	00071
	MUAIC1 = MUAIC(1,NUBAR+1)	SMPLW	00072
	MUAIC2 = MUAIC(2,NUBAR+1)	SMPLW	00073
	IF (MUAIC2 .EQ. 0) GO TO 185	SMPLW	00074
	IF (YBAR .GE. 0) GO TO 150	SMPLW	00075
	JCOLR = JBAR + NUBAR - MUAIC2 + 1	SMPLW	00076
	GO TO 160	SMPLW	00077
150	JCOLR = JBAR - NUBAR + MUAIC1 - 1	SMPLW	00078
160	CONTINUE	SMPLW	00079
C		SMPLW	00080
C	LOOP ON A ROW OF WING BOXES, COMPUTING RIGHT WING CONTRIBUTION	SMPLW	00081
	DO 180 MUAI = MUAIC1,MUAIC2	SMPLW	00082
	IF (JCOLR .LE. 0) GO TO 170	SMPLW	00083
	CALL DCODR(IBOX,LBOXD, I,JCOLR, I,JCOLR, .F., ICD)	SMPLW	00084
	IF (ICD .EQ. 0) GO TO 170	SMPLW	00085
C	A CONTRIBUTING BOX HAS BEEN FOUND. GET THE AIC LOCATION	SMPLW	00086
	KAIC = IATC + MUAI	SMPLW	00087
C	GET THE NORMAL-WASH LOCATION	SMPLW	00088
	IDB = LOCSDW(I,JCOLR, IPNTDW,LPNTDW, 1, LPNTDW)	SMPLW	00089
C	ADD THIS CONTRIBUTION TO THE SUMS	SMPLW	00090
	IF (EL .LT. 0) GO TO 163	SMPLW	00091
	EN = ENRUS(IDB)	SMPLW	00092
	GO TO 165	SMPLW	00093
163	EN = ENRLS(IDB)	SMPLW	00094

165	CONTINUE	SMPLW	00095
	WSUM = WSUM + W(KAIC) * EN	SMPLW	00096
	VSUM = VSUM + V(KAIC) * EN	SMPLW	00097
	PHISUM = PHISUM + C(KAIC) * EN	SMPLW	00098
170	CONTINUE	SMPLW	00099
	JCOLR = JCOLR + JINCR	SMPLW	00100
180	CONTINUE	SMPLW	00101
C	END OF LOOP FOR RIGHT WING ROW CONTRIBUTIONS	SMPLW	00102
C		SMPLW	00103
185	CONTINUE	SMPLW	00104
	I = I - 1	SMPLW	00105
	IF (I .LE. 0) GO TO 195	SMPLW	00106
	IAIC = IAIC + INCAIC	SMPLW	00107
	INCAIC = INCAIC + 2	SMPLW	00108
190	CONTINUE	SMPLW	00109
C	END OF LOOP FORWARD ON RIGHT WING ROWS, FROM 140*	SMPLW	00110
C		SMPLW	00111
195	CONTINUE	SMPLW	00112
	DW(IBX) = BINW * (CPSI*WSUM + SPSI*VSUM)	SMPLW	00113
	SW(IBX) = BINW * (CPSI*WSUM - SPSI*VSUM)	SMPLW	00114
	PHI(IBX) = PHISUM	SMPLW	00115
	IBX = IBX + 1	SMPLW	00116
200	CONTINUE	SMPLW	00117
C	END OF LOOP ON RECEIVING BOXES, FOR RIGHT WING CONTRIBUTIONS	SMPLW	00118
C		SMPLW	00119
	NBXS = IBX - 1	SMPLW	00120
C	IS LEFT WING CONTRIBUTION NEEDED -	SMPLW	00121
	IF (SYM .EQ. 0) GO TO 310	SMPLW	00122
C	YES. GET THE AIC ARRAYS FOR LEFT WING CONTRIBUTIONS.	SMPLW	00123
	CALL GETAIC(JOHRD, LWT, 0, IR)	SMPLW	00124
	IF (IR .NE. 0) GO TO 800	SMPLW	00125
	NUBMIN = ABS(EL) + .5	SMPLW	00126
	IBX = 1	SMPLW	00127
	YMUBAR = (-JT+.5 + EL*SPSI) / CPSI	SMPLW	00128
	JBAR = YMUBAR	SMPLW	00129
	IF (YMUBAR .GE. 0) JBAR = JBAR + 1	SMPLW	00130
C	LOOP ON BOXES ALONG THE SAMPLE CHORD	SMPLW	00131
	DO 300 IBXX = IFRST, ILAST	SMPLW	00132
	NUBMAX = IBXX - 1	SMPLW	00133
	I = IBXX - NUBMIN	SMPLW	00134
C	ZERO OUT THE SUMMATION VARIABLES	SMPLW	00135
	WSUM = (0.,0.)	SMPLW	00136
	VSUM = (0.,0.)	SMPLW	00137
	PHISUM = (0.,0.)	SMPLW	00138
C		SMPLW	00139
	IF (YBAR) 220,225,230	SMPLW	00140
220	JINCR = 1	SMPLW	00141
	GO TO 235	SMPLW	00142
225	IAIC = NUBMIN**2	SMPLW	00143
	INCAIC = 2*NUBMIN + 1	SMPLW	00144
	JINCR = -1	SMPLW	00145
	GO TO 240	SMPLW	00146
230	JINCR = -1	SMPLW	00147
235	IAIC = NUBMIN**2 + NUBMIN	SMPLW	00148
	INCAIC = 2*NUBMIN + 2	SMPLW	00149
240	CONTINUE	SMPLW	00150
C		SMPLW	00151

C	LOOP FORWARD OVER THE LEFT WING	SMPLW	00152
	DO 290 NUBAR = NUBMIN,NUBMAX	SMPLW	00153
	MUAIC1 = MUAIC(1,NUBAR+1)	SMPLW	00154
	MUAIC2 = MUAIC(2,NUBAR+1)	SMPLW	00155
	IF (MUAIC2 .LE. 3) GO TO 285	SMPLW	00156
	IF (YBAR .GE. 0) GO TO 250	SMPLW	00157
	JCOLL = JBAR -NUBAR +MUAIC1 -1	SMPLW	00158
	GO TO 260	SMPLW	00159
	250 JCOLL = JBAR +NUBAR -MUAIC1 +1	SMPLW	00160
	260 CONTINUE	SMPLW	00161
C		SMPLW	00162
C	LOOP ON A ROW OF WING BOXES, COMPUTING LEFT WING CONTRIBUTIONS	SMPLW	00163
	DO 280 MUAI = MUAIC1,MUAIC2	SMPLW	00164
	IF (JCOLL .LE. 0) GO TO 270	SMPLW	00165
	CALL DCCDER(ibox,lbxcd, I,JCOLL, I,JCOLL, .F., ICD)	SMPLW	00166
	IF (ICD .EQ. 0) GO TO 270	SMPLW	00167
C	A CONTRIBUTING BOX HAS BEEN FOUND. GET THE AIC LOCATION	SMPLW	00168
	KAIC = IAIC + MUAI	SMPLW	00169
C	GET THE NORMAL-WASH LOCATION	SMPLW	00170
	IDS = LOCSDW(I,JCOLL, IPNTDW, LPNTDW, 1, LPNTDW)	SMPLW	00171
C	ADD THIS CONTRIBUTION TO THE SUMS	SMPLW	00172
	IF (EL .LT. 0) GO TO 263	SMPLW	00173
	EN = ENRUS(IDS)	SMPLW	00174
	GO TO 265	SMPLW	00175
	263 EN = ENRUS(IDS)	SMPLW	00176
	265 CONTINUE	SMPLW	00177
	WSUM = WSUM + W(KAIC) * EN	SMPLW	00178
	VSUM = VSUM + V(KAIC) * EN	SMPLW	00179
	PHISUM = PHISUM + C(KAIC) * EN	SMPLW	00180
	270 CONTINUE	SMPLW	00181
	JCOLL = JCOLL + JINCR	SMPLW	00182
	280 CONTINUE	SMPLW	00183
C	END OF LOOP FOR LEFT WING ROW CONTRIBUTIONS	SMPLW	00184
C		SMPLW	00185
	285 CONTINUE	SMPLW	00186
	I = I - 1	SMPLW	00187
	IF (I .LE. 0) GO TO 295	SMPLW	00188
	IAIC = IAIC + INCAIC	SMPLW	00189
	INCAIC = INCAIC + 2	SMPLW	00190
	290 CONTINUE	SMPLW	00191
C	END OF LOOP FORWARD ON LEFT WING ROWS, FROM 240	SMPLW	00192
C		SMPLW	00193
C		SMPLW	00194
	295 CONTINUE	SMPLW	00195
	DW(IBX) = DW(IBX) + BINV*(CPSI*WSUM - SPSI*VSUM) * SYM	SMPLW	00196
	SW(IBX) = SW(IBX) + BINV*(CPSI*VSUM + SPSI*WSUM) * SYM	SMPLW	00197
	PHI(IBX) = PHI(IBX) + PHISUM*SYM	SMPLW	00198
	IBX = IBX + 1	SMPLW	00199
	300 CONTINUE	SMPLW	00200
C	END OF LOOP ON RECEIVING BOXES, FOR LEFT WING CONTRIBUTIONS	SMPLW	00201
C		SMPLW	00202
C	DETERMINE WHAT TO PRINT	SMPLW	00203
	310 CONTINUE	SMPLW	00204
	WRITE (NT6,6001) TITLE, XMACH,XKVL, JT, IFRST,ILAST	SMPLW	00205
	IF (.NOT. PRDW) GO TO 330	SMPLW	00206
	WRITE (NT6,6010	SMPLW	00207
	WRITE (NT6,6010, (DW(I),I = 1,NBXS)	SMPLW	00208

330 CONTINUE	SMPLW 00209
IF (.NOT. PRSW) GO TO 340	SMPLW 00210
WRITE (NT6,6011)	SMPLW 00211
WRITE (NT6,6013) (SW(I),I = 1,NBXS)	SMPLW 00212
340 CONTINUE	SMPLW 00213
IF (.NOT. PRLW) GO TO 400	SMPLW 00214
IF (NBXS .LT. 2) GO TO 400	SMPLW 00215
ISUB = 0	SMPLW 00216
PHI(ISUB) = 2.*PHI(1) - PHI(2)	SMPLW 00217
PHI(NBXS+1) = 2.*PHI(NBXS) - PHI(NBXS-1)	SMPLW 00218
FACTOR = B1BETA/2.0 * BINV**2	SMPLW 00219
DO 350 I = 1,NBXS	SMPLW 00220
LW(I) = FACTOR*(PHI(I+1) - PHI(I-1))	SMPLW 00221
350 CONTINUE	SMPLW 00222
WRITE (NT6,6012)	SMPLW 00223
WRITE (NT6,6013) (LW(I),I=1,NBXS)	SMPLW 00224
C	SMPLW 00225
400 RETURN	SMPLW 00226
C	SMPLW 00227
800 WRITE (NT6,8000) IR	SMPLW 00228
GO TO 400	SMPLW 00229
C	SMPLW 00230
6001 FORMAT(1H1,20X,8A10/ 51X,16HFLOWFIELD SAMPLING /40X,7H(MACH ,	SMPLW 00231
1 F5.3,5X,12HRED. FREQ. =,F8.5, 2H)/ 41X,16HSAMPLED AT CHORD	SMPLW 00232
2 13, 6H, BOX I2, 6H TO BOX I2 / 1H0,2X, 4(10X,4HREAL,8X,	SMPLW 00233
3 9HIMAGINARY))	SMPLW 00234
6010 FORMAT(15H0 UP-WASHES -)	SMPLW 00235
6011 FORMAT(15H0SIDE-WASHES -)	SMPLW 00236
6012 FORMAT(22H0LONGITUDINAL-WASHES -)	SMPLW 00237
6013 FORMAT(4X,8E16.8)	SMPLW 00238
C	SMPLW 00239
8000 FORMAT(54H*** WARNING - PROBLEMS ENCOUNTERED WHILE GETTING AICS	SMPLW 00240
1 39H FOR FLOW-FIELD SAMPLING. ERROR CODE = ,I5, 4H ***)	SMPLW 00241
END	SMPLW 00242

	PROGRAM SMTH	FTNX1 00066
C		SMOOTH 00004
C	THIS PROGRAM WILL FIT A LEAST SQUARES POLYNOMIAL SURFACE	SMOOTH 00005
C	THROUGH THE VELOCITY POTENTIALS FOR A PLANFORM. A NEW SET OF	SMOOTH 00006
C	VELOCITY POTENTIALS WILL BE CALCULATED FROM THE POLYNOMIALS	SMOOTH 00007
C	AND WRITTEN ON THE IVPSC FILE.	SMOOTH 00008
C		SMOOTH 00009
	COMMON /ARRAYS/ KBXCW, LBXCW, LBOXC, KBXCDT, LBXCDT, KJALPH, LJALPH,	ARRAYS 00002
1	KALPHA, KKERNL, LKERNL, KPNTRM, LPNTRM, KDEFSL, KELPHI,	ARRAYS 00003
2	LMODES, KPNTSD, LPNTSD, KSDW, LSDW, KPNTDW, LPNTDW,	ARRAYS 00004
3	KDW, LDW, KTVP, LTVP	ARRAYS 00005
	COMMON /FILES / NT5, NT6, INTAPE, INFSP, NPLAIC, NSPAIC, NOUTP,	FILES 00002
1	IQUFSP, MODESC, IVPSC, IGEOSC, IWFSC, IAISC	FILES 00003
	COMMON /IOCONT/ OPLAIC, OSPAIC, WTGEOM, WTGNF, WTSI, WTBL, FRBOX,	IOCONT 00002
1	FRPAIC, FRSAIC, FRMODS, FRCOEF, FRDW, FRSW, FRVP,	IOCONT 00003
2	FRBL, FRDCP, FRGNF, FRGNAC, FRSL, FRLW, FRNW, FRM	BCSFRB 00001
	EQUIVALENCE (FRUW, FRDW)	IOCONT 00005
	LOGICAL OPLAIC, OSPAIC, WTGEOM, WTGNF, WTSI, WTBL, FRBOX, FRPAIC,	IOCONT 00006
1	FRSAIC, FRMODS, FRCOEF, FRDW, FRSW, FRVP, FRBL, FRSL, FRGNF,	IOCONT 00007
2	FRDCP, FRGNAC, FRUW, FRLW, FRNW, FRM	BCSFRB 00002
	COMMON /PROBLM/ XMACH, NMODES, NTSLOP, NKVALS, SMOOTH, NDEG, CRDFIT,	PROBLM 00002
1	EXAIC, SUBDV, PLYWOOD	PROBLM 00003
	LOGICAL SMOOTH, CRDFIT, EXAIC, SUBDV, PLYWOOD	PROBLM 00004
	COMMON /KVAL / IKVAL, XKVAL(20), XKS(20)	KVAL 00002
	COMMON /GEOMTY/ COPLAN, NSUBDV, XSUBDV, NSUBD2, NSUBCN, NSURF,	GEOMTY 00002
1	B1, B1BETA, B1S, B1BTAS, WLAX, WLAZ, PSIW,	GEOMTY 00003
2	MXCW, MXBBW, MYBW, MYBBW, MXBSW, MYBSW, MYBBSW,	GEOMTY 00004
3	IXBW, XCENR	GEOMTY 00005
	LOGICAL COPLAN	GEOMTY 00006
	COMMON /GEOM2 / TLAX, TLAZ, FSIT, MXBT, MYBT, MYBBT, MXBST, MYBST,	GEOM2 00002
1	MYBBST, IXBT, IXBST, CARL	GEOM2 00003
	COMMON /TAPEIO/ NFS, NMS, LS, NMR, ID(20), NID, ITYPE, LRS, LWS, M, N,	TAPEIO 00002
1	PARM(10), IRR	TAPEIO 00003
	DIMENSION IPARM(10)	TAPEIO 00004
	EQUIVALENCE (PARM, IPARM)	TAPEIO 00005
	COMMON /CHECKFR/ DFFCFR, GEOCFR, MODCFR, AICCFR, NMSCFR, SMCFR, GAFCFR	CHECKFR 00002
	LOGICAL DFFCFR, GEOCFR, MODCFR, AICCFR, NMSCFR, SMCFR, GAFCFR	CHECKFR 00003
	EQUIVALENCE (CHECKFR, SMCFR)	SMOOTH 00020
	LOGICAL CHECKFR	SMOOTH 00021
C		SMOOTH 00022
C	DELPHI(NBOXES), TVP(NCOLS1 + NCOLS2 * NSUBDV)	SMOOTH 00023
	COMPLEX DELPHI(1000), TVP(250), AVPS(1250)	SMOOTH 00024
C	X(NO. DELPHI + NO. TVP), Y(SAME)	SMOOTH 00025
	COMPLEX SDELPH	SMOOTH 00026
	DIMENSION X(1250), Y(1250)	SMOOTH 00027
C	A(NO. COEFF.)	SMOOTH 00028
	COMPLEX A(66)	SMOOTH 00029
C		SMOOTH 00030
	COMMON /INDEX/ IS(100), NOC(100), JS(100), JOC(100)	SMOOTH 00031
C	XP(NO. COEFF. + 1), YP(SAME)	SMOOTH 00032
	DIMENSION XP(11), YP(11)	SMOOTH 00033
C	FELOC((MYBW*MYBT)*NSUBDV), TEXLOC(SAME)	SMOOTH 00034
	DIMENSION FELOC(250), TEXLOC(250)	SMOOTH 00035
C	IPNTRM(2, NROWS*NSUBDV)	SMOOTH 00036
	DIMENSION IPNTRM(2, 150)	SMOOTH 00037
	DIMENSION TITL(3)	SMOOTH 00038
C		SMOOTH 00039

REAL K1	SMOOTH 00040
COMPLEX VP	SMOOTH 00041
LOGICAL MXREAD,MXWRITE,RANDIN,RANDOUT	SMOOTH 00042
K1 = MXVAL(IKVAL)	SMOOTH 00043
MXREAD = .FALSE.	SMOOTH 00044
RANDIN = .FALSE.	SMOOTH 00045
MXWRITE = .FALSE.	SMOOTH 00046
RANDOUT = .FALSE.	SMOOTH 00047
MXB = MXBW	SMOOTH 00048
IF(COPLAN) MXB = MXBT	SMOOTH 00049
C	SMOOTH 00050
C	SMOOTH 00051
C PUT NAME OF SCRATCH FILE FOR SMOOTHED VALUES INTO PLACE	SMOOTH 00052
C	SMOOTH 00053
REWIND IGEOSC	SMOOTH 00054
REWIND NEVPSC	SMOOTH 00055
C	SMOOTH 00056
C GET THE PLANFORM POINTERS FROM THE MODESC FILE	SMOOTH 00057
C	SMOOTH 00058
REWIND MODESC	SMOOTH 00059
CALL RDINIT	SMOOTH 00060
ITYPE = SHMIXED	SMOOTH 00061
MXARRY = 6H1PNTRM	SMOOTH 00062
CALL READMX(MODESC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,II),ITYPE,	SMOOTH 00063
1 LRS,1PNTRM,2,NPTRS,FARM,IRR)	SMOOTH 00064
IOVLAP = IFARM(3)	SMOOTH 00065
IF(IRR.NE.0) GO TO 6020	SMOOTH 00066
C	SMOOTH 00067
CALL RDINIT	SMOOTH 00068
ITYPE = SHMIXED	SMOOTH 00069
NFS = 1	SMOOTH 00070
MXARRY = 6H1S PT.	SMOOTH 00071
CALL READMX(MODESC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,	SMOOTH 00072
1 LRS,IS,M,N,FARM,IRR)	SMOOTH 00073
IF(IRR.NE.0) GO TO 6020	SMOOTH 00074
C	SMOOTH 00075
C	SMOOTH 00076
C READ THE FEXLOC AND TEXLOC ARRAYS FROM THE GEOMETRY SCRATCH	SMOOTH 00077
C FILE. THESE ARE NEEDED TO INTERPOLATE VELOCITY POTENTIALS AT	SMOOTH 00078
C BOX EDGES.	SMOOTH 00079
C	SMOOTH 00080
REWIND IGEOSC	SMOOTH 00081
CALL RDINIT	SMOOTH 00082
NMS = 2	SMOOTH 00083
IF(NSURF.EQ.1.OR.COPLAN) NMS=1	SMOOTH 00084
ITYPE = SHMIXED	SMOOTH 00085
MXARRY = 6H1FEXLOC	SMOOTH 00086
CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,	SMOOTH 00087
1 LRS,FEXLOC,M,N,PARM,IRR)	SMOOTH 00088
IF(IRR.NE.0) GO TO 6010	SMOOTH 00089
C	SMOOTH 00090
CALL RDINIT	SMOOTH 00091
ITYPE = SHMIXED	SMOOTH 00092
MXARRY = 6H1TEXLOC	SMOOTH 00093
CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,	SMOOTH 00094
1 LRS,TEXLOC,M,N,PARM,IRR)	SMOOTH 00095
IF(IRR.NE.0) GO TO 6010	SMOOTH 00096

C		SMOOTH	00097
C	REORDER THE FEXLOC AND TEXLOC ARRAYS SO THAT THERE ARE	SMOOTH	00098
C	VALUES FOR UNSUBDIVIDED CHORDS ONLY.	SMOOTH	00099
C		SMOOTH	00100
	IF(NSUBDV.EQ.1) GO TO 120	SMOOTH	00101
	XSLIDE = NSUBDV - IXBW	SMOOTH	00102
	JCOL = NSUBCN	SMOOTH	00103
	NCOLS = MYBW + MYBT	SMOOTH	00104
	DO 110 I=1,NCOLS	SMOOTH	00105
	TEXLOC(I) = (TEXLOC(JCOL) + XSLIDE)/XSUBDV	SMOOTH	00106
	FEXLOC(I) = (FEXLOC(JCOL) + XSLIDE)/XSUBDV	SMOOTH	00107
	JCOL = JCOL + NSUBDV	SMOOTH	00108
	110 CONTINUE	SMOOTH	00109
	120 CONTINUE	SMOOTH	00110
C		SMOOTH	00111
C		SMOOTH	00112
C	LOOP ON NUMBER OF MODES (ALSO NO. OF V.P.)	SMOOTH	00113
	REWIND IVPSC	SMOOTH	00114
	DO 2000 NM=1,NMODES	SMOOTH	00115
C		SMOOTH	00116
C	READ DELPHI ARRAY FROM IVPSC. THE TVP ARRAY MUST BE SKIPPED	SMOOTH	00117
C	IF NM IS NOT 1	SMOOTH	00118
C		SMOOTH	00119
	CALL RDINIT	SMOOTH	00120
C		SMOOTH	00121
C		SMOOTH	00122
	ITYPE = SHMIXED	SMOOTH	00123
	CALL READMX(IVPSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	SMOOTH	00124
	1 LRS,DELPHI,M,N,FARM,IRR)	SMOOTH	00125
	IF(IRR.NE.0) GO TO 6040	SMOOTH	00126
C		SMOOTH	00127
	CALL RDINIT	SMOOTH	00128
	ITYPE = SHMIXED	SMOOTH	00129
	CALL READMX(IVPSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	SMOOTH	00130
	1 LRS,TVP,M,N,FARM,IRR)	SMOOTH	00131
	IF(IRR.NE.0) GO TO 6040	SMOOTH	00132
C		SMOOTH	00133
C	LOOP ON NSURF TO FIT EACH PLANFORM INDEPENDENTLY.	SMOOTH	00134
C		SMOOTH	00135
	DO 1000 NS=1,NSURF	SMOOTH	00136
C		SMOOTH	00137
C	MOVE DELPHI FOR PLANFORM NS TO AVF ARRAY DELETING ZERO	SMOOTH	00138
C	VALUES AND OBTAINING THE (X,Y) COORDINATES IN (I,J) INDICES	SMOOTH	00139
C		SMOOTH	00140
	IF(NS.EQ.2) GO TO 100	SMOOTH	00141
C		SMOOTH	00142
C	FIRST PLANFORM	SMOOTH	00143
	IBEG = 1	SMOOTH	00144
	ILIM = MXBW	SMOOTH	00145
	IC = 0	SMOOTH	00146
	NCH = 0	SMOOTH	00147
	GO TO 200	SMOOTH	00148
C		SMOOTH	00149
C	SECOND PLANFORM	SMOOTH	00150
	100 CONTINUE	SMOOTH	00151
	IFBT = (IXBT-IXBW)/NSUBDV + 1	SMOOTH	00152
	IBEG = IFBT	SMOOTH	00153

ILIM = MYBT	SMOOTH 00154
IC = 0	SMOOTH 00155
NCH = MYBW	SMOOTH 00156
IUP = MYBW	SMOOTH 00157
IF(COPLAN) IUP = IFBT - 1	SMOOTH 00158
DO 125 I=1,IUP	SMOOTH 00159
IC = IC + JOC(I)	SMOOTH 00160
125 CONTINUE	SMOOTH 00161
200 CONTINUE	SMOOTH 00162
ICS = IC	SMOOTH 00163
INO = 0	SMOOTH 00164
DO 400 IX=IBEG,ILIM	SMOOTH 00165
I = IX	SMOOTH 00166
IF(NS.EQ.2) I = IX + IOWAP	SMOOTH 00167
JST = JS(I)	SMOOTH 00168
JEND = JS(I) + JOC(I) - 1	SMOOTH 00169
DO 400 J=JST,JEND	SMOOTH 00170
IC = IC + 1	SMOOTH 00171
IB = IS(J+NCH)	SMOOTH 00172
IT = IB + LOC(J+NCH) - 1	SMOOTH 00173
IF(IX.LT.IB) GO TO 400	SMOOTH 00174
IF(IX.GT.IT) GO TO 400	SMOOTH 00175
INO = INO + 1	SMOOTH 00176
AVPS(INO) = DELPHI(IC)	SMOOTH 00177
X(INO) = I	SMOOTH 00178
Y(INO) = J	SMOOTH 00179
400 CONTINUE	SMOOTH 00180
C	SMOOTH 00181
C ADD THE LEADING EDGE VELOCITY POTENTIAL TO THE AVP ARRAY	SMOOTH 00182
C VEL. POT. = 0. UNLESS IT IS FOR SECOND PLAINFORM IN COPLANAR	SMOOTH 00183
C ANALYSIS	SMOOTH 00184
C	SMOOTH 00185
JLAST = MYBW	SMOOTH 00186
IF(NS.EQ.2) JLAST = MYBT	SMOOTH 00187
DO 600 J=1,JLAST	SMOOTH 00188
IB = IS(J+NCH)	SMOOTH 00189
INO = INO + 1	SMOOTH 00190
X(INO) = FEXLOC(J)	SMOOTH 00191
Y(INO) = J	SMOOTH 00192
AVPS(INO) = (0.,0.)	SMOOTH 00193
IF(NS.EQ.1) GO TO 600	SMOOTH 00194
IF(.NOT.COPLAN) GO TO 600	SMOOTH 00195
XDKVL = (FEXLOC(J+MYBW)-TEXLOC(J)) * K1	SMOOTH 00196
JT = J	SMOOTH 00197
IF(NSUBDV.NE.1) JT = NSUBDV * (J-1) + NSUBCN	SMOOTH 00198
AVPS(INO) = TVP(JT) * CMPLX(COS(XDKVL),-SIN(XDKVL))	SMOOTH 00199
600 CONTINUE	SMOOTH 00200
C	SMOOTH 00201
C CALL LEAST SQUARES SURFACE FITTING ROUTINE	SMOOTH 00202
C	SMOOTH 00203
IDIM = 2	SMOOTH 00204
CN = 1.0	SMOOTH 00205
IDEG = NDEG	SMOOTH 00206
IF(NDEG.NE.0) GO TO 675	SMOOTH 00207
DO 650 I=1,10	SMOOTH 00208
IDEG = 10 - I + 1	SMOOTH 00209
XM = IDEG + 1	SMOOTH 00210

X42 = XM/2.0	SMOOTH 00211
MC = XM * X42 + X42 + 1.0E-04	SMOOTH 00212
MC = (3*MC) / 2	SMOOTH 00213
IF(MC.LE.INO) GO TO 675	SMOOTH 00214
650 CONTINUE	SMOOTH 00215
675 CONTINUE	SMOOTH 00216
CALL FITTER(IDEQ,IND,X,Y,AVPS,A,CN,IDIM)	SMOOTH 00217
C	SMOOTH 00218
EVALUATE THE POLYNOMIAL EQUATION FOR DELPHI	SMOOTH 00219
C	SMOOTH 00220
MDEG = IDEQ + 1	SMOOTH 00221
IC = ICS	SMOOTH 00222
DO 900 IX=IBEG,ILIM	SMOOTH 00223
I = IX	SMOOTH 00224
IF(NS.EQ.2) I = IX + IOWLAP	SMOOTH 00225
XP(1) = 1.	SMOOTH 00226
DO 700 IP=2,MDEG	SMOOTH 00227
700 XP(IP) = XP(IP-1) * FLOAT(I)	SMOOTH 00228
JI = JS(I)	SMOOTH 00229
JT = JOC(I) + JI - 1	SMOOTH 00230
DO 900 J=JI,JT	SMOOTH 00231
IC = IC + 1	SMOOTH 00232
IB = IS(J+NCH)	SMOOTH 00233
IT = IB + NOC(J+NCH) - 1	SMOOTH 00234
IF(IX.LT.1B) GO TO 900	SMOOTH 00235
IF(IX.GT.IT) GO TO 900	SMOOTH 00236
YP(1) = 1.0	SMOOTH 00237
DO 800 JP=2,MDEG	SMOOTH 00238
800 YP(JP) = YP(JP-1) * FLOAT(J)	SMOOTH 00239
VP = A(1)	SMOOTH 00240
IA = 1	SMOOTH 00241
DO 850 L2=2,MDEG	SMOOTH 00242
DO 850 L3=1,L2	SMOOTH 00243
L4 = L2 - L3 + 1	SMOOTH 00244
IA = IA + 1	SMOOTH 00245
VP = VP + XP(L4)*YP(L3)*A(IA)	SMOOTH 00246
850 CONTINUE	SMOOTH 00247
DELPHI(IC) = VP	SMOOTH 00248
900 CONTINUE	SMOOTH 00249
C	SMOOTH 00250
CALCULATE THE TRAILING EDGE VELOCITY POTENTIALS (TVP ARRAY)	SMOOTH 00251
C	SMOOTH 00252
IF(NS.EQ.2) GO TO 910	SMOOTH 00253
NTST = 1	SMOOTH 00254
NTVPS = MYBSW	SMOOTH 00255
JJ = 0	SMOOTH 00256
GO TO 920	SMOOTH 00257
910 CONTINUE	SMOOTH 00258
NTST = NTVPS + 1	SMOOTH 00259
NTVPS = MYBSW + MYBST	SMOOTH 00260
920 CONTINUE	SMOOTH 00261
DO 930 J=NTST,NTVPS	SMOOTH 00262
930 TVP(J) = (0.,0.)	SMOOTH 00263
C	SMOOTH 00264
MDEG = NTST + NSUBD2	SMOOTH 00265
MEND = NTVPS - NSUBD2	SMOOTH 00266
JC = 0	SMOOTH 00267

DO 960 J=NBEG,NEND,NSUBDV	SMOOTH 00268
C J = LOCATION IN THE TYP ARRAY (SUBDIVIDED VALUES)	BCSSMA 00001
C JJ = LOCATION IN ARRAYS TEXLOC, NOC, IS, ETC.	BCSSMA 00002
C JC = UNSUBDIVIDED CHORD NUMBER	BCSSMA 00003
C IROW = ROW NUMBER OF LAST BOX ON THE CHORD	BCSSMA 00004
C I = ROW LOCATION OF IROW IN ARRAY IPNTRM	BCSSMA 00005
JJ = JJ + 1	SMOOTH 00269
JC = JC + 1	SMOOTH 00270
I = TEXLOC(JJ)	SMOOTH 00271
IROW = I	BCSSMA 00005
XINCR = TEXLOC(JJ) - I	SMOOTH 00273
IF (.NOT. COPLAN .AND. J .GT. MYBW) I = I + IOWLAP	SMOOTH 00274
INDB = IPNTRM(1,I) + JC - IPNTRM(2,I)	BCSSMA 00007
C	SMOOTH 00275
C TEST FOR 3 BOXES ON CHORD JJ	SMOOTH 00276
IF(NOC(JJ).LT.3) GO TO 940	SMOOTH 00277
C	SMOOTH 00278
C 2 BOXES AND NO MACH RAY AVAILABLE, OR	BCSSMA 00008
C 3 BOXES OR MORE. DO LINEAR EXTRAPOLATION.	SMOOTH 00279
935 CONTINUE	BCSSMA 00009
INDE = IPNTRM(1,I-1) + JC - IPNTRM(2,I-1)	SMOOTH 00280
SDELPH = DELPHI(INDB) - DELPHI(INDE)	SMOOTH 00281
GO TO 950	SMOOTH 00282
C	SMOOTH 00283
C TEST FOR MACH RAY EXTRAPOLATION.	SMOOTH 00284
940 CONTINUE	SMOOTH 00285
IB = IS(JJ-1)	SMOOTH 00286
IX = IB + NOC(JJ-1) + 1	SMOOTH 00287
IF (IROW .LT. IB .OR. IROW .GT. IX) GO TO 945	BCSSMA 00010
IB = IS(JJ-2)	SMOOTH 00289
IX = IB + NOC(JJ-2) + 1	SMOOTH 00290
IML = IROW-1	BCSSMA 00011
IF (IML .GE. IB .AND. IML .LE. IX) GO TO 948	BCSSMA 00012
C	BCSSMA 00013
C MACH RAY CANNOT BE USED. TEST FOR 2 BOXES ON CHORD JJ	BCSSMA 00014
945 CONTINUE	BCSSMA 00015
IF (NOC(JJ) .LT. 2) GO TO 7010	BCSSMA 00016
GO TO 935	BCSSMA 00017
C	SMOOTH 00293
C MACH RAY CAN BE USED	SMOOTH 00294
948 CONTINUE	BCSSMA 00018
INDA = IPNTRM(1,I-1) + JC - IPNTRM(2,I-1) - 2	SMOOTH 00295
INDC = IPNTRM(1,I) + JC - IPNTRM(2,I) - 1	SMOOTH 00296
SDELPH = 2.0 * DELPHI(INDC) - DELPHI(INDA) - DELPHI(INDB)	SMOOTH 00297
950 CONTINUE	SMOOTH 00298
JT = JJ	SMOOTH 00299
IF(NSUBDV.NE.1) JT = NSUBCN + NSUBDV * (JJ-1)	SMOOTH 00300
TYP(JT) = DELPHI(INDB) + XINCR * SDELPH	SMOOTH 00301
960 CONTINUE	SMOOTH 00302
C	SMOOTH 00303
1000 CONTINUE	SMOOTH 00304
C	SMOOTH 00305
C WRITE THE DELPHI AND TYP ARRAY ON THE NIVPSC FILE	SMOOTH 00306
CALL RDINIT	SMOOTH 00307
ITYPE = SMIXED	SMOOTH 00308
N = IPNTRM(1,NENTRS) - 1	SMOOTH 00309
MXARRY = 6*DELPHI	SMOOTH 00310

CALL WRTMX(NIVPSC,MXWRT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,	SMOOTH 00311
1 DELPHI, ITYPE,2,N,PARM,IRR)	SMOOTH 00312
IF(IRR.NE.0) GO TO 6040	SMOOTH 00313
C	SMOOTH 00314
N = NTVPS	SMOOTH 00315
MXARRY = 6HTVP	SMOOTH 00316
CALL WRTMX(NIVPSC,MXWRT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,	SMOOTH 00317
1 TVP, ITYPE,2,N,PARM,IRR)	SMOOTH 00318
IF(IRR.NE.0) GO TO 6040	SMOOTH 00319
C	SMOOTH 00320
C	SMOOTH 00321
C PRINT THE SMOOTHED VEL. POT. ARRAY	SMOOTH 00322
IF(.NOT.FRVP) GO TO 1500	SMOOTH 00323
TITL(1) = 8K WING	SMOOTH 00324
TITL(2) = 10HVELOCITY P	SMOOTH 00325
TITL(3) = 10HPOTENTIALS	SMOOTH 00326
IF(COPLAN) TITL(1) = 10HWIN/TAIL	SMOOTH 00327
CALL PRINTR(TITL, NM, DELPHI, 1, 1, MXB, MYBW, IPNTRM)	SMOOTH 00328
IF(NSURF.EQ.1.OR.COPLAN) GO TO 1500	SMOOTH 00329
TITL(1) = 8H TAIL	SMOOTH 00330
CALL PRINTR(TITL, NM, DELPHI, 1, 1, IFT, MXST, MYBT,	SMOOTH 00331
1 IPNTRM(1,IOVLAP+1))	SMOOTH 00332
1500 CONTINUE	SMOOTH 00333
IF(.NOT.CHECKPR) GO TO 2000	SMOOTH 00334
WRITE (NT6,9400) (TVP(I),I=1,NTVPS)	SMOOTH 00335
9400 FORMAT(/ (1X,8E16.8))	SMOOTH 00336
C	SMOOTH 00337
2000 CONTINUE	SMOOTH 00338
C	SMOOTH 00339
END FILE NIVPSC	SMOOTH 00340
REWIND NIVPSC	SMOOTH 00341
C CHANGE FILE NAMES	SMOOTH 00342
C	SMOOTH 00343
IAICSC = IVPSC	SMOOTH 00344
IVPSC = NIVPSC	SMOOTH 00345
C	SMOOTH 00346
RETURN	SMOOTH 00347
6010 CONTINUE	SMOOTH 00348
WRITE (NT6,9010) ICEOSC,IRR	SMOOTH 00349
WRITE (NT6,9011) MXARRY	SMOOTH 00350
GO TO 6100	SMOOTH 00351
6020 CONTINUE	SMOOTH 00352
WRITE (NT6,9010) MODESC,IRR	SMOOTH 00353
WRITE (NT6,9011) MXARRY	SMOOTH 00354
GO TO 6100	SMOOTH 00355
6040 CONTINUE	SMOOTH 00356
WRITE (NT6,9010) IVPSC,IRR	SMOOTH 00357
WRITE (NT6,9041) NM	SMOOTH 00358
6100 CONTINUE	SMOOTH 00359
WRITE (NT6,9101) ID(1),ID(2)	SMOOTH 00360
WRITE (NT6,9102) PARM,IPARM	SMOOTH 00361
WRITE (NT6,9103) NFS,NMS	SMOOTH 00362
WRITE (NT6,9104) ITYPE,M,N	SMOOTH 00363
WRITE (NT6,9900)	SMOOTH 00364
GO TO 8000	SMOOTH 00365
C	SMOOTH 00366
7010 CONTINUE	SMOOTH 00367

WRITE (NT6,8020)	SMOOTH	00368
WRITE (NT6,2021) IROM,JC	BCSSMA	00019
C	SMOOTH	00370
8000 CONTINUE	SMOOTH	00371
CALL FLUSH(1)	SMOOTH	00372
C	SMOOTH	00373
9020 FORMAT(73H)*** ERROR - NO TIP TRAILING EDGE VELOCITY POTENTIAL CAN	SMOOTH	00374
1 BE COMPUTED. ***)	SMOOTH	00375
8021 FORMAT(5X,13HCOORDINATES (I2,1H,I2,1H))	SMOOTH	00376
9010 FORMAT(53H)*** ERROR - WHILE READING THE GEOMETRY SCRATCH FILE A10	SMOOTH	00377
1, 15H, ERROR CODE = I4,4H ***)	SMOOTH	00378
9011 FORMAT(5X,32HAN ATTEMPT WAS MADE TO READ THE A6, 8H MATRIX.//)	SMOOTH	00379
9041 FORMAT(5X,54HAN ATTEMPT WAS MADE TO READ THE VEL. POT. ARRAY NUMBE	SMOOTH	00380
1R I3,1H.)	SMOOTH	00381
9090 FORMAT(56H)*** ERROR - WHILE WRITING ON THE VEL. POT. SCRATCH FILE	SMOOTH	00382
1 A10, 15H, ERROR CODE = I4,4H ***)	SMOOTH	00383
9051 FORMAT(5X,36HATTEMPTING TO WRITE VEL. POT. NUMBER I3)	SMOOTH	00384
9101 FORMAT(5X,44MATRIX ID = *, A10, I10)	SMOOTH	00385
9102 FORMAT(5X,44PARAMETERS *,10E11.3, / 10X,*(INTEGER)*, I7, 9I11)	SMOOTH	00386
9103 FORMAT(5X,44FILE SPACING = *,I3,* MATRIX SPACING = *,I3)	SMOOTH	00387
9104 FORMAT(5X,44MATRIX TYPE =*,A10,*, DIMENSIONED (*I4,2H X,I4,1H))	SMOOTH	00388
9900 FORMAT(40 ERROR OCCURRED IN SMOOTHING SECTION. *)	FTND1	00068
END	SMOOTH	00390

	SUBROUTINE PRINTR(TITL,IMODE,ARRAY,K,IXB,MXB,MYB,IPNTRM)	PRNTVP	00002
C		PRNTVP	00003
C	TITL - TITLE TO PRINT FOR THE ARRAY	PRNTVP	00004
C	IMODE - MODE SHAPE NUMBER	PRNTVP	00005
C	ARRAY - ARRAY TO BE PRINTED	PRNTVP	00006
C		PRNTVP	00007
	DIMENSION ARRAY(K,1), TITL(3)	PRNTVP	00008
	COMPLEX ARRAY	PRNTVP	00009
	DIMENSION IPNTRM(2,100)	PRNTVP	00010
	COMMON /CONTRL/ PREVEX,OMACH, TITL(8), PRVGEOM,PRVMODE,DIHW,DIHT,	CONTRL	00002
1	DEFAULT	CONTRL	00003
	LOGICAL PRVGEOM,PRVMODE,DIHW,DIHT,DEFAULT	CONTRL	00004
	COMMON /PROBLM/ OMACH,NMODES,NTSLOP,NKVALS,SMOOTH,NDEG,CRDFIT,	PROBLM	00002
1	EXAIC,SUBDV,PLYWOOD	PROBLM	00003
	LOGICAL SMOOTH,CRDFIT,EXAIC,SUBDV,PLYWOOD	PROBLM	00004
	COMMON /FILES / NT5,NT6,INTAFE,INFSP,NPLAIC,NSPAIC,NOUTP,	FILES	00002
1	IQUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC	FILES	00003
	COMMON /KVAL / IKVAL,XKVAL(20), XKS(20)	KVAL	00002
	DIMENSION PC(2)	PRNTVP	00016
	DIMENSION S(50),D(50)	PRNTVP	00020
	EQUIVALENCE (S(1),BUFF(1)), (D(1),BUFF(1251))	PRNTVP	00021
	REAL K1	PRNTVP	00022
	INTEGER PAGE	PRNTVP	00023
	COMMON /RWBUFF/ BFCODE,IBFCNT, BUFF(3280)	RWBUFF	00002
	DATA PC / 10HPAGE CONTI,4HNUEJ /	FTNXI	00070
	DATA BLANK / 1H /	FTNXI	00071
	DATA XINIT / -1.0 /	FTNXI	00072
	K1 = XKVAL(IKVAL)	PRNTVP	00024
	IF (XKS(IKVAL) .NE. XINIT) K1 = XKS(IKVAL)	PRNTVP	00025
C		PRNTVP	00026
C		PRNTVP	00027
	PAGE = 0	PRNTVP	00028
	N = 1	PRNTVP	00029
	M = 4	PRNTVP	00030
	IF(M.GT.MYB) M = MYB	PRNTVP	00031
100	LINE = 100	PRNTVP	00032
200	DO 1400 I = 1XB,MXB	PRNTVP	00033
	DO 300 J=N,M	PRNTVP	00034
	S(J) = 0.0	PRNTVP	00035
	D(J) = 0.0	PRNTVP	00036
300	CONTINUE	PRNTVP	00037
	IF(LINE.LE.50) GO TO 900	PRNTVP	00038
	PAGE = PAGE + 1	PRNTVP	00039
	LINE = 4	PRNTVP	00040
	WRITE (NT6,9001) TITL,TITL, OMACH, K1, IMODE	PRNTVP	00041
C		PRNTVP	00042
	IF(PAGE.EQ.1) GO TO 700	PRNTVP	00043
	WRITE (NT6,9005) PC	PRNTVP	00044
	GO TO 800	PRNTVP	00045
700	WRITE(NT6,9005)	PRNTVP	00046
800	CONTINUE	PRNTVP	00047
	WRITE(NT6,6006) (BLANK,J,J=N,M)	PRNTVP	00048
	WRITE(NT6,6007) (BLANK, J=N,M)	PRNTVP	00049
900	CONTINUE	PRNTVP	00050
	JS = IPNTRM(2,1)	PRNTVP	00051
	IDX = IPNTRM(1,1)	PRNTVP	00052
	JE = IPNTRM(1,I+1) - IDX + JS - 1	PRNTVP	00053

IF(JE.EQ.0) GO TO 1400	PRNTVP 00054
DO 1000 J=JS,JE	PRNTVP 00055
S(J) = REAL(ARRAY(1,IDX))	PRNTVP 00056
D(J) = AIMAG(ARRAY(1,IDX))	PRNTVP 00057
IDX = IDX +1	PRNTVP 00058
1000 CONTINUE	PRNTVP 00059
DO 1200 J=N,M	PRNTVP 00060
IF(S(J)) 1300,1100,1300	PRNTVP 00061
1100 CONTINUE	PRNTVP 00062
IF(D(J)) 1300,1200,1300	PRNTVP 00063
1200 CONTINUE	PRNTVP 00064
GO TO 1400	PRNTVP 00065
1300 WRITE (NT6,9013) I, (S(J),D(J),J=N,M)	PRNTVP 00066
LINE = LINE + 1	PRNTVP 00067
1400 CONTINUE	PRNTVP 00068
M = M+4	PRNTVP 00069
N = N+4	PRNTVP 00070
IF(N.GT.MYB) GO TO 1500	PRNTVP 00071
IF(M.GT.MYB) M=MYB	PRNTVP 00072
IF(LINE.GT.45) GO TO 100	PRNTVP 00073
WRITE(NT6,6006) (BLANK,J,J=N,M)	PRNTVP 00074
WRITE(NT6,6007) (BLANK, J=N,M)	PRNTVP 00075
LINE = LINE+3	PRNTVP 00076
GO TO 200	PRNTVP 00077
1500 CONTINUE	PRNTVP 00078
RETURN	PRNTVP 00079
9001 FORMAT(11H1,20X,8A10/ 46X,*SMOOTHED *,3A10/ 46X,7H(MACH F5.3,5X,	PRNTVP 00080
1 12HRED. FREQ. =,F8.5, *)* / 52X,**OCE SHAPE*, I3)	PRNTVP 00081
9005 FORMAT(44X,42(1H-),20X,A10,A4)	PRNTVP 00082
6006 FORMAT(4HGRCH, A1,14X,5HCHORD,I3, 3(A1,22X,5HCHORD,I3))	PRNTVP 00083
6007 FORMAT(3X, 4(A1,9X,4HREAL,8X,9HIMAGINARY))	PRNTVP 00084
9013 FORMAT(I4,8E16.8)	PRNTVP 00085
END	PRNTVP 00086

SUBROUTINE FITTER(M,N,X,Y,Z,C,CN,IDIM)	FITTER 00002
DIMENSION X(100), Y(100), Z(100), C(100,66)	FITTER 00003
DIMENSION A1(66), A(66,66), XP(11), YP(11)	FITTER 00004
DIMENSION VS(10)	FITTER 00005
LOGICAL COMPLEX	FITTER 00006
C	FITTER 00007
C M - DEGREE OF POLYNOMIAL EQUATION	FITTER 00008
C N - NUMBER OF DATA POINTS TO FIT CURVE THROUGH	FITTER 00009
C X - X COORDINATE OF DATA POINT	FITTER 00010
C Y - Y COORDINATE OF DATA POINT	FITTER 00011
C Z - Z COORDINATE OF DATA POINT	FITTER 00012
C C - OUTPUT COEFFICIENT ARRAY	FITTER 00013
C CN - SCALE FACTOR	FITTER 00014
C CM - SCALE FACTOR	FITTER 00015
C IDIM - INDICATOR OF REAL OR COMPLEX FUNCTION	FITTER 00016
C = 1, FUNCTION IS REAL	FITTER 00017
C = 2, FUNCTION IS COMPLEX	FITTER 00018
C IF COMPLEX SET DIMENSIONS OF FUNCTION AND COEFFICIENTS	FITTER 00019
C TO (IDIM * —)	FITTER 00020
C	FITTER 00021
C DETERMINE NUMBER OF COEFFICIENTS	FITTER 00022
C	FITTER 00023
EPS = 1.0E-04	FITTER 00024
COMPLEX = .FALSE.	FITTER 00025
IF(IDIM.EQ.2) COMPLEX = .TRUE.	FITTER 00026
C	FITTER 00027
C SCALE DATA TO REDUCE MAGNITUDE OF MATRIX TERMS.	FITTER 00028
C SHOULD AVOID BOMB OUTS DUE TO OVERFLOW CONDITIONS.	FITTER 00029
IF(CN.EQ.0) CN=1.0	FITTER 00030
IF(CN.EQ.1.0) GO TO 15	FITTER 00031
DO 5 I=1,N	FITTER 00032
X(I) = X(I)/CN	FITTER 00033
Y(I) = Y(I)/CN	FITTER 00034
5 CONTINUE	FITTER 00035
15 CONTINUE	FITTER 00036
XM = M + 1	FITTER 00037
XM2 = XM/2.	FITTER 00038
NC = XM*XM2 + XM2 + EPS	FITTER 00039
IF(NC.LE.N) GO TO 25	FITTER 00040
M = M-1	FITTER 00041
GO TO 15	FITTER 00042
25 CONTINUE	FITTER 00043
C	FITTER 00044
C MAC = NC	FITTER 00045
C	FITTER 00046
C DETERMINE THE MAXIMUM DEGREE THAT CAN BE COMPUTED IN	FITTER 00047
C EACH DIRECTION AND SET UP ORDER OF SOLUTION.	FITTER 00048
C	FITTER 00049
NDV = 1	FITTER 00050
NDX = M	FITTER 00051
VS(1) = X(1)	FITTER 00052
DO 60 J=1,N	FITTER 00053
DO 10 J=1,NDV	FITTER 00054
IF(X(J).EQ.VS(J)) GO TO 35	FITTER 00055
30 CONTINUE	FITTER 00056
NDV = NDV + 1	FITTER 00057
VS(NDV) = X(J)	FITTER 00058

IF (NDV-1.EQ.M) GO TO 65	FITTER 00059
55 CONTINUE	FITTER 00060
60 CONTINUE	FITTER 00061
NDX = NDV - 1	FITTER 00062
65 CONTINUE	FITTER 00063
C	FITTER 00064
NDV = 1	FITTER 00065
NDY = M	FITTER 00066
VS(1) = Y(1)	FITTER 00067
DO 80 I=1,N	FITTER 00068
DO 70 J=1,NDV	FITTER 00069
IF (Y(I).EQ.VS(J)) GO TO 75	FITTER 00070
70 CONTINUE	FITTER 00071
NDV = NDV + 1	FITTER 00072
VS(NDV) = Y(I)	FITTER 00073
IF (NDV-1.EQ.M) GO TO 85	FITTER 00074
75 CONTINUE	FITTER 00075
80 CONTINUE	FITTER 00076
NDY = NDV - 1	FITTER 00077
85 CONTINUE	FITTER 00078
C	FITTER 00079
ITOF = NC + 1	FITTER 00080
ITOF1 = ITOF	FITTER 00081
IF (COMPLX) ITOF = ITOF + 1	FITTER 00082
C	FITTER 00083
C	FITTER 00084
C	FITTER 00085
DO 95 I=1,NC	FITTER 00086
C(I) = 0.0	FITTER 00087
IF (.NOT.COMPLX) GO TO 90	FITTER 00088
C(2,I) = 0.0	FITTER 00089
90 CONTINUE	FITTER 00090
DO 95 J=1,ITOF	FITTER 00091
95 A(I,J) = 0.0	FITTER 00092
C	FITTER 00093
C	FITTER 00094
C	FITTER 00095
DETERMINE DEVIATION EQUATION AND SQUARE THE EQUATION	FITTER 00096
AI(1) = 1.0	FITTER 00097
XP(1) = 1.0	FITTER 00098
YP(1) = 1.0	FITTER 00099
MM = M + 1	FITTER 00100
DO 200 K=1,N	FITTER 00101
LO 10 L=2,MM	FITTER 00102
X(L) = XP(L-1)*X(K)	FITTER 00103
YP(L) = YP(L-1)*Y(K)	FITTER 00104
10 CONTINUE	FITTER 00105
C	FITTER 00106
I = 1	FITTER 00107
DO 40 L=2,MM	FITTER 00108
DO 20 LL=1,L	FITTER 00109
IL = L - LL + 1	FITTER 00110
IF (LL-1.GT.MDY) GO TO 30	FITTER 00111
IF (LL-1.GT.MDX) GO TO 20	FITTER 00112
I = I + 1	FITTER 00113
AI(I) = XP(IL)*YP(LL)	FITTER 00114
20 CONTINUE	FITTER 00115
30 CONTINUE	

40 CONTINUE	FITTER 00116
AI(I+1) = Z(1,K)	FITTER 00117
IF(COMPLX) AI(I+2) = Z(2,K)	FITTER 00118
IF(K.GT.1) GO TO 45	FITTER 00119
NC = I	FITTER 00120
ITOT = NC + 1	FITTER 00121
ITOT1 = ITOT	FITTER 00122
IF(COMPLX) ITOT = ITOT + 1	FITTER 00123
45 CONTINUE	FITTER 00124
C	FITTER 00125
DO 1100 I=1,NC	FITTER 00126
DO 1100 J=1,ITOT	FITTER 00127
ASAV = AI(I)*AI(J)	FITTER 00128
A(I,J)=A(I,J)+ASAV	FITTER 00129
1100 CONTINUE	FITTER 00130
200 CONTINUE	FITTER 00131
C	FITTER 00132
C SQUARE ROOT METHOD	FITTER 00133
C INTERMEDIATE MATRIX	FITTER 00134
DO 1200 I=1,NC	FITTER 00135
IM1 = I-1	FITTER 00136
TMP=0.0	FITTER 00137
IF(I.EQ.1) GO TO 1150	FITTER 00138
DO 1120 L=1,IM1	FITTER 00139
1120 TMP= TMP+ A(L,I)*A2	FITTER 00140
1150 CONTINUE	FITTER 00141
T = A(I,I) - TMP	FITTER 00142
IF(T.GT.EPS) GO TO 4	FITTER 00143
A(I,I) = 0.0	FITTER 00144
GO TO 1200	FITTER 00145
4 CONTINUE	FITTER 00146
A(I,I) = SQRT(T)	FITTER 00147
IF(A(I,I).GT.EPS) GO TO 1155	FITTER 00148
A(I,ITOT) = 0.0	FITTER 00149
GO TO 1200	FITTER 00150
1155 CONTINUE	FITTER 00151
C	FITTER 00152
JS = I+1	FITTER 00153
DO 1180 J = JS,ITOT	FITTER 00154
TMP= 0.0	FITTER 00155
IF(I.EQ.1) GO TO 1175	FITTER 00156
DO 1160 L=1,IM1	FITTER 00157
1160 TMP = TMP + A(L,I)*A(L,J)	FITTER 00158
1175 A(I,J) = (A(I,J)-TMP)/A(I,I)	FITTER 00159
1180 CONTINUE	FITTER 00160
1200 CONTINUE	FITTER 00161
C	FITTER 00162
C	FITTER 00163
C BACK SUBSTITUTE FOR COEFFICIENTS	FITTER 00164
DO 1400 K=1,NC	FITTER 00165
I = NC - K + 1	FITTER 00166
IP1=I+1	FITTER 00167
TMP1 = 0.0	FITTER 00168
TMP2 = 0.0	FITTER 00169
IF(A(I,I).GT.EPS) GO TO 1325	FITTER 00170
C(I,I) = 0.0	FITTER 00171
IF(COMPLX) C(2,I) = 0.0	FITTER 00172

```

      GO TO 1400
1325 CONTINUE
      IF(I.EQ.NC) GO TO 1375
      DO 1350 L=1,NC
      TMP1 = TMP1 + A(I,L)*C(1,L)
      IF(.NOT.COMPLX) GO TO 1350
      TMP2 = TMP2 + A(I,L)*C(2,L)
1350 CONTINUE
1375 CONTINUE
      C(1,I) = (A(I,ITOT1)-TMP1)/A(I,I)
      IF(.NOT.COMPLX) GO TO 1400
      C(2,I) = (A(I,ITOT) -TMP2)/A(I,I)
1400 CONTINUE
C
C
C      REORDER THE COEFFICIENTS IN CORRECT POWERS
C      OF X AND Y.
C
      IF(NAC.EQ.NC) GO TO 1475
C
      IZ = 1
      I = 1
      DO 1440 L=2,MM
      DO 1420 LL=1,L
      IL = L -LL +1
      I = I +1
      IF(LL-1.LE.MDY.AND.IL-1.LE.MDX) GO TO 1410
      X(I) = 0.0
      Y(I) = 0.0
      GO TO 1420
1410 CONTINUE
      IZ = IZ + 1
      X(I) = C(1,IZ)
      IF(COMPLX) Y(I) = C(2,IZ)
1420 CONTINUE
1440 CONTINUE
C
      DO 1450 I=2,NAC
      C(1,I) = X(I)
      IF(COMPLX) C(2,I) = Y(I)
1450 CONTINUE
1475 CONTINUE
C
C      ELIMINATE THE SCALE FACTOR FROM THE COEFFICIENTS.
C
      IF(CN.EQ.1.0) GO TO 1700
      I=1
      CP= 1.0/CN
      DO 1600 L1=2,MM
      DO 1500 L2=1,L1
      I = I+1
      C(1,I) = C(1,I)*CP
      C(2,I) = C(2,I)*CP
1500 CONTINUE
      CP= CP/CN
1600 CONTINUE
1700 CONTINUE

```

```

FITTER 00173
FITTER 00174
FITTER 00175
FITTER 00176
FITTER 00177
FITTER 00178
FITTER 00179
FITTER 00180
FITTER 00181
FITTER 00182
FITTER 00183
FITTER 00184
FITTER 00185
FITTER 00186
FITTER 00187
FITTER 00188
FITTER 00189
FITTER 00190
FITTER 00191
FITTER 00192
FITTER 00193
FITTER 00194
FITTER 00195
FITTER 00196
FITTER 00197
FITTER 00198
FITTER 00199
FITTER 00200
FITTER 00201
FITTER 00202
FITTER 00203
FITTER 00204
FITTER 00205
FITTER 00206
FITTER 00207
FITTER 00208
FITTER 00209
FITTER 00210
FITTER 00211
FITTER 00212
FITTER 00213
FITTER 00214
FITTER 00215
FITTER 00216
FITTER 00217
FITTER 00218
FITTER 00219
FITTER 00220
FITTER 00221
FITTER 00222
FITTER 00223
FITTER 00224
FITTER 00225
FITTER 00226
FITTER 00227
FITTER 00228
FITTER 00229

```

C
C THE C ARRAY NOW CONTAINS THE COEFFICIENTS.
C
RETURN
END

FITTER 00230
FITTER 00231
FITTER 00232
FITTER 00233
FITTER 00234

OVERLAY (LHBOX,1,7)	CHORDF	00002
PROGRAM CHORDF	CHORDF	00003
C	CHORDF	00004
C THIS PROGRAM WILL FIT A CURVE THROUGH THE VELOCITY POTENTIALS	CHORDF	00005
C ALONG EACH CHORD INDEPENDENTLY. THESE CURVES WILL BE USED TO	CHORDF	00006
C EVALUATE A SMOOTHED VELOCITY POTENTIAL.	CHORDF	00007
C	CHORDF	00008
COMMON /ARRAYS/ KBXCDW,LBXCDW,LBOXC,KBXCDT,LBXCDT,KJALPH,LJALPH,	ARRAYS	00002
1 KALPHA,KKERNL,LKERNL,KPNTRM,LPNTRM,KDEFSL,KELPHI,	ARRAYS	00003
2 LMODES,KPNTSD,LPTSD,KSDW,LSDW,KPNTDW,LPTDW,	ARRAYS	00004
3 KDW,LDW,KTVP,LTVF	ARRAYS	00005
COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUTP,	FILES	00002
1 IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC	FILES	00003
COMMON /IOCONT/ OPLAIC,OSPAIC,WTGEOM,WTGNMF,WTSL,WTBL,FRBOX,	IOCONT	00002
1 PRPAIC,PRSAIC,PRMODS,PRCOEF,PRDW,PRSW,PRVP,	IOCONT	00003
2 PRBL,PRDCP,PRGNMF,PRGNAC,PRSL,PRLW,PRNW,PRCM	BCSFRB	00001
EBUIVALENCE (PRUW,PRDW)	IOCONT	00005
LOGICAL OPLAIC,OSPAIC,WTGEOM,WTGNMF,WTSL,WTBL,FRBOX,PRPAIC,	IOCONT	00006
1 PRSAIC,PRMODS,PRCOEF,PRDW,PRSW,PRVP,PRBL,PRSL,PRGNMF,	IOCONT	00007
2 PRDCP,PRGNAC,PRUW,PRLW,PRNW,PRCM	BCSFRB	00002
COMMON /PROBLM/ XMACH,NMODES,NTSLOC,NKVALS,SMOOTH,NDEG,CDFIT,	PROBLM	00002
1 EXAIC,SUBDV,PLYWOOD	PROBLM	00003
LOGICAL SMOOTH,CDFIT,EXAIC,SUBDV,PLYWOOD	PROBLM	00004
COMMON /KVAL / IKVAL,XKVAL(20),XKS(20)	KVAL	00002
COMMON /GEOMTY/ COPLAN,NSUBDV,XSUBDV,NSUBD2,NSUBCN,NSURF,	GEOMTY	00002
1 B1,B1BETA,B1S,B1BTAS,WLAX,WLAZ,PSIW,	GEOMTY	00003
2 MXBW,MXBBW,MYBW,MYBBW,MXBSW,MYBSW,MYBBSW,	GEOMTY	00004
3 IXBW,XCENTR	GEOMTY	00005
LOGICAL COPLAN	GEOMTY	00006
COMMON /GEOM2 / TLAX,TLAZ,PSIT,MXBT,MYBT,MYBBT,MXBST,MYBST,	GEOM2	00002
1 MYBBST,IXBT,IXBST,CAPL	GEOM2	00003
COMMON /TAPEIO/ NFS,NMS,LS,NMR,ID(20),NID,ITYPE,LRS,LWS,M,N,	TAPEIO	00002
1 PARM(10),IRR	TAPEIO	00003
DIMENSION IPARM(10)	TAPEIO	00004
EBUIVALENCE (IPARM,IPARM)	TAPEIO	00005
COMMON /CHECKPR/ DPPCPR,GEOPCR,MODCPR,AICCPR,NASCPR,SMCPR,GAFCPR	CHECKPR	00002
LOGICAL DPPCPR,GEOPCR,MODCPR,AICCPR,NASCPR,SMCPR,GAFCPR	CHECKPR	00003
EBUIVALENCE (CHECKPR,SMCPR)	CHORDF	00019
LOGICAL CHECKPR	FTN01	00075
C	CHORDF	00020
C DELPHI(NBOXES), TVP(NCCLS1 + NCCLS2 * NSUBDV)	CHORDF	00021
COMPLEX DELPHI(1000), TVP(250), AVPS(52)	CHORDF	00022
C X(NO. DELPHI + NO. TVP), Y(SAME)	CHORDF	00023
DIMENSION X(52), Y(52)	CHORDF	00024
C A(NO. COEFF.)	CHORDF	00025
COMPLEX A(21)	CHORDF	00026
C	CHORDF	00027
COMMON /INDEX/ IS(100),NOC(100),JS(100),JOC(100)	CHORDF	00028
C PEXLOC((MYBW*MYBT)*NSUBDV), TEXLOC(SAME)	CHORDF	00029
DIMENSION PEXLOC(250), TEXLOC(250)	CHORDF	00030
C IPNTRM(2,NROWS*NSUBDV)	CHORDF	00031
DIMENSION IPNTRM(2,150)	CHORDF	00032
DIMENSION TITL(3)	CHORDF	00033
C	CHORDF	00034
REAL K1	CHORDF	00035
COMPLEX VP, SDELPHI, VC, AVPA, AVPS	CHORDF	00036
LOGICAL MXREAD,MXWRIT,RANDIN,RANDOU	CHORDF	00037

K1 = XXVAL(IKVAL)	CHORDF 00038
MXREAD = .FALSE.	CHORDF 00039
RANDIN = .FALSE.	CHORDF 00040
MXWRIT = .FALSE.	CHORDF 00041
RANDOU = .FALSE.	CHORDF 00042
MXB = MXBW	CHORDF 00043
IF(COPLAN) MXB = MXBT	CHORDF 00044
C	CHORDF 00045
C	CHORDF 00046
C PUT NAME OF SCRATCH FILE FOR SMOOTHED VALUES INTO PLACE	CHORDF 00047
C	CHORDF 00048
NEVPSC = IAICSC	CHORDF 00049
REWIND NEVPSC	CHORDF 00050
C	CHORDF 00051
C GET THE PLAFORM POINTERS FROM THE MODESC FILE	CHORDF 00052
C	CHORDF 00053
REWIND MODESC	CHORDF 00054
CALL RDINIT	CHORDF 00055
ITYPE = SHMIXED	CHORDF 00056
MXARRY = 6H1PNTRM	CHORDF 00057
CALL READMX(MODESC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	CHORDF 00058
1 LRS,IPNTRM,2,NPNTRS,FARM,IRR)	CHORDF 00059
IOMAP = IARM(3)	CHORDF 00060
IF(IRR.NE.0) GO TO 6020	CHORDF 00061
C	CHORDF 00062
CALL RDINIT	CHORDF 00063
ITYPE = SHMIXED	CHORDF 00064
NFS = 1	CHORDF 00065
MXARRY = 6HIS FT.	CHORDF 00066
CALL READMX(MODESC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,	CHORDF 00067
1 LRS,IS,M,N,FARM,IRR)	CHORDF 00068
IF(IRR.NE.0) GO TO 6020	CHORDF 00069
C	CHORDF 00070
C	CHORDF 00071
C READ THE FEXLOC AND TEXLOC ARRAYS FROM THE GEOMETRY SCRATCH	CHORDF 00072
C FILE. THESE ARE NEEDED TO INTERPOLATE VELOCITY POTENTIALS AT	CHORDF 00073
C BOX EDGES.	CHORDF 00074
C	CHORDF 00075
REWIND IGEOSC	CHORDF 00076
CALL RDINIT	CHORDF 00077
NMS = 2	CHORDF 00078
IF(NSURF.EQ.1.OR.COPLAN) NMS=1	CHORDF 00079
ITYPE = SHMIXED	CHORDF 00080
MXARRY = 6HFEXLOC	CHORDF 00081
CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,	CHORDF 00082
1 LRS,FEXLOC,M,N,FARM,IRR)	CHORDF 00083
IF(IRR.NE.0) GO TO 6010	CHORDF 00084
C	CHORDF 00085
CALL RDINIT	CHORDF 00086
ITYPE = SHMIXED	CHORDF 00087
MXARRY = 6HTEXLOC	CHORDF 00088
CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,	CHORDF 00089
1 LRS,TEXLOC,M,N,FARM,IRR)	CHORDF 00090
IF(IRR.NE.0) GO TO 6010	CHORDF 00091
C	CHORDF 00092
C REORDER THE FEXLOC AND TEXLOC ARRAYS SO THAT THERE ARE	CHORDF 00093
C VALUES FOR UNSUBDIVIDED CHORDS ONLY.	CHORDF 00094

C	IF(NSUBDV.EQ.1) GO TO 120	CHORDF 00095
	XSLIDE = NSUBDV - IXBW	CHORDF 00096
	JCOL = NSUBCN	CHORDF 00097
	NCOLS = MYBW + MYBT	CHORDF 00098
	DO 110 I=1,NCOLS	CHORDF 00099
	TEXLOC(I) = (TEXLOC(JCOL) + XSLIDE)/XSUBDV	CHORDF 00100
	FEXLOC(I) = (FEXLOC(JCOL) + XSLIDE)/XSUBDV	CHORDF 00101
	JCOL = JCOL + NSUBDV	CHORDF 00102
	110 CONTINUE	CHORDF 00103
	120 CONTINUE	CHORDF 00104
C		CHORDF 00105
	IFBT = (IXBT-IXBW)/NSUBDV + 1	CHORDF 00106
C		CHORDF 00107
C	LOOP ON NUMBER OF MODES (ALSO NO. OF V.P.)	CHORDF 00108
	REWIND IVPSC	CHORDF 00109
	DO 2000 NM=1,NMODES	CHORDF 00110
C		CHORDF 00111
	READ DELPHI ARRAY FROM IVPSC. THE TVP ARRAY MUST BE SKIPPED	CHORDF 00112
C		CHORDF 00113
	IF NM IS NOT 1	CHORDF 00114
C		CHORDF 00115
	CALL RDINIT	CHORDF 00116
	ITYPE = SMIXED	CHORDF 00117
	NMS = 1	CHORDF 00118
	IF(NM.EQ.1) NMS = 0	CHORDF 00119
	CALL READMX(IVPSC,MYREAD,RANDIN,MFS,NMS,LS,NMR,2,NID,ID,ITYPE,	CHORDF 00120
	1 LRS,DELPHI,M,N,PARM,IRR)	CHORDF 00121
	IF(IRR.NE.0) GO TO 6040	CHORDF 00122
C		CHORDF 00123
C	LOOP ON NUMBER OF CHORDS	CHORDF 00124
C		CHORDF 00125
	NCHRDS = MYBW	CHORDF 00126
	IF(NSURF.EQ.2) NCHRDS = MYBW + MYBT	CHORDF 00127
	DO 1000 J=1,NCHRDS	CHORDF 00128
	NC = 1	CHORDF 00129
	IF(J.GT.MYBW) NC = MYBW + 1	CHORDF 00130
	IST = IS(J)	CHORDF 00131
	NK = IST + NOC(J) - 1	CHORDF 00132
	JSUM = 0	CHORDF 00133
	ITROW = IST	CHORDF 00134
	IF(.NOT.COPLAN.AND.J.GT.MYBW) ITROW = IST + IOVLAP	CHORDF 00135
	DO 100 I=1,ITROW	CHORDF 00136
100	JSUM = JSUM + JOC(I)	CHORDF 00137
	JSUM = JSUM - JOC(ITROW) + 1	CHORDF 00138
	IND = 0	CHORDF 00139
	DO 200 I=IST,NK	CHORDF 00140
	IX = I	CHORDF 00141
	IND = IND + 1	CHORDF 00142
	IF(.NOT.COPLAN.AND.J.GT.MYBW) IX = I + IOVLAP	CHORDF 00143
	ISUB = JSUM + J - JS(IX) - NC + 1	CHORDF 00144
	X(IND) = I	CHORDF 00145
	AVPS(IND) = DELPHI(ISUB)	CHORDF 00146
	JSUM = JSUM + JOC(IX)	CHORDF 00147
	200 CONTINUE	CHORDF 00148
C		CHORDF 00149
C	FIND THE DERIVATIVE OF DELPHI, AND SMOOTH THESE	CHORDF 00150
C		CHORDF 00151

VC = AVPS(1)	CHORDF 00152
INDM1 = INC - 1	CHORDF 00153
AVPA = 0.5 * (AVPS(1) + AVPS(2))	CHORDF 00154
AVPS(1) = (AVPS(2) - AVPS(1)) / (X(2) - X(1))	CHORDF 00155
DO 350 I=2,INDM1	CHORDF 00156
AVPB = 0.5 * (AVPS(I) + AVPS(I+1))	CHORDF 00157
AVPS(I) = AVPB - AVPA	CHORDF 00158
AVPA = AVPB	CHORDF 00159
350 CONTINUE	CHORDF 00160
AVPS(IND) = (AVPS(IND) - AVPA) / 0.5	CHORDF 00161
IND = IND + 1	CHORDF 00162
INDM1 = INDM1 + 1	CHORDF 00163
AVPS(IND) = AVPS(INDM1)	CHORDF 00164
X(IND) = X(INDM1) + 0.5	CHORDF 00165
C	CHORDF 00166
XINC = X(1)	CHORDF 00167
DO 375 I=1,IND	CHORDF 00168
X(I) = X(I) - XINC	CHORDF 00169
375 CONTINUE	CHORDF 00170
IDEG = NDEG	CHORDF 00171
C	CHORDF 00172
C CALL FITTING ROUTINE LEAST SQUARES ERROR CURVE	CHORDF 00173
C	CHORDF 00174
CALL CURVE(IDEG,IND,X,AVPS,A)	CHORDF 00175
C	CHORDF 00176
C EVALUATE THE CURVE FOR SMOOTH DELPHI VALUES	CHORDF 00177
C	CHORDF 00178
MDEG = IDEG + 1	CHORDF 00179
JSUM = 0	CHORDF 00180
DO 400 I=1,ITROW	CHORDF 00181
400 JSUM = JSUM + JOC(I)	CHORDF 00182
C	CHORDF 00183
JSUM = JSUM - JOC(ITROW) + 1	CHORDF 00184
DO 500 I = 1ST,NK	CHORDF 00185
IX = I	CHORDF 00186
IF(.NOT.COPLAN.AND.J.GT.MYBW) IX = I + IOVLAP	CHORDF 00187
ISUB = JSUM + J - JS(IX) - NC + 1	CHORDF 00188
VP = VC	CHORDF 00189
XV = FLOAT(I) - XINC	CHORDF 00190
XP = 1.0	CHORDF 00191
DO 450 L = 1,MDEG	CHORDF 00192
XP = XP * XV	CHORDF 00193
XD = L	CHORDF 00194
XPI = XP / XD	CHORDF 00195
VP = VP + A(L) * XPI	CHORDF 00196
450 CONTINUE	CHORDF 00197
DELPHI(ISUB) = VP	CHORDF 00198
JSUM = JSUM + JOC(IX)	CHORDF 00199
500 CONTINUE	CHORDF 00200
C	CHORDF 00201
C CALCULATE THE TRAILING EDGE VELOCITY POTENTIALS (TVP ARRAY)	CHORDF 00202
C	CHORDF 00203
L = J	CHORDF 00204
IF(NSUBDV.NE.1) L = NSUBDV * (J-1) + NSUBCN	CHORDF 00205
TVP(L) = (0.,0.)	CHORDF 00206
JJ = J	CHORDF 00207
JC = J	CHORDF 00208

IF(J.GT.MYBW) JC = J - MYBW	CHORDF 00209
I = TEXLOC(JJ)	CHORDF 00210
XINCR = TEXLOC(JJ) - I	CHORDF 00211
IROW = I	BCSCFA 00001
IF(.NOT.COPLAN.AND.J.GT.MYBW) I = I + IOMLAP	CHORDF 00212
INDB = IPNTRM(1,I) + JC - IPNTRM(2,I)	CHORDF 00213
C	CHORDF 00214
C TEST FOR 3 BOXES ON CHORD JJ	CHORDF 00215
IF(NOC(JJ).LT.3) GO TO 940	CHORDF 00216
C	CHORDF 00217
C 2 BOXES AND NO MACH RAY AVAILABLE, OR	BCSCFA 00002
C 3 BOXES OR MORE. DO LINEAR EXTRAPOLATION.	CHORDF 00218
900 CONTINUE	BCSCFA 00003
INDE = IPNTRM(1,I-1) + JC - IPNTRM(2,I-1)	CHORDF 00219
SDELPH = DELPHI(INDB) - DELPHI(INDE)	CHORDF 00220
GO TO 950	CHORDF 00221
C	CHORDF 00222
C TEST FOR MACH RAY EXTRAPOLATION.	CHORDF 00223
940 CONTINUE	CHORDF 00224
IB = IS(JJ-1)	CHORDF 00225
IX = IB + NOC(JJ-1) + 1	CHORDF 00226
IF (IROW .LT. IB .OR. IROW .GT. IX) GO TO 945	BCSCFA 00004
IB = IS(JJ-2)	CHORDF 00228
IX = IB + NOC(JJ-2) + 1	CHORDF 00229
IMI = IROW-1	BCSCFA 00005
IF (IMI .GE. IB .AND. IMI .LE. IX) GO TO 948	BCSCFA 00006
C	BCSCFA 00007
C MACH RAY CANNOT BE USED. TEST FOR 2 BOXES ON CHORD JJ	BCSCFA 00008
945 CONTINUE	BCSCFA 00009
IF (NOC(JJ) .LT. 2) GO TO 7010	BCSCFA 00010
GO TO 900	BCSCFA 00011
C	CHORDF 00232
C MACH RAY CAN BE USED	CHORDF 00233
948 CONTINUE	BCSCFA 00012
INDA = IPNTRM(1,I-1) + JC - IPNTRM(2,I-1) - 2	CHORDF 00234
INDC = IPNTRM(1,I) + JC - IPNTRM(2,I) - 1	CHORDF 00235
SDELPH = 2.0*DELPHI(INDC) - DELPHI(INDA) - DELPHI(INDB)	CHORDF 00236
950 CONTINUE	CHORDF 00237
JT = JJ	CHORDF 00238
IF(NSUBDV.NE.1) JT = NSUBDV * (JJ-1) + NSUBCN	CHORDF 00239
TVP(JT) = DELPHI(INDB) + XINCR*SDELPH	CHORDF 00240
980 CONTINUE	CHORDF 00241
C	CHORDF 00242
1000 CONTINUE	CHORDF 00243
C	CHORDF 00244
C WRITE THE DELPHI AND TVP ARRAY ON THE NIVPSC FILE	CHORDF 00245
CALL RDINIT	CHORDF 00246
ITYPE = SHMIXED	CHORDF 00247
N = IPNTRM(1,NPACTS) - 1	CHORDF 00248
MXARRY = @DELPHI	CHORDF 00249
CALL WRTEXX(NIVPSC,WRWRIT,RANXCU,NFS,NAB,LS,NMR,LWS,2,10,	CHORDF 00250
1 DELPHI,ITYPE,2,N,PARM,IRR)	CHORDF 00251
IF(IRR.NE.0) GO TO 6040	CHORDF 00252
C	CHORDF 00253
MTVPS = NSUBDV * NCHADS	CHORDF 00254
MTVPS	CHORDF 00255
MXARRY = @MTVPS	CHORDF 00256

CALL WRTMX(NIVPSC,MXWRT,RANDOU,NFS,NMS,LS,NMR,LWS,2,10,	CHORDF	00257
1 TVP, ITYPE,2,N,PARM,IRR)	CHORDF	00258
IF(IRR.NE.0) GO TO 6040	CHORDF	00259
C	CHORDF	00260
C PRINT THE SMOOTHED VEL. POT. ARRAY	CHORDF	00261
C	CHORDF	00262
IF(.NOT.PRVP) GO TO 1500	CHORDF	00263
TITL(1) = 8H WING	CHORDF	00264
TITL(2) = 10HVELOCITY P	CHORDF	00265
TITL(3) = 10HPOTENTIALS	CHORDF	00266
IF(COPLAN) TITL(1) = 10HWING/TAIL	CHORDF	00267
CALL PRINTR(TITL,NM,DELPHI,1,1,MXB,MYBW,IPNTRM)	CHORDF	00268
IF(NSURF.EQ.1.OR.COPLAN) GO TO 1500	CHORDF	00269
TITL(1) = 8H TAIL	CHORDF	00270
CALL PRINTR(TITL,NM,DELPHI,1,IFBT,MXBT,MYBT,IPNTRM(1,IOVLAP+1))	CHORDF	00271
1500 CONTINUE	CHORDF	00272
IF(.NOT.CHECKPR) GO TO 2000	CHORDF	00273
WRITE(NT6,940C) (TVP(I),I=1,NTVPS)	CHORDF	00274
9400 FORMAT(// (1X,8E16.8))	CHORDF	00275
C	CHORDF	00276
2000 CONTINUE	CHORDF	00277
C	CHORDF	00278
END FILE NIVPSC	CHORDF	00279
REWIND NIVPSC	CHORDF	00280
C CHANGE FILE NAMES	CHORDF	00281
C	CHORDF	00282
IAICSC = IVPSC	CHORDF	00283
IVPSC = NIVPSC	CHORDF	00284
C	CHORDF	00285
RETURN	CHORDF	00286
6010 CONTINUE	CHORDF	00287
WRITE(NT6,9010) IGEOSC,IRR	CHORDF	00288
WRITE(NT6,9011) MXARRY	CHORDF	00289
GO TO 6100	CHORDF	00290
6020 CONTINUE	CHORDF	00291
WRITE(NT6,9010) MODESC,IRR	CHORDF	00292
WRITE(NT6,9011) MXARRY	CHORDF	00293
GO TO 6100	CHORDF	00294
6040 CONTINUE	CHORDF	00295
WRITE(NT6,9010) IVPSC,IRR	CHORDF	00296
WRITE(NT6,9041) NM	CHORDF	00297
6100 CONTINUE	CHORDF	00298
WRITE(NT6,9101) ID(1),ID(2)	CHORDF	00299
WRITE(NT6,9102) PARM,IPARM	CHORDF	00300
WRITE(NT6,9103) NFS,NMS	CHORDF	00301
WRITE(NT6,9104) ITYPE,M,N	CHORDF	00302
WRITE(NT6,9900)	CHORDF	00303
GO TO 8000	CHORDF	00304
C	CHORDF	00305
7010 CONTINUE	CHORDF	00306
WRITE(NT6,8020)	CHORDF	00307
WRITE(NT6,8021) IROW,JC	BCSCFA	00013
C	CHORDF	00309
8000 CONTINUE	CHORDF	00310
CALL FLUSH(1)	CHORDF	00311
C	CHORDF	00312
8020 FORMAT(73H*** ERROR - NO TIP TRAILING EDGE VELOCITY POTENTIAL CAN	CHORDF	00313

1 BE COMPUTED. ***)	CHORDF 00314
8021 FORMAT(5X,13HCOORDINATES (I2,1H,I2,1H))	CHORDF 00315
9010 FORMAT(53H0*** ERROR - WHILE READING THE GEOMETRY SCRATCH FILE A10	CHORDF 00316
1, 15H, ERROR CODE = I4,4H ***)	CHORDF 00317
9011 FORMAT(5X,32HAN ATTEMPT WAS MADE TO READ THE A6, 8H MATRIX.//)	CHORDF 00318
9041 FORMAT(5X,54HAN ATTEMPT WAS MADE TO READ THE VEL. POT. ARRAY NUMBE	CHORDF 00319
1R I3,1H.)	CHORDF 00320
9050 FORMAT(56H0*** ERROR - WHILE WRITING ON THE VEL. POT. SCRATCH FILE	CHORDF 00321
1 A10, 15H, ERROR CODE = I4,4H ***)	CHORDF 00322
9051 FORMAT(5X,36HATTEMPTING TO WRITE VEL. POT. NUMBER I3)	CHORDF 00323
9101 FORMAT(5X,**MATRIX ID = *, A10, I10)	CHORDF 00324
9102 FORMAT(5X,*PARAMETERS *,10E11.3, / 10X,*(INTEGER)*, I7, 9I11)	CHORDF 00325
9103 FORMAT(5X,*FILE SPACING = *,I3,* MATRIX SPACING = *,I3)	CHORDF 00326
9104 FORMAT(5X,**MATRIX TYPE =*,A10,*, DIMENSIONED (*I4,2H X,I4,1H))	CHORDF 00327
9900 FORMAT(*0 ERROR OCCURRED IN CHORD-FIT SMOOTHING SECTION. *)	FTNX1 00074
END	CHORDF 00329

	SUBROUTINE PRINTR(TITL,IMODE,ARRAY,K,IXB,MXB,MYB,IPNTRM)	PRINTR	00002
C		PRINTR	00003
C	TITL - TITLE TO PRINT FOR THE ARRAY	PRINTR	00004
C	IMODE - MODE SHAPE NUMBER	PRINTR	00005
C	ARRAY - ARRAY TO BE PRINTED	PRINTR	00006
C		PRINTR	00007
	DIMENSION ARRAY(K,1), TITL(3)	PRINTR	00008
	COMPLEX ARRAY	PRINTR	00009
	COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSFAIC,NOUTP,	FILES	00002
1	IQUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC	FILES	00003
	COMMON /CONTRL/ PREVEX,OMACH, TITL(8), PRVGEOM,PRVMODE,DIHW,DIHT,	CONTRL	00002
1	DEFAULT	CONTRL	00003
	LOGICAL PRVGEOM,PRVMODE,DIHW,DIHT,DEFAULT	CONTRL	00004
	COMMON /PROBLM/ XMACH,NMODES,NTSLOF,NKVALS,SMOOTH,NDEG,CRDFIT,	PROBLM	00002
1	EXAIC,SUBDV,PLYWOOD	PROBLM	00003
	LOGICAL SMOOTH,CRDFIT,EXAIC,SUBDV,PLYWOOD	PROBLM	00004
	COMMON /KVAL / IKVAL,XKVAL(20), XKS(20)	KVAL	00002
	DIMENSION IPNTRM(2,50)	PRINTR	00015
	DIMENSION PC(2)	PRINTR	00016
	DIMENSION S(50),D(50)	PRINTR	00020
	EQUIVALENCE (S(1),BUFF(1)), (D(1),BUFF(1251))	PRINTR	00021
	REAL K1	PRINTR	00022
	INTEGER PAGE	PRINTR	00023
	COMMON /RWBUFF/ BFCCDE,IBFCNT, BUFF(3280)	RWBUFF	00002
	DATA PC / 10HPAGE CONTI,4HNUE /	FTNX1	00076
	DATA BLANK / 1H /	FTNX1	00077
	DATA XINIT / -1.0 /	FTNX1	00078
	K1 = XKVAL(IKVAL)	PRINTR	00024
	IF(XKS(IKVAL).NE.XINIT) K1 = XKS(IKVAL)	PRINTR	00025
	PAGE = 0	PRINTR	00026
	N = 1	PRINTR	00027
	M = 4	PRINTR	00028
	IF(M.GT.MYB) M = MYB	PRINTR	00029
100	LINE = 100	PRINTR	00030
200	CONTINUE	PRINTR	00031
	DO 1400 I=IXB,MXB	PRINTR	00032
	DO 300 J=N,M	PRINTR	00033
	S(J) = 0.0	PRINTR	00034
	D(J) = 0.0	PRINTR	00035
300	CONTINUE	PRINTR	00036
	IF(LINE.LE.50) GO TO 900	PRINTR	00037
	PAGE = PAGE + 1	PRINTR	00038
	LINE = 4	PRINTR	00039
	WRITE (NT6,9001) TITL,TITL,XMACH,K1,IMODE	PRINTR	00040
C		PRINTR	00041
	IF(PAGE.EQ.1) GO TO 700	PRINTR	00042
	WRITE (NT6,9005) PC	PRINTR	00043
	GO TO 800	PRINTR	00044
700	WRITE(NT6,9005)	PRINTR	00045
800	CONTINUE	PRINTR	00046
	WRITE(NT6,6006) (BLANK,J,J=N,M)	PRINTR	00047
	WRITE(NT6,6007) (BLANK, J=N,M)	PRINTR	00048
900	CONTINUE	PRINTR	00049
	JS = IPNTRM(2,I)	PRINTR	00050
	IF(JS .LE. 0) GO TO 1400	PRINTR	00051
	IDX = IPNTRM(1,I)	PRINTR	00052
	JE = IPNTRM(1,I+1) - IDX + JS - 1	PRINTR	00053

IF(JE.EQ.0) GO TO 1400	PRINTR 00054
DO 1000 J=JS,JE	PRINTR 00055
S(J) = REAL(ARRAY(1,IDX))	PRINTR 00056
D(J) = AIMAG(ARRAY(1,IDX))	PRINTR 00057
IDX = IDX +1	PRINTR 00058
1000 CONTINUE	PRINTR 00059
DO 1200 J=N,M	PRINTR 00060
IF(S(J)) 1300,1100,1300	PRINTR 00061
1100 CONTINUE	PRINTR 00062
IF(D(J)) 1300,1200,1300	PRINTR 00063
1200 CONTINUE	PRINTR 00064
GO TO 1400	PRINTR 00065
1300 WRITE (NT6,9013) I, (S(J),D(J),J=N,M)	PRINTR 00066
LINE = LINE + 1	PRINTR 00067
1400 CONTINUE	PRINTR 00068
M = M+4	PRINTR 00069
N = N+4	PRINTR 00070
IF(N.GT.MYB) GO TO 1500	PRINTR 00071
IF(M.GT.MYB) M = MYB	PRINTR 00072
IF(LINE.GT.45) GO TO 100	PRINTR 00073
WRITE(NT6,6006) (BLANK,J,J=N,M)	PRINTR 00074
WRITE(NT6,6007) (BLANK, J=N,M)	PRINTR 00075
LINE = LINE+3	PRINTR 00076
GO TO 200	PRINTR 00077
1500 CONTINUE	PRINTR 00078
RETURN	PRINTR 00079
9001 FORMAT(1H1,20X,8A10/ 46X,*SMOOTHED *,3A10/ 46X,7H(MACH F5.3,5X,	PRINTR 00080
1 12HRED. FREQ. =,F8.5, *)* / 52X,*MODE SHAPE*, 13)	PRINTR 00081
9005 FORMAT(44X,42(1H-),20X,A10,A4)	PRINTR 00082
6006 FORMAT(4HOROW, A1,14X,5HCHORD,I3, 3(A1,22X,5HCHORD,I3))	PRINTR 00083
6007 FORMAT(3X, 4(A1,9X,4HREAL,8X,9HIMAGINARY))	PRINTR 00084
9013 FORMAT(14,8E16.8)	PRINTR 00085
END	PRINTR 00086

```

SUBROUTINE CURVE(M,N,X,Z,C)
  DIMENSION X(50), Z(2,50), C(2,21)
  DIMENSION AI(23), A(21,23), XP(5)

C
C      M - DEGREE OF POLYNOMIAL EQUATION
C      N - NUMBER OF DATA POINTS TO FIT CURVE THROUGH
C      X - X COORDINATE OF DATA POINT
C      Z - Z COORDINATE
C      C - OUTPUT COEFFICIENT ARRAY
C
  EPS = 1.0E-04
  IF(N.LT.M+1) M = N-1
  NC = M + 1

C
C      ZERO OUT THE ARRAYS NEEDED
C
  ITOT = NC + 2
  DO 100 I=1,NC
    C(1,I) = 0.0
    C(2,I) = 0.0
  50 CONTINUE
  DO 100 J=1,ITOT
    A(I,J) = 0.0
  100 CONTINUE

C
C      DETERMINE DEVIATION EQUATION AND SQUARE THE EQUATION
C
  AI(1) = 1.00
  DO 400 K=1,N
    AI(1) = 1.0
    DO 200 L=2,NC
      AI(L) = AI(L-1) * X(K)
    200 CONTINUE
    AI(NC+1) = Z(1,K)
    AI(NC+2) = Z(2,K)
  400 CONTINUE

C
  DO 300 I=1,NC
    DO 300 J=1,ITOT
      ASAV = AI(I) * AI(J)
      A(I,J) = A(I,J) + ASAV
    300 CONTINUE
  400 CONTINUE

C
C      SQUARE ROOT METHOD INTERMEDIATE MATRIX
C
  DO 1200 I=1,NC
    IM1 = I-1
    TMP = 0.0
    IF(I.EQ.1) GO TO 600
    DO 500 L=1,IM1
      500 TMP = TMP + A(L,I) ** 2
    600 CONTINUE
    T = A(I,I) - TMP
    IF(T.GT.EPS) GO TO 700
    A(I,I) = 0.0
    GO TO 1200
  700 CONTINUE

```

```

CURVE 00002
CURVE 00003
CURVE 00004
CURVE 00005
CURVE 00006
CURVE 00007
CURVE 00008
CURVE 00009
CURVE 00010
CURVE 00011
CURVE 00012
CURVE 00013
CURVE 00014
CURVE 00015
CURVE 00016
CURVE 00017
CURVE 00018
CURVE 00019
CURVE 00020
CURVE 00021
CURVE 00022
CURVE 00023
CURVE 00024
CURVE 00025
CURVE 00026
CURVE 00027
CURVE 00028
CURVE 00029
CURVE 00030
CURVE 00031
CURVE 00032
CURVE 00033
CURVE 00034
CURVE 00035
CURVE 00036
CURVE 00037
CURVE 00038
CURVE 00039
CURVE 00040
CURVE 00041
CURVE 00042
CURVE 00043
CURVE 00044
CURVE 00045
CURVE 00046
CURVE 00047
CURVE 00048
CURVE 00049
CURVE 00050
CURVE 00051
CURVE 00052
CURVE 00053
CURVE 00054
CURVE 00055
CURVE 00056
CURVE 00057
CURVE 00058

```

```

      A(I,I) = SORT(T)
      IF(A(I,I).GT.EPS) GO TO 800
      A(I,ITOT) = 0.0
      GO TO 1200
800  CONTINUE
C
      JS = I+1
      DO 1100 J=JS,ITOT
      TMP = 0.0
      IF(I.EQ.1) GO TO 1000
      DO 900 L=1,IM1
900  TMP = TMP + A(L,I)*A(L,J)
1000 A(I,J) = (A(I,J)-TMP)/A(I,I)
1100 CONTINUE
1200 CONTINUE
C
      BACK SUBSTITUTE FOR COEFFICIENTS
C
      DO 1600 K=1,NC
      I = NC - K + 1
      IP1 = I + 1
      TMP1 = 0.0
      TMP2 = 0.0
      IF(A(I,I).GT.EPS) GO TO 1300
      C(1,I) = 0.0
      C(2,I) = 0.0
      GO TO 1600
1300 CONTINUE
      IF(I.EQ.NC) GO TO 1500
      DO 1400 L=IP1,NC
      TMP1 = TMP1 + A(I,L) * C(1,L)
      TMP2 = TMP2 + A(I,L) * C(2,L)
1400 CONTINUE
1500 CONTINUE
      C(1,I) = (A(I,NC+1)-TMP1)/A(I,I)
      C(2,I) = (A(I,ITOT)-TMP2)/A(I,I)
1600 CONTINUE
      RETURN
      END

```

```

CURVE 00059
CURVE 00060
CURVE 00061
CURVE 00062
CURVE 00063
CURVE 00064
CURVE 00065
CURVE 00066
CURVE 00067
CURVE 00068
CURVE 00069
CURVE 00070
CURVE 00071
CURVE 00072
CURVE 00073
CURVE 00074
CURVE 00075
CURVE 00076
CURVE 00077
CURVE 00078
CURVE 00079
CURVE 00080
CURVE 00081
CURVE 00082
CURVE 00083
CURVE 00084
CURVE 00085
CURVE 00086
CURVE 00087
CURVE 00088
CURVE 00089
CURVE 00090
CURVE 00091
CURVE 00092
CURVE 00093
CURVE 00094
CURVE 00095
CURVE 00096
CURVE 00097

```

	OVERLAY (AFMBOX,1,10)	FORCES	00002
	PROGRAM FORCES	FORCES	00003
C		FORCES	00004
C	THIS PROGRAM CALCULATES BOX LIFTS, SECTION LIFTS, TOTAL LIFT,	FORCES	00005
C	AND GENERALIZED AIR FORCES. THE PROGRAM MUST READ INFORMATION	FORCES	00006
C	FROM THE GEOMETRY SCRATCH FILE AND THE MODE SCRATCH FILE.	FORCES	00007
C		FORCES	00008
	COMMON /ARRAYS/ KBXCDW,LBXCDW,LBOXC,KBXCDT,LBXCDT,KJALPH,LJALPH,	ARRAYS	00002
1	KALPHA,KKERNL,LKERNL,KPNTRM,LPNTRM,KDEFSL,KELPHI,	ARRAYS	00003
2	LMODES,KPNTSD,LPTSL,KSDW,LSDW,KPNTDW,LPTDW,	ARRAYS	00004
3	KDW,LDW,KTVP,LTVF	ARRAYS	00005
	COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUFP,	FILES	00002
1	IQUFSP,MODESC,IVPSC,IGEOBC,IWTFSC,IAICSC	FILES	00003
	EQUIVALENCE (IWTFSC,ITSLSC)	FORCES	00011
	COMMON /IOCONT/ OPLAIC,OSPAIC,WTGEOM,WTGNF,WTSL,WTBL,FRBOX,	IOCONT	00002
1	FRPAIC,FRSAIC,FRMDS,FRCOEF,FRDW,FRSW,FRVP,	IOCONT	00003
2	FRBL,FRDCP,FRGNF,FRGNAC,FRSL,FRWL,FRNW,FRCM	BCSFRB	00001
	EQUIVALENCE (FRWL,FRDW)	IOCONT	00005
	LOGICAL OPLAIC,OSPAIC,WTGEOM,WTGNF,WTSL,WTBL,FRBOX,FRPAIC,	IOCONT	00006
1	FRSAIC,FRMDS,FRCOEF,FRDW,FRSW,FRVP,FRBL,FRSL,FRGNF,	IOCONT	00007
2	FRDCP,FRGNAC,FRWL,FRWL,FRNW,FRCM	BCSFRB	00002
	COMMON / KERN / ERR,MXSKRN,IPKERN,NPLKRN,NSPATK,NRONEA	KERN	00002
	COMMON /KVAL / IKVAL,XKVAL(20),XKS(20)	KVAL	00002
	COMMON /PROBLM/ XMACH,NMODES,NTSLOF,NKVALS,SMOOTH,NDEG,CDFIT,	PROBLM	00002
1	EXAIC,SUBDV,PLYWOOD	PROBLM	00003
	LOGICAL SMOOTH,CDFIT,EXAIC,SUBDV,PLYWOOD	PROBLM	00004
	COMMON / MODES/ SYM,SYMT,MTYPEW,MTYPEF	MODECOM	00002
	COMMON /GEOMTY/ COPLAN,NSUBDV,XSUBDV,NSUBD2,NSUBCN,NSURF,	GEOMTY	00002
1	B1,B1BETA,B1S,B1BTAS,WLAX,WLAZ,PSIW,	GEOMTY	00003
2	MXBW,MXBBW,MYBW,MYBBW,MXESW,MYBSW,MYBBSW,	GEOMTY	00004
3	IXBW,XCENTR	GEOMTY	00005
	LOGICAL COPLAN	GEOMTY	00006
	COMMON /GEOM2 / TLAX,TLAZ,PSIT,MXBT,MYBT,MYBDT,MXST,MYST,	GEOM2	00002
1	MYBBST,IXBT,IXBST,CAPL	GEOM2	00003
	COMMON /TAPEIO/ NFS,NMS,LS,NMR,IND(20),NID,ITYPE,LRS,LWS,N,N,	TAPEIO	00002
1	PARM(10),IRR	TAPEIO	00003
	DIMENSION IARM(10)	TAPEIO	00004
	EQUIVALENCE (IARM,IARM)	TAPEIO	00005
	COMMON /CHECKFR/ DPPCFR,GEOPFR,MODCFR,AICCFR,NMSCFR,SMCFR,GAFCFR	CHECKFR	00002
	LOGICAL DPPCFR,GEOPFR,MODCFR,AICCFR,NMSCFR,SMCFR,GAFCFR	CHECKFR	00003
	EQUIVALENCE (CHECKFR,GAFCFR)	FORCES	00022
	LOGICAL CHECKFR	FORCES	00023
	DIMENSION RWBF(1250)	FORCES	00024
	EQUIVALENCE (RWBF,BUFF)	FORCES	00025
	COMPLEX RWBF	FORCES	00026
	COMMON /LOCL/ XKVL	FORCES	00027
	COMMON BLANK(1)	FORCES	00028
	DIMENSION TITL(2)	FORCES	00029
C		FORCES	00030
C	FELOC((MYBW*MYBT)*NSUBDV),TEXLOC(SAME)	FORCES	00031
	DIMENSION FELOC(250),TEXLOC(250)	FORCES	00032
C	IPNTRM(2,NROWS)	FORCES	00033
	DIMENSION IPNTRM(2,100)	FORCES	00034
C	IBOXW(NROWS,150/20),IBOXT(90,150/20)	FORCES	00035
	DIMENSION IBOXW(150,8),IBOXT(90,8)	FORCES	00036
C	IBXCDF(NCCLS),IBXCD(NCCLS),IBXCDA(NCCLS)	FORCES	00037
	DIMENSION IBXCDF(150),IBXCD(150),IBXCDA(150)	FORCES	00038

C	DEFSL(2,NBOXES), DEFLTE(NCOLS)	FORCES	00039
	DIMENSION DEFSL(2,1000), DEFLTE(50)	FORCES	00040
C	ALPHA(NCOLS*2*NSURF), IJALPH(SAME)	FORCES	00041
C	TSLFN(NBOXES)	FORCES	00042
	DIMENSION TSLFN(1000)	FORCES	00043
	DIMENSION ALPHA(200), IJALPH(200)	FORCES	00044
C	DELPHI(NBOXES), TVP(NCOLS1+NCOLS2*NSUBDV)	FORCES	00045
	COMPLEX DELPHI(1000), TVP(250)	FORCES	00046
C	BXLIFT(NBOXES), SLIFT(NCOLS*NMODES), GENAF(NMODES*NMODE)	FORCES	00047
	COMPLEX BXLIFT(1000), SLIFT(100), TLIFT, GENAF(400),	FORCES	00048
C	AFROW(NMODES)	FORCES	00049
1	AFROW(20), BL2, TLIFT1, TLIFT2	FORCES	00050
C	DELCP(NBOXES)	FORCES	00051
	COMPLEX DELCP(1000)	FORCES	00052
C	GRAFC(NMODES*NMODES), GPPAFC(SAME)	FORCES	00053
	DIMENSION GRAFC(400), GPPAFC(400)	FORCES	00054
C		FORCES	00055
	DIMENSION AFC(2)	FORCES	00056
	EQUIVALENCE (AFCSTR,AFC)	FORCES	00057
C		BCSFRB	00008
	COMPLEX SECMON(100), GAF	BCSFRB	00009
C		BCSFRB	00010
C	VPTE(NCOLS)	FORCES	00058
	COMPLEX VPTE(50), VPLE, TEMP1, TEMP2, TEMP3, BL	FORCES	00059
	LOGICAL MREAD,RANDIN,MWRIT,RANDOU	FORCES	00060
	LOGICAL BLNEED	FORCES	00061
	COMPLEX XINDEF	FORCES	00062
	DIMENSION XINDEF(2)	FTN01	00079
	EQUIVALENCE (XINDEF,XINDEF)	FTN01	00080
	COMMON /RWBUFF/ BFCODE,IBFCNT, BUFF(3280)	RWBUFF	00082
	MREAD = .FALSE.	FORCES	00064
	RANDIN = .FALSE.	FORCES	00065
	MWRIT = .FALSE.	FORCES	00066
	RANDOU = .FALSE.	FORCES	00067
C		FORCES	00068
C		FORCES	00069
	XKVL = XKVAL(IKVAL)	FORCES	00070
	BLNEED = WTBL .OR. PRBL .OR. PRSL .OR. PRDCP .OR. PRCH	BCSFRB	00011
	TWOBT = (2.0*B1BETA)/B1	FORCES	00072
	TWOBB = TWOBT/B1	FORCES	00073
C	CONSTANTS FOR AGARD GENERALIZED AERODYNAMIC COEFFICIENTS,	BCSFRA	00001
C	BASED ON WING SEMI-SPAN	BCSFRA	00002
	S = MYBW * B1BETA	BCSFRA	00003
	S3 = S*S*S	FORCES	00076
	S4 = S*S*S	FORCES	00077
	BS3BET = -B1BETA/S3	FORCES	00078
	BK84BT = 0.0	BCSFRA	00004
	IF (XKVL .EQ. 0.) GO TO 5	BCSFRA	00005
	BK84BT = -1. *(B1*B1BETA)/(XKVL*S4)	FORCES	00079
5	CONTINUE	BCSFRA	00006
	NVPS = NMODES	FORCES	00082
	MAX = NVPS * NMODES	FORCES	00083
C		FORCES	00084
C		FORCES	00085
C		FORCES	00086
	REWING MODESC	FORCES	00087
	NMSPE = 0	FORCES	00088

C		FORCES	00089
C	READ THE POINTERS FROM THE MODESC FILE.	FORCES	00090
C		FORCES	00091
	CALL RDINIT	FORCES	00092
	ITYPE = SHMIXED	FORCES	00093
	MXARRY = GHIPNTRM	FORCES	00094
	CALL READMX(MODESC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	FORCES	00095
1	LRS,IPNTRM,M,N,FARM,IRR)	FORCES	00096
	ICVLAP = IFARM(3)	FORCES	00097
	NPNTRS = N	FORCES	00098
	MXB = NPNTRS -1	FORCES	00099
	MYB = MAXD(MYBW,MYBT)	FORCES	00100
	NBOXES = MYB * MXB	FORCES	00101
	IF(IRR.NE.0) GO TO 6020	FORCES	00102
C		FORCES	00103
C	REWIND IGEOGC	FORCES	00104
		FORCES	00105
C		FORCES	00106
C	READ BOX CODES INTO STORAGE FROM GEOMETRY SCRATCH FILE	FORCES	00107
C		FORCES	00108
	CALL RDINIT	FORCES	00109
	ITYPE = GHMIXED	FORCES	00110
	MXARRY = GHIBOXW	FORCES	00111
	CALL READMX(IGEOGC,MXREAD,RANDIN,NFS,NMS,LS,NMR,150,NID,ID,ITYPE,	FORCES	00112
1	LRS,IBOXW,M,N,FARM,IRR)	FORCES	00113
	IF(IRR.NE.0) GO TO 6010	FORCES	00114
C		FORCES	00115
	NPLS = 1	FORCES	00116
	IF(NSURF.EQ.1.OR.COPLAN) GO TO 10	FORCES	00117
	NPLS = 2	FORCES	00118
C		FORCES	00119
	CALL RDINIT	FORCES	00120
	ITYPE = GHMIXED	FORCES	00121
	MXARRY = GHIBOXT	FORCES	00122
	CALL READMX(IGEOGC,MXREAD,RANDIN,NFS,NMS,LS,NMR, 90,NID,ID,ITYPE,	FORCES	00123
1	LRS,IBOXT,M,N,FARM,IRR)	FORCES	00124
	IF(IRR.NE.0) GO TO 6010	FORCES	00125
C		FORCES	00126
	10 CONTINUE	FORCES	00127
C		FORCES	00128
C	READ THE TEXLOC AND FEXLOC ARRAYS FROM THE GEOMETRY SCRATCH	FORCES	00129
C	FILE. THESE ARE NEEDED TO INTERPOLATE VELOCITY POTENTIALS AT	FORCES	00130
C	BOX EDGES.	FORCES	00131
C		FORCES	00132
	CALL RDINIT	FORCES	00133
	ITYPE = SHMIXED	FORCES	00134
	MXARRY = GHFEXLOC	FORCES	00135
	CALL READMX(IGEOGC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,	FORCES	00136
1	LRS,FEXLOC,M,N,FARM,IRR)	FORCES	00137
	IF(IRR.NE.0) GO TO 6010	FORCES	00138
C		FORCES	00139
	CALL RDINIT	FORCES	00140
	ITYPE = SHMIXED	FORCES	00141
	MXARRY = GHTEXLOC	FORCES	00142
	CALL READMX(IGEOGC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,	FORCES	00143
1	LRS,TEXLOC,M,N,FARM,IRR)	FORCES	00144
	IF(IRR.NE.0) GO TO 6010	FORCES	00145

C		FORCES	00146
C	READ AREAS AND POINTERS FOR AREAS FROM GEOMETRY SCRATCH FILE.	FORCES	00147
C	THESE ARE USED IN CALCULATION OF BOX LIFTS AND GEN. FORCES.	FORCES	00148
C	CALL RDINIT	FORCES	00149
	ITYPE = SHIMIXED	FORCES	00150
	MXARRY = GHAI.PHAS	FORCES	00151
	CALL READMX(IGEOBC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,	FORCES	00152
	1 LRS,ALPHA,M,N,PARM,IRR)	FORCES	00153
	IF(IRR.NE.0) GO TO 6010	FORCES	00154
C		FORCES	00155
	CALL RDINIT	FORCES	00156
	ITYPE = SHIMIXED	FORCES	00157
	MXARRY = GHJALPH	FORCES	00158
	CALL READMX(IGEOBC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,	FORCES	00159
	1 LRS,IJALPH,M,N,PARM,IRR)	FORCES	00160
	NALPH = N	FORCES	00161
	NALPHW = I.PARM(3)	FORCES	00162
	NALPHY = N - NALPHW	FORCES	00163
	IF(IRR.NE.0) GO TO 6010	FORCES	00164
C		FORCES	00165
C		FORCES	00166
C	REORDER THE FEXLOC AND TEXLOC ARRAYS SO THAT THERE ARE	FORCES	00167
C	VALUES FOR UNSUBDIVIDED CHORDS ONLY.	FORCES	00168
C		FORCES	00169
	IF(NSUBDV.EQ.1) GO TO 120	FORCES	00170
	XSLIDE = NSUBDV-IXBW	FORCES	00171
	JCOL = NSUBCN	FORCES	00172
	NCOLS = MYBW + MYBT	FORCES	00173
	DO 110 I=1,NCOLS	FORCES	00174
	TEXLOC(I) = (TEXLOC(JCOL)+XSLIDE)/XSUBDV	FORCES	00175
	FEXLOC(I) = (FEXLOC(JCOL)+XSLIDE)/XSUBDV	FORCES	00176
	JCOL = JCOL + NSUBDV	FORCES	00177
	110 CONTINUE	FORCES	00178
	120 CONTINUE	FORCES	00179
C		FORCES	00180
C	LOOP ON THICKNESS SLOPE FUNCTIONS (IF NONE WERE REQUESTED,	FORCES	00181
C	ONE DUMMY SET OF ONES WILL HAVE BEEN GENERATED.)	FORCES	00182
C		FORCES	00183
	REWIND ITSLS	FORCES	00184
	DO 750 ITSLOP=1,NTSLOP	FORCES	00185
C		FORCES	00186
C	READ THICKNESS SLOPE FUNCTIONS	FORCES	00187
C		FORCES	00188
	CALL RDINIT	FORCES	00189
	ITYPE = AHREAL	FORCES	00190
	MXARRY = GHITSLFN	FORCES	00191
	CALL READMX(ITSLS,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,	FORCES	00192
	1 LRS,TSLFN,M,N,PARM,IRR)	FORCES	00193
	IF(IRR.NE.0) GO TO 6040	FORCES	00194
C		FORCES	00195
C	ZERO OUT THE AIR FORCES ARRAY	FORCES	00196
C		FORCES	00197
	DO 150 J=1,MAX	FORCES	00198
	150 GENAF(J) = (0.,0.)	FORCES	00199
C	LOOP ON NUMBER OF MODE SHAPES	FORCES	00200
	DO 650 NM=1,NMODES	FORCES	00201
		FORCES	00202

C		FORCES	00203
C	GET MODE SHAPE NM FROM MODESC SCRATCH FILE	FORCES	00204
	CALL RDINIT	FORCES	00205
	IF(NM.EQ.1) NMS = NMSPC	FORCES	00206
	ITYPE = 4HREAL	FORCES	00207
	MXARRY = 6HDEFSL	FORCES	00208
	CALL READMX(MODESC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	FORCES	00209
	1 LRS,DEFSL, M,N,PARM,IRR)	FORCES	00210
	IF(IRR.NE.0) GO TO 6020	FORCES	00211
C		FORCES	00212
C		FORCES	00213
C		FORCES	00214
C	LOOP ON VELOCITY POTENTIALS	FORCES	00215
	REWIND IVPSC	FORCES	00216
C		FORCES	00217
	DO 600 JVP=1,NVPS	FORCES	00218
C		FORCES	00219
C	READ ONE SET OF VELOCITY POTENTIALS	FORCES	00220
	CALL RDINIT	FORCES	00221
	ITYPE = 4HREAL	FORCES	00222
	MXARRY = 6HDELPHI	FORCES	00223
	CALL READMX(IVPSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	FORCES	00224
	1 LRS,DELPHI,M,N,PARM,IRR)	FORCES	00225
	IF(IRR.NE.0) GO TO 6030	FORCES	00226
C		FORCES	00227
	CALL RDINIT	FORCES	00228
	ITYPE = 4HREAL	FORCES	00229
	MXARRY = 3HTVP	FORCES	00230
	CALL READMX(IVPSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,	FORCES	00231
	1 LRS,TVP,M,N,PARM,IRR)	FORCES	00232
C		FORCES	00233
	IF(IRR.NE.0) GO TO 6030	FORCES	00234
C		FORCES	00235
C	CONDENSE THE TRAILING EDGE VELOCITY POTENTIAL ARRAY TO	FORCES	00236
C	UNSUBDIVIDED BOXES.	FORCES	00237
C	ALSO ZERO OUT THE SECTIONAL GENERALIZED FORCES.	BCSFRB	00012
C		BCSFRB	00013
	NTVPS = MYBW + MYBT	BCSFRB	00014
	DO 210 I=1,NTVPS	BCSFRB	00015
	SECMOM(I) = (0.,0.)	BCSFRB	00016
	210 CONTINUE	BCSFRB	00017
C		FORCES	00238
	IF(NSUBDV.EQ.1) GO TO 220	FORCES	00239
	JCOL = NSUBCN	FORCES	00240
	DO 215 I=1,NTVPS	FORCES	00242
	TVP(I) = TVP(JCOL)	FORCES	00243
	JCOL = JCOL + NSUBDV	FORCES	00244
	215 CONTINUE	FORCES	00245
	220 CONTINUE	FORCES	00246
C		FORCES	00247
C	ZERO OUT THE BOX LIFT ARRAY	FORCES	00248
C		FORCES	00249
	IF(NM.NE.1) GO TO 240	FORCES	00250
	IF (.NOT. BLNEED) GO TO 240	FORCES	00251
	NBX = IPNTRM(1,NPNTRS) - 1	FORCES	00252
	DO 230 I=1,NBX	FORCES	00253
	DELCP(I) = (0.,0.)	FORCES	00254

230 DXLIFT(I) = (0.,0.)	FORCES	00255
240 CONTINUE	FORCES	00256
C	FORCES	00257
C	FORCES	00258
C*****	FORCES	00259
C	FORCES	00260
C THE FOLLOWING BLOCK OF CODE COMPUTES	FORCES	00261
C BLIFT - BOX LIFTS	FORCES	00262
C SLIFT - SECTION (CHORD) LIFTS	FORCES	00263
C TLIFT - TOTAL LIFT	FORCES	00264
C GENAF - GENERALIZED AIRFORCES	FORCES	00265
C	FORCES	00266
C*****	FORCES	00267
C	FORCES	00268
C AFROM(JVP) = (0.,0.)	FORCES	00269
C AFROM = CURRENT ROW OF GENERALIZED AIRFORCES	FORCES	00270
C	FORCES	00271
C SET UP INITIAL CONDITIONS FOR DOUBLE LOOP OVER THE ENTIRE	FORCES	00272
C BOX PATTERN	FORCES	00273
C IBXCDF = FORWARD	FORCES	00274
C IBXCD = CENTER ROWS OF BOX CODES, EXPANDED	FORCES	00275
C IBXCDA = AFT /	FORCES	00276
C VPLE = VELOCITY POTENTIAL AT BOX LEADING EDGE	FORCES	00277
C VPTE = ARRAY OF BOX TRAILING EDGE VELOCITY POTENTIALS	FORCES	00278
C	FORCES	00279
DO 565 NP=1,NPLS	FORCES	00280
IF (NP.EQ.2) GO TO 245	FORCES	00281
ISROW = 1	FORCES	00282
NBX = IPNTRM(1,2)	FORCES	00283
CALL DCDER(IBOXW,150,ISROW,1,ISROW,NBX,.F.,IBXCDA)	FORCES	00284
NBX = NBX	FORCES	00285
GO TO 250	FORCES	00286
245 CONTINUE	FORCES	00287
ISROW = (IXBT-IXBW)/NSUBDV + 1	FORCES	00288
IXBT = ISROW	FORCES	00289
IDEX = ISROW + IOVLAP	FORCES	00290
NBX = IPNTRM(1,INDEX+1) - IPNTRM(1,INDEX)	FORCES	00291
ISUBT = 2-IXBT	FORCES	00292
CALL DCDER(IBOXT(ISUBT,1),LBXCDT,ISROW,1,ISROW,NBX,.F.,IBXCDA)	FORCES	00293
NBX = NBX	FORCES	00294
250 CONTINUE	FORCES	00295
DO 270 JCQL = 1,NBX	FORCES	00296
IBXCD(JCQL) = IBXCDA(JCQL)	FORCES	00297
IF (IBXCDA(JCQL).EQ.1) GO TO 260	FORCES	00298
VPTE(JCQL) = XINDEF	FORCES	00299
DEFLTE(JCQL) = XINDEF	FORCES	00300
GO TO 270	FORCES	00301
260 CONTINUE	FORCES	00302
VPTE(JCQL) = (0.,0.)	FORCES	00303
IDC = JCQL	FORCES	00304
DEFLTE(JCQL) = DEFSL(1,IDC) + DEFSL(2,IDC)*81*(PEXLOC(JCQL)-1.0)	FORCES	00305
270 CONTINUE	FORCES	00306
C	FORCES	00307
C	FORCES	00308
C LOOP ON ROWS OF THE BOX PATTERN	FORCES	00309
IF(NP.EQ.2) GO TO 275	FORCES	00310
IRS = 1	FORCES	00311

NXBT = NXBW	FORCES 00312
IF(COPLAN) NXBT = NXBT	FORCES 00313
NROWS = NXBT	FORCES 00314
GO TO 280	FORCES 00315
275 CONTINUE	FORCES 00316
IRS = (IXBT-IXBW)/NSUBDV + 1 + IOVLAP	FORCES 00317
NXBT = NXBT + IOVLAP	FORCES 00318
280 CONTINUE	FORCES 00319
DO 560 IROW=IRS,NXBT	FORCES 00320
C	FORCES 00321
C TRANSFER BOX CODES TO CORRECT ARRAYS FOR NEW ROW	FORCES 00322
NBXF = NBX	FORCES 00323
NBX = NBXA	FORCES 00324
IF (NBXF .EQ. 0) GO TO 315	FORCES 00325
DO 310 JCQL = 1,NBXF	FORCES 00326
310 IBXCD(JCQL) = IBXCD(JCQL)	FORCES 00327
315 CONTINUE	FORCES 00328
IF (NBX .EQ. 0) GO TO 325	FORCES 00329
DO 320 JCQL = 1,NBX	FORCES 00330
320 IBXCD(JCQL) = IBXCDA(JCQL)	FORCES 00331
325 CONTINUE	FORCES 00332
IF (IROW .EQ. NXBT) GO TO 335	FORCES 00333
ISROWA = ISROWA + 1	FORCES 00334
IF(NP.EQ.2) GO TO 330	FORCES 00335
NBXA = IPNTRM(1,ISROWA+1) - IPNTRM(1,ISROWA) + IPNTRM(2,ISROWA) - 1	FORCES 00336
CALL DCDER(IBOXW,150,ISROWA,1,ISROWA,NBXA,.F.,IBXCDA)	FORCES 00337
GO TO 340	FORCES 00338
330 CONTINUE	FORCES 00339
IDEX = ISROWA + IOVLAP	FORCES 00340
NBXA = IPNTRM(1,IDEX+1) - IPNTRM(1,IDEX) + IPNTRM(2,IDEX) - 1	FORCES 00341
ISUBT = 2-IXBST	FORCES 00342
CALL DCDER(IBOXW,ISUBT,1,IBXCDA,ISROWA,1,ISROWA,NBXA,.F.,IBXCDA)	FORCES 00343
GO TO 340	FORCES 00344
335 CONTINUE	FORCES 00345
NBXA = 0	FORCES 00346
340 CONTINUE	FORCES 00347
C	FORCES 00348
ITROW = IROW	FORCES 00349
IF(NP.EQ.2) ITROW = IROW - IOVLAP	FORCES 00350
C	FORCES 00351
C LOOP ON CHORDS OF THE BOX PATTERN	FORCES 00352
IF (NBX .EQ. 0) GO TO 560	FORCES 00353
DO 550 JCQL = 1,NBX	FORCES 00354
C	FORCES 00355
IF (IBXCD(JCQL) .NE. 1) GO TO 550	FORCES 00356
IDC = LOCSDW(IROW,JCQL, IPNTRM,LPNTRM,1,LPNTRM)	FORCES 00357
IF (IDC .EQ. 0) GO TO 970	FORCES 00358
C	FORCES 00359
C GET THE SUBSCRIPT TO USE IN THE EDGE ARRAYS, JJ	FORCES 00360
JJ = JCQL	FORCES 00361
IF (NP .EQ. 2) GO TO 350	FORCES 00362
IF (.NOT. COPLAN) GO TO 355	FORCES 00363
IF (JJ .GT. NXBT) GO TO 355	FORCES 00364
IF (YEXLOC(JJ) .GE. FLOAT(ITROW)) GO TO 355	FORCES 00365
350 JJ = JJ + MYBW	FORCES 00366
355 CONTINUE	FORCES 00367
C	FORCES 00368

C	DETERMINE BOX LEADING EDGE VALUES	FORCES	00369
	IF (FEXLOC(JJ) .GT. FLOAT(ITROW*1)) GO TO 410	FORCES	00370
C	BOX LEADING EDGE IS INTERNAL TO THE PLANFORM	FORCES	00371
	VPLE = VPTE(JCOL)	FORCES	00372
	DEFLLE = DEFLTE(JCOL)	FORCES	00373
	GO TO 440	FORCES	00374
C	BOX IS ON PLANFORM LEADING EDGE. IS IT INFLUENCED BY THE WAKE-	FORCES	00375
410	CONTINUE	FORCES	00376
	IF (JJ .EQ. JCOL .OR. .NOT. COPLAN) GO TO 420	FORCES	00377
		FORCES	00378
C	LEADING EDGE OF SECOND PLANFORM. VELOCITY POTENTIAL	FORCES	00379
C	COMPUTED FROM WAKE EQUATION.	FORCES	00380
C	XDKVL = (FEXLOC(JJ) - TEXLOC(JCOL)) * XKVL	FORCES	00381
	VPLE = TVP(JCOL) * CMPLX(COS(XDKVL),-SIN(XDKVL))	FORCES	00382
	GO TO 425	FORCES	00383
		FORCES	00384
C	LEADING EDGE OF SECOND PLANFORM OF SPATIAL ANALYSIS	FORCES	00385
C	OR LEADING EDGE OF FIRST PLANFORM (WING)	FORCES	00386
420	CONTINUE	FORCES	00387
	VPLE = (0.,0.)	FORCES	00388
425	CONTINUE	FORCES	00389
C	TEST FOR SINGLE BOX	FORCES	00390
	IF (TEXLOC(JJ) .LT. FLOAT(ITROW*1)) GO TO 430	FORCES	00391
C	BOX IS A SIMPLE LEADING EDGE BOX	FORCES	00392
	IDA = LOCSDW(ITROW*1,JCOL, IPNTRM,LPNTRM,1,LPNTRM)	FORCES	00393
	IF (IDA .EQ. 0) GO TO 970	FORCES	00394
	SLOPE = B1*DEFSL(2,IDA)	FORCES	00395
	XDIF = FLOAT(ITROW) - FEXLOC(JJ)	FORCES	00396
	DEFLLE = DEFSL(1,IDA) - SLOPE * XDIF	FORCES	00397
	GO TO 450	FORCES	00398
		FORCES	00399
C	SINGLE BOX. GET LEADING AND TRAILING VALUES	FORCES	00400
430	CONTINUE	FORCES	00401
	SLOPE = B1*DEFSL(2,IDA)	FORCES	00402
	DEFLLE = DEFSL(1,IDA) - SLOPE * (FLOAT(ITROW) - FEXLOC(JJ))	FORCES	00403
	VPTE(JCOL) = TVP(JJ)	FORCES	00404
	DEFLTE(JCOL) = DEFSL(1,IDA) + SLOPE * (TEXLOC(JJ) - FLOAT(ITROW))	FORCES	00405
	GO TO 500	FORCES	00406
		FORCES	00407
C	DETERMINE BOX TRAILING EDGE VALUES	FORCES	00408
440	CONTINUE	FORCES	00409
	IF (TEXLOC(JJ) .LT. FLOAT(ITROW*1)) GO TO 460	FORCES	00410
		FORCES	00411
C	BOX TRAILING EDGE IS INTERNAL TO THE PLANFORM	FORCES	00412
C	IDA = LOCSDW(ITROW*1,JCOL, IPNTRM,LPNTRM,1,LPNTRM)	FORCES	00413
450	CONTINUE	FORCES	00414
	VPTE(JCOL) = .5 * (DELPHI(IDC) + DELPHI(IDA))	FORCES	00415
	DEFLTE(JCOL) = 0.5*(DEFSL(1,IDA) + DEFSL(1,IDA))	FORCES	00416
	GO TO 500	FORCES	00417
		FORCES	00418
C	BOX IS ON SURFACE TRAILING EDGE	FORCES	00419
460	CONTINUE	FORCES	00420
	VPTE(JCOL) = TVP(JJ)	FORCES	00421
	SLOPE = B1*DEFSL(2,IDA)	FORCES	00422
	DEFLTE(JCOL) = DEFSL(1,IDA) + SLOPE * (TEXLOC(JJ) - FLOAT(ITROW))	FORCES	00423
		FORCES	00424
C	BOX LEADING AND TRAILING EDGE VALUES ARE COMPUTED. GET	FORCES	00425
C			

C	ALPHA, THE AREA MULTIPLIER	FORCES	00426
500	CONTINUE	FORCES	00427
	IF(NP.EQ.2) GO TO 505	FORCES	00428
	NAS = 1	FORCES	00429
	NAL = NALPHW	FORCES	00430
	GO TO 506	FORCES	00431
C		FORCES	00432
505	CONTINUE	FORCES	00433
	NAS = NALPHW + 1	FORCES	00434
	NAL = NALPH	FORCES	00435
506	CONTINUE	FORCES	00436
C		FORCES	00437
	ALPH = 1.0	FORCES	00438
	JCOMP = JCQL*512	FORCES	00439
	JCOMP1 = JCOMP+512	FORCES	00440
	DO 510 I=NAS,NAL	FORCES	00441
	IF (IJALPH(I) .LT. JCOMP) GO TO 510	FORCES	00442
	IF (IJALPH(I) .GT. JCOMP1) GO TO 520	FORCES	00443
	IF (IJALPH(I) .NE. JCOMP+ITROW) GO TO 510	FORCES	00444
	ALPH = ALPHA(I)	FORCES	00445
	GO TO 520	FORCES	00446
510	CONTINUE	FORCES	00447
520	CONTINUE	FORCES	00448
	IF(PLYWOOD) ALPH = 1.0	FORCES	00449
C		FORCES	00450
C	COMPUTE TEMP1 = K1*ALPHA*(I)*(DELTA PHI)	FORCES	00451
	TEMP1 = CMPLX(-AIMAG(DELPHI(IDC)), REAL(DELPHI(IDC)))	FORCES	00452
	TEMP1 = TEMP1 * (XXVL*ALPH)	FORCES	00453
C		FORCES	00454
	IF(NM.NE.1) GO TO 530	FORCES	00455
C	ARE BOX LIFTS DESIRED -	FORCES	00456
	IF (.NOT. BLNEED) GO TO 530	FORCES	00457
	BXLIFT(IDC) = (TEMP1 + ALPH*(VFTE(JCQL)-VPLE))*TWOBET* TSLFN(IDC)	FORCES	00458
	DELCP(IDC) = BXLIFT(IDC)/(ALPH*B1)	FORCES	00459
530	CONTINUE	FORCES	00460
C		FORCES	00461
	TEMP2 = DEFLTE(JCQL)*VFTE(JCQL) - DEFLTE* VPLE	FORCES	00462
	TEMP3 = (B1*ALPH*DEFSL(2,IDC)) * DELCP(IDC)	FORCES	00463
	GAF = (TEMP1*DEFSL(1,IDC) + TEMP2 - TEMP3) * TSLFN(IDC)	BCSFRB	00018
	AFROW(JVP) = AFROW(JVP) + GAF	BCSFRB	00019
C		BCSFRB	00020
	SECMOM(JJ) = SECMOM(JJ) + GAF	BCSFRB	00021
C		BCSFRB	00022
C		FORCES	00466
550	CONTINUE	FORCES	00467
C	END OF LOOP ON CHORDS OF THE BOX PATTERN, FROM 340*	FORCES	00468
560	CONTINUE	FORCES	00469
C	END OF LOOP ON ROWS OF THE BOX PATTERN, FROM 270*	FORCES	00470
C		FORCES	00471
565	CONTINUE	FORCES	00472
C	END OF LOOP ON NUMBER OF PLATFORMS	FORCES	00473
C		FORCES	00474
C	SET UP TO WRITE RESULTS ON TAPE	FORCES	00475
	CALL RDINIT	FORCES	00476
	ITYPE = 7HCOMPLEX	FORCES	00477
	PARM(1) = XXVL	FORCES	00478
	PARM(2) = B1	FORCES	00479

PARM(3) = XMACH	FORCES 00480
C	FORCES 00481
C IF THIS IS THE FIRST WEIGHTING FUNCTION, IT MAY BE NECESSARY	FORCES 00482
C TO WRITE AND/OR PRINT BOX LIFTS, ETC	FORCES 00483
IF (.NOT. BLNEED) GO TO 600	FORCES 00484
IF (NM.NE.1) GO TO 600	FORCES 00485
C	FORCES 00486
IF (.NOT. PRBL) GO TO 570	FORCES 00487
C PRINT BOX LIFTS	FORCES 00488
TITL(1) = 8H WING	FORCES 00489
TITL(2) = 10HBOX LIFTS	FORCES 00490
TITL(3) = 2H	FORCES 00491
IF(COPLAN) TITL(1) = 10HWING/TAIL	FORCES 00492
CALL PRNTBL(TITL,JVP,BXLIFT,1,NROWS,MYBW,IPNTRM)	FORCES 00493
IF(NSURF.EQ.1 .OR. COPLAN) GO TO 570	FORCES 00494
TITL(1) = 8H TAIL	FORCES 00495
CALL PRNTBL(TITL,JVP,BXLIFT,IXBUT,MXBT,MYBT,IPNTRM(1,IOVLAP+1))	FORCES 00496
570 CONTINUE	FORCES 00497
C	FORCES 00498
IF (.NOT. PRDCP) GO TO 572	FORCES 00499
C	FORCES 00500
C PRINT PRESSURE DIFFERENTIAL	FORCES 00501
TITL(1) = 8H WING	FORCES 00502
TITL(2) = 10HPRESS. DI	FORCES 00503
TITL(3) = 10HFFERENCE	FORCES 00504
IF(COPLAN) TITL(1) = 10HWING/TAIL	FORCES 00505
CALL PRNTBL(TITL,JVP,DELCP,1,NROWS,MYBW,IPNTRM)	FORCES 00506
IF (NSURF.EQ.1 .OR. COPLAN) GO TO 572	FORCES 00507
TITL(1) = 8H TAIL	FORCES 00508
CALL PRNTBL(TITL,JVP,DELCP,IXBUT,MXBT,MYBT,IPNTRM(1,IOVLAP+1))	FORCES 00509
572 CONTINUE	FORCES 00510
C	FORCES 00511
IF(.NOT. (WTBL.OR. PRSL)) GO TO 600	FORCES 00512
C	FORCES 00513
~~C *****~~	FORCES 00514
C EXPAND BOX LIFTS FOR WRITING ON TAPE. WRTEP FORMAT ONLY	FORCES 00515
IF (MXRIT) WRITE (NT6,9999)	FORCES 00516
C INITIALIZE COUNTERS FOR PASSING OVER ARRAY BACKWARDS	FORCES 00517
C IJKL = CURRENT LOCATION IN INPUT (COMPRESSED) ARRAY	FORCES 00518
C IJ = CURRENT LOCATION IN OUTPUT (EXPANDED) ARRAY	FORCES 00519
C IJFST = FIRST LOCATION FOR CURRENT ROW IN INPUT ARRAY	FORCES 00520
C IJPRV = FIRST LOCATION FOR PREVIOUS ROW IN INPUT ARRAY	FORCES 00521
C	FORCES 00522
IROW = MXB	FORCES 00523
IJFST = IPNTRM(1,MXB)	FORCES 00524
LOCFT = IPNTRM(2,MXB)	FORCES 00525
IJPRV = IPNTRM(1,MXB+1)	FORCES 00526
IJKL = IJPRV	FORCES 00527
IJ = NBOXES - MYB + IJPRV - IJFST + LOCFT	FORCES 00528
DO 575 I = 1,NBOXES	FORCES 00529
MBF(I) = (0.,0.)	FORCES 00530
575 CONTINUE	FORCES 00531
C	FORCES 00532
C LOOP BACK HERE ON ROWS, AND ON CHORDS WITHIN A ROW	FORCES 00533
580 CONTINUE	FORCES 00534
IJ = IJ - 1	FORCES 00535
IJKL = IJKL - 1	FORCES 00536


```

      BL = BXLIFT(IJKL)
      BXLIFT(IJKL) = (0.,0.)
      RWBF(IJ)= BL
      IF (IJKL .GT. IJFST) GO TO 580
C      END OF LOOP ON CHORDS WITHIN ONE ROW. STEP TO NEXT ROW
      IJPRV = IJFST
      LOCPV = LOCFST
C      LOOP BACK HERE ON EMPTY ROWS (COPLANAR CASE)
582 CONTINUE
      IROW = IROW - 1
      IJ = IJ - MYB
C      DETERMINE WHETHER DONE -
      IF (IROW .EQ. 0) GO TO 584
C      IS THE ROW EMPTY -
      IF (IPNTRM(1,IROW) .EQ. IJPRV) GO TO 582
      IJFST = IPNTRM(1,IROW)
      LOCFST = IPNTRM(2,IROW)
      IJ = IJ - LOCPV + IJPRV-IJFST + LOCFST
      GO TO 580
C      END OF LOOP ON ROWS.
C*****
C
584 CONTINUE
C      ARE THE SECTION LIFTS TO BE PRINTED OR WRITTEN -
      IF(.NOT. PRSL) GO TO 595
C      COMPUTE SECTION LIFTS
      TLIFT = (0.,0.)
      TLIFT1 = (0.,0.)
      TLIFT2 = (0.,0.)
      DO 590 JCQL = 1,MYB
      BL = (0.,0.)
      BL2 = (0.,0.)
      IROW = 0
      DO 587 IJ = JCQL,NBOXES,MYB
      IROW = IROW + 1
      IF(IROW.GT.TEXLOC(JCQL)) GO TO 586
      BL = BL + RWBF(IJ)
      GO TO 587
586 BL2 = BL2 + RWBF(IJ)
587 CONTINUE
      TLIFT1 = TLIFT1 + BL
      TLIFT2 = TLIFT2 + BL2
      SLIFT(JCQL+MYBW) = BL2
      SLIFT(JCQL) = BL
590 CONTINUE
C
      IF (.NOT. PRSL ) GO TO 595
C      PRINT SECTION LIFTS AND TOTAL LIFT
      CALL PRNTSL(JVP,SLIFT,TLIFT1,TLIFT2,MYBW,MYBT )
C
595 CONTINUE
      IF(.NOT.PRCM) GO TO 599
      DO 596 I=1,NTVPS
      SECMOM(I) = SECMOM(I) + TWBET
596 CONTINUE
      NCM = -JVP
      CALL PRNTSL(NCM,SECMOM,TLIFT1,TLIFT2,MYBW,MYBT)

```

```

FORCES 00537
FORCES 00538
FORCES 00539
FORCES 00540
FORCES 00541
FORCES 00542
FORCES 00543
FORCES 00544
FORCES 00545
FORCES 00546
FORCES 00547
FORCES 00548
FORCES 00549
FORCES 00550
FORCES 00551
FORCES 00552
FORCES 00553
FORCES 00554
FORCES 00555
FORCES 00556
FORCES 00557
FORCES 00558
FORCES 00559
FORCES 00560
FORCES 00561
FORCES 00562
FORCES 00563
FORCES 00564
FORCES 00565
FORCES 00566
FORCES 00567
FORCES 00568
FORCES 00569
FORCES 00570
FORCES 00571
FORCES 00572
FORCES 00573
FORCES 00574
FORCES 00575
FORCES 00576
FORCES 00577
FORCES 00578
FORCES 00579
FORCES 00580
FORCES 00581
FORCES 00582
FORCES 00583
FORCES 00584
FORCES 00585
FORCES 00586
FORCES 00587
BCSFRB 00023
BCSFRB 00024
BCSFRB 00025
BCSFRB 00026
BCSFRB 00027
BCSFRB 00028

```

C	599 CONTINUE	BC3FRB 00029
	IF (.NOT. WTEL) GO TO 600	BC3FRB 00030
C	WRITE BOX LIFTS ONTO THE BINARY OUTPUT FILE	FORCES 00588
	ITYPE = 7HCOMPLEX	FORCES 00589
	M = MXB	FORCES 00590
	K = - M	FORCES 00591
	N = MYB	FORCES 00592
	MXARRY = 7HBXLFIS	FORCES 00593
	ID(2) = 1000000*IKVAL + JVP	FORCES 00594
	CALL WRTEMX(NOUTP, MXWRT, RANDOU, NFS, NMS, LS, NMR, LWS, K, ID,	FORCES 00595
	1 RWBF, ITYPE, M, N, FARM, IRR)	FORCES 00596
	IF (IRR .NE. 0) GO TO 922	FORCES 00597
C	600 CONTINUE	FORCES 00598
	END OF LOOP ON VELOCITY POTENTIALS	FORCES 00599
C		FORCES 00600
C	STORE THE ROW OF GENERALIZED AIRFORCES INTO THE FULL MATRIX	FORCES 00601
	IJ = NM	FORCES 00602
	DO 620 JVP = 1, NVPS	FORCES 00603
	GENAF(IJ) = AFRON(JVP)*TWOBET	FORCES 00604
	GPAFC(IJ) = BS3BET * REAL(GENAF(IJ))	FORCES 00605
	GPPAFC(IJ) = 0.	FORCES 00606
	IF(XVL.NE.0.) GPPAFC(IJ) = BKS4BT * AIMAG(GENAF(IJ))	FORCES 00607
	IJ = IJ + NMODES	FORCES 00608
	620 CONTINUE	FORCES 00609
	630 CONTINUE	FORCES 00610
C		FORCES 00611
C	IF(.NOT.WTGNAF) GO TO 670	FORCES 00612
	K = NMODES	FORCES 00613
	M = NMODES	FORCES 00614
	N = NMODES	FORCES 00615
	ID(2) = IKVAL	FORCES 00616
	CALL WRTEMX(NOUTP, MXWRT, RANDOU, NFS, NMS, LS, NMR, LWS, K, ID,	FORCES 00617
	1 GENAF, ITYPE, M, N, FARM, IRR)	FORCES 00618
	IF (IRR .NE. 0) GO TO 928	FORCES 00619
C		FORCES 00620
C	ARE THE FORCES TO BE PRINTED -	FORCES 00621
	670 CONTINUE	FORCES 00622
	IF (.NOT. PRGNMF) GO TO 700	FORCES 00623
C		FORCES 00624
	CALL PRNTAF(GENAF, PRGNMF, GPAFC, GPPAFC)	FORCES 00625
	700 CONTINUE	FORCES 00626
C		FORCES 00627
	REWIND MODESC	FORCES 00628
	NMSPC = 1	FORCES 00629
	730 CONTINUE	FORCES 00630
C		FORCES 00631
	RETURN	FORCES 00632
C		FORCES 00633
C	DIAGNOSTICS - ALL CALL FLUSH	FORCES 00634
C		FORCES 00635
C	READING FROM SCRATCH FILE	FORCES 00636
	8010 CONTINUE	FORCES 00637
	WRITE (NT6,9100) IGEOSC	FORCES 00638
	GO TO 930	FORCES 00639
		FORCES 00640
		FORCES 00641
		FORCES 00642

```

6020 CONTINUE
      WRITE (NT6,9120) MODESC
      GO TO 950
6030 CONTINUE
      WRITE (NT6,9180) IVPSC
      GO TO 950
C
6040 CONTINUE
      WRITE (NT6,9140) ITSLS
      GO TO 950
C
C      WRITING ON THE OUTPUT TAPE
922 CONTINUE
      WRITE (NT6,9220) NOUTP
      GO TO 952
928 CONTINUE
      WRITE (NT6,9280) NOUTP
      GO TO 952
C      INCORRECT DIMENSIONS READ
930 CONTINUE
      I = 1
      GO TO 932
931 I = 2
932 WRITE (NT5,9300) I
      IF(MXREAD) GO TO 960
      GO TO 962
C      ERROR DETECTED READING A MATRIX
950 CONTINUE
      WRITE (NT6,9500) IRR
      IF(MXREAD) GO TO 960
      GO TO 962
C      ERROR DETECTED WRITING A MATRIX
952 CONTINUE
      WRITE (NT6,9520) IRR
      IF(MXWRITE) GO TO 960
      GO TO 962
C      MATRIX DESCRIPTION
960 CONTINUE
      WRITE (NT6,9600) (ID(I),I=1,10),(ID(I),I=1,10)
      WRITE (NT6,9622) FARM,FARM
      WRITE (NT6,9614) NMR,NMR,LRS,LRS
      GO TO 964
962 WRITE (NT6,9620) ID(1),ID(2)
      WRITE (NT6,9622) FARM,FARM
      WRITE (NT6,9624) NFS,NFS
964 WRITE (NT6,9640) ITYPE,M,N
      WRITE (NT6,9650) MXARRY
      GO TO 990
970 CONTINUE
      WRITE (NT6,9700) IROW, JCCL
      GO TO 990
C
990 CC INUE
      WRITE (NT6,9900)
C
      CALL FLUSH(1)
C

```

```

FORCES 00643
FORCES 00644
FORCES 00645
FORCES 00646
FORCES 00647
FORCES 00648
FORCES 00649
FORCES 00650
FORCES 00651
FORCES 00652
FORCES 00653
FORCES 00654
FORCES 00655
FORCES 00656
FORCES 00657
FORCES 00658
FORCES 00659
FORCES 00660
FORCES 00661
FORCES 00662
FORCES 00663
FORCES 00664
FORCES 00665
FORCES 00666
FORCES 00667
FORCES 00668
FORCES 00669
FORCES 00670
FORCES 00671
FORCES 00672
FORCES 00673
FORCES 00674
FORCES 00675
FORCES 00676
FORCES 00677
FORCES 00678
FORCES 00679
FORCES 00680
FORCES 00681
FORCES 00682
FORCES 00683
FORCES 00684
FORCES 00685
FORCES 00686
FORCES 00687
FORCES 00688
FORCES 00689
FORCES 00690
FORCES 00691
FORCES 00692
FORCES 00693
FORCES 00694
FORCES 00695
FORCES 00696
FORCES 00697
FORCES 00698
FORCES 00699

```

C	DIAGNOSTIC FORMATS	FORCES	00700
9100	FORMAT(47H0*** ERROR WHILE READING GEOMETRY SCRATCH FILE ,A10,	FORCES	00701
1	4H ***)	FORCES	00702
9120	FORMAT(44H0*** ERROR WHILE READING MODES SCRATCH FILE ,A10,	FORCES	00703
1	4H ***)	FORCES	00704
9140	FORMAT(54H0*** ERROR WHILE READING THICKNESS SLOPE SCRATCH FILE	FORCES	00705
1	A10, 4H ***)	FORCES	00706
9180	FORMAT(51H0*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH	FORCES	00707
1	6H FILE ,A10,4H ***)	FORCES	00708
	DATA XINCF /2* 60000000000200377777B /	FTNX1	00082
9220	FORMAT(49H0*** ERROR WHILE WRITING BOX LIFTS ON OUTPUT TAPE,I2,	FORCES	00711
1	4H ***)	FORCES	00712
9280	FORMAT(56H0*** ERROR WHILE WRITING GENERALIZED AIRFORCES ON OUTPUT	FORCES	00713
1	5H TAPE,I2,4H ***)	FORCES	00714
9300	FORMAT(11H0, 48H*** MATRIX READ ERROR. THE M DIMENSION SHOULD	FORCES	00715
1	4H BE ,I2, 4H ***)	FORCES	00716
9300	FORMAT(16H0 *** ERROR CODE ,I5, 28H WHILE READING THE FOLLOWING	FORCES	00717
1	11H MATRIX ***)	FORCES	00718
9320	FORMAT(16H0 *** ERROR CODE ,I5, 28H WHILE WRITING THE FOLLOWING	FORCES	00719
1	11H MATRIX ***)	FORCES	00720
9600	FORMAT(5X,MATRIX ID = *, 10A10 / (20X,10A10))	FTNX1	00083
9614	FORMAT(5X,22HMATRIX INDEX (NAME) = ,I5,2H (A10,1H) /	FORCES	00722
1	5X,33HLEVEL NUMBER READ (OR WRITTEN) = 02,3H, (,02,1H))	FTNX1	00084
9620	FORMAT(5X,MATRIX ID = *, A10, I10)	FORCES	00724
9622	FORMAT(5X,11HPARAMETERS, 10E11.3 /10X, 9H(INTEGER), I7,9I11)	FORCES	00725
9624	FORMAT(5X,15HFILE SPACING = ,I3, 19H, MATRIX SPACING = ,I3)	FORCES	00726
9640	FORMAT(5X,MATRIX TYPE - *,I10, *, DIMENSIONED (*I4,* X*,I4,*)*)	FORCES	00727
9650	FORMAT(5X,*ARRAY - *, A10)	FORCES	00728
9700	FORMAT(37H0*** POINTER ARRAY EXCEEDED FOR BOX (,I4,1H,I4,5H) ***)	FORCES	00729
9900	FORMAT(48H0*** ERROR OCCURRED DURING GENERALIZED AIRFORCES	FORCES	00730
1	17H CALCULATIONS ***)	FORCES	00731
9999	FORMAT(54H0*** W A R N I N G - BOX LIFTS CANNOT BE WRITTEN IN	FORCES	00732
1	10HMARK FORMAT ***)	FORCES	00733
C		FORCES	00734
	END	FORCES	00735

SUBROUTINE DCODER(ibox,lbox, ia,ja, il,jl, subd, icode)	DCODAF	00002
dimension ibox(lbox,1), icode(1)	DCODAF	00003
C	DCODAF	00004
C ibox - array of box codes in packed word format	DCODAF	00005
C lbox - row dimension of box codes array	DCODAF	00006
C ia - i-th index of first code to retrieve	DCODAF	00007
C ja - j-th index of first code to retrieve	DCODAF	00008
C il - last box code on the ja-th chord to retrieve	DCODAF	00009
C jl - last box on the ia-th row to retrieve	DCODAF	00010
C subd - .T., subdivided box codes desired, .F. unsubdivided.	DCODAF	00011
C icode - array into which box code will be stored.	DCODAF	00012
C	DCODAF	00013
C comment on usage	DCODAF	00014
C box codes can be retrieved for one box, a row or part of	DCODAF	00015
C a row, or a column or part of a column. a row and column can	DCODAF	00016
C not be retrieved at the same time. if only 1 box is desired	DCODAF	00017
C set il = ia and jl = ja. if both il .ne. ia and jl .ne.	DCODAF	00018
C ja, one row will be returned, il being ignored.	DCODAF	00019
C	DCODAF	00020
common /geomy/ coplan,nsubdv,xsubdv,nsubd2,nsubcn,nsurf,	GEOMTY	00002
1 b1,b1beta,b1s,b1btas,wlax,wlaz,psiw,	GEOMTY	00003
2 mxbw,mxbbw,mybw,mybbw,mxbsw,mybsw,mybbsw,	GEOMTY	00004
3 ixbw,xcenr	GEOMTY	00005
logical coplan	GEOMTY	00006
logical subd	DCODAF	00022
integer shift	DCODAF	00023
data nword /20/	DCODAF	00024
mask = 7	DCODAF	00025
ib = 1	DCODAF	00026
if (subd) go to 50	DCODAF	00027
i = nsubdv * (ia-1) + ixbw	DCODAF	00028
j = nsubdv * (ja-1) + nsubcn	DCODAF	00029
iskip = nsubdv	DCODAF	00030
iend = nsubdv * (il-1) + ixbw	DCODAF	00031
jend = nsubdv * (jl-1) + nsubcn	DCODAF	00032
go to 60	DCODAF	00033
50 continue	DCODAF	00034
i = ia	DCODAF	00035
j = ja	DCODAF	00036
iskip = 1	DCODAF	00037
iend = il	DCODAF	00038
jend = jl	DCODAF	00039
60 continue	DCODAF	00040
if (jl .eq. ja) go to 1100	DCODAF	00041
C	DCODAF	00042
C program will retrieve ni boxes from row i	DCODAF	00043
100 continue	DCODAF	00044
do 1000 jj = j,jend,iskip	DCODAF	00045
jsb = (jj-1)/nword + 1	DCODAF	00046
ijword = ibox(i,jsb)	DCODAF	00047
jb = (nword - mod(jj,nword)) * 3	DCODAF	00048
if(jb.eq.60) jb = 0	DCODAF	00049
C jb = number of bits to shift left.	DCODAF	00050
ijmask = shift(mask,jb)	DCODAF	00051
ijcode = ijword.and.ijmask	DCODAF	00052
nub = -jb	DCODAF	00053
icode(ib) = shift(ijcode,nub)	DCODAF	00054

```

      IB = IB + 1
1000 CONTINUE
      GO TO 3000
C
      PROGRAM WILL RETRIEVE NJ BOXES FROM CHORD J
C
1100 CONTINUE
      JSB = (J-1)/NEWRD + 1
      JB = (NEWRD - MOD(J,NEWRD)) * 3
      IF(JB.EQ.60) JB = 0
      IJMASK = SHIFT(MASK,JB)
      NJB = -JB
      DO 2000 II = 1,IEND,ISKIP
      IJWORD = IBOX(II,JSB)
      IJCODE = IJWORD.AND.IJMASK
      ICODE(IB) = SHIFT(IJCODE,NJB)
      IB = IB + 1
2000 CONTINUE
C
3000 CONTINUE
      RETURN
      END

```

```

DCODAF 00055
DCODAF 00056
DCODAF 00057
DCODAF 00058
DCODAF 00059
DCODAF 00060
DCODAF 00061
DCODAF 00062
DCODAF 00063
DCODAF 00064
DCODAF 00065
DCODAF 00066
DCODAF 00067
DCODAF 00068
DCODAF 00069
DCODAF 00070
DCODAF 00071
DCODAF 00072
DCODAF 00073
DCODAF 00074
DCODAF 00075

```

SUBROUTINE PRNTBL(TITL,IMODE,ARRAY,IXB,MYB,IPNTRM)	PRNTBL 00002
C PRINTS BOX LIFTS, USES /RWBUFF/ FOR INTERMEDIATE SCRATCH	PRNTBL 00003
C IMODE - MODE SHAPE NUMBER	PRNTBL 00004
C ARRAY - ARRAY TO BE PRINTED	PRNTBL 00005
C IPNTRM - POINTER ARRAY FOR ROWS IN -ARRAY-	PRNTBL 00006
C NPNTRS - NUMBER OF POINTERS	PRNTBL 00007
C	PRNTBL 00008
COMPLEX ARRAY(1)	PRNTBL 00009
DIMENSION TITL(3)	PRNTBL 00010
DIMENSION IPNTRM(2,50)	PRNTBL 00011
C	PRNTBL 00012
COMMON /CONTRL/ PRVEX,OMACH, TITLE(8), PRVGEOM,PRVMODE,DIHW,DIHT,	PRNTBL 00013
1 DEFAULT	CONTRL 00002
LOGICAL PRVGEOM,PRVMODE,DIHW,DIHT,DEFAULT	CONTRL 00003
COMMON /PROBLM/ XMACH,NMODES,NTSLOF,NKVALS,SMOOTH,NDEG,CRDFIT,	CONTRL 00004
1 EXAIC,SUBDV,PLYWOOD	PROBLM 00002
LOGICAL SMOOTH,CRDFIT,EXAIC,SUBDV,PLYWOOD	PROBLM 00003
COMMON /KVAL / IKVAL,XXVAL(20), XKS(20)	PROBLM 00004
COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUTP,	KVAL 00002
1 IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC	FILES 00002
C	FILES 00003
INTEGER PAGE	PRNTBL 00019
DIMENSION S(1),D(1)	PRNTBL 00020
EQUIVALENCE (S,BUFF),(D,BUFF(1251))	PRNTBL 00021
COMPLEX TLIFT	PRNTBL 00022
COMPLEX TLIFT1,TLIFT2	PRNTBL 00023
DIMENSION PC(2), IPNT(2)	PRNTBL 00024
EQUIVALENCE (IPNT, TLIFT)	PRNTBL 00025
COMMON /RWBUFF/ BFCCDE,IBFCNT, BUFF(3280)	PRNTBL 00027
DATA PC / 10HFACE CONT1,4HNVED /	RWBUFF 00002
DATA BLANK /1H /	FTNCD 00086
DATA XINIT / -1.0 /	PRNTBL 0003
DATA LINEMX /50 /	PRNTBL 00029
C	PRNTBL 00030
C	PRNTBL 00031
XXVL = XXVAL(IKVAL)	PRNTBL 00032
IF(XKS(IKVAL).NE.XINIT) XXVL = XKS(IKVAL)	PRNTBL 00033
PAGE = 0	PRNTBL 00034
N = 1	PRNTBL 00035
M = 4	PRNTBL 00036
IF(M.GT.MYB) M = MYB	PRNTBL 00037
C	PRNTBL 00038
100 LINE = 100	PRNTBL 00039
200 DO 1400 I=IXB,MYB	PRNTBL 00040
DO 300 J=N,M	PRNTBL 00041
S(J) = 0.0	PRNTBL 00042
D(J) = 0.0	PRNTBL 00043
300 CONTINUE	PRNTBL 00044
IF(LINE.LE.50) GO TO 900	PRNTBL 00045
PAGE = PAGE + 1	PRNTBL 00046
LINE = 4	PRNTBL 00047
WRITE (NT6,9001) TITLE,TITL,XMACH,XXVL,IMODE	PRNTBL 00048
IF(PAGE.EQ.1) GO TO 700	PRNTBL 00049
WRITE(NT6,9005) PC	PRNTBL 00050
GO TO 800	PRNTBL 00051
700 WRITE(NT6,9005)	PRNTBL 00052
	PRNTBL 00053

800 CONTINUE	PRNTBL 00054
WRITE(NT6,6006) (BLANK,J,J=N,M)	PRNTBL 00055
WRITE(NT6,6007) (BLANK, J=N,M)	PRNTBL 00056
C	PRNTBL 00057
900 CONTINUE	PRNTBL 00058
JS = IPNTRM(2,I)	PRNTBL 00059
IF(JS.LE.0) GO TO 1400	PRNTBL 00060
IDX = IPNTRM(1,I)	PRNTBL 00061
JE = IPNTRM(1,I+1) - IDX + JS-	PRNTBL 00062
IF(JE.EQ.0) GO TO 1400	PRNTBL 00063
DO 1000 J=JS,JE	PRNTBL 00064
S(J) = REAL(ARRAY(IDX))	PRNTBL 00065
D(J) = AIMAG(ARRAY(IDX))	PRNTBL 00066
IDX = IDX + 1	PRNTBL 00067
1000 CONTINUE	PRNTBL 00068
DO 1200 J =N,M	PRNTBL 00069
IF(S(J)) 1300,1100,1300	PRNTBL 00070
1100 CONTINUE	PRNTBL 00071
IF(D(J)) 1300,1200,1300	PRNTBL 00072
1200 CONTINUE	PRNTBL 00073
GO TO 1400	PRNTBL 00074
1300 WRITE(NT6,9013) I, (S(J),D(J),J=N,M)	PRNTBL 00075
LINE = LINE + 1	PRNTBL 00076
1400 CONTINUE	PRNTBL 00077
C	PRNTBL 00078
M = M+4	PRNTBL 00079
N = N+4	PRNTBL 00080
IF(M.GT.MYB) GO TO 1900	PRNTBL 00081
IF(M.GT.MYB) M = MYB	PRNTBL 00082
IF(LINE.GT.45) GO TO 100	PRNTBL 00083
WRITE(NT6,6006) (BLANK,J,J=N,M)	PRNTBL 00084
WRITE(NT6,6007) (BLANK, J=N,M)	PRNTBL 00085
LINE = LINE+3	PRNTBL 00086
GO TO 200	PRNTBL 00087
1900 CONTINUE	PRNTBL 00088
RETURN	PRNTBL 00089
9001 FORMAT(1H1,20X,8A10,/50X,3A10,/ 46X,7H(MACH F5.3,5X,1/HRED.FREQ.	PRNTBL 00090
1 * =*,FB.5, *) * /52X,4HODE SHAPE*, 13)	PRNTBL 00091
9005 FORMAT(44X,42(1H-),20X,A10,A4)	PRNTBL 00092
9006 FORMAT(4HROW,A1,14X,5HCHORD,13.3(A1,22X,5HCHORD,13))	PRNTBL 00093
9007 FORMAT(3X, 4(A1,9X,4HREAL,8X,9HIMAGINARY))	PRNTBL 00094
9013 FORMAT(14,8E16.8)	PRNTBL 00095
END	PRNTBL 00096

C	SUBROUTINE PRINTSL(IMODE,SLIFT,TLIFT1,TLIFT2,MYBW,MYDT)	PRNTBL 00097
C		PRNTBL 00098
C	PRINTS THE SECTION LIFTS AND TOTAL LIFTS	PRNTBL 00099
C		PRNTBL 00100
C	IMODE - MODE SHAPE NUMBER	PRNTBL 00101
C	SLIFT - SECTION LIFT ARRAY	PRNTBL 00102
C	TLIFT1 - WING TOTAL LIFT	PRNTBL 00103
C	TLIFT2 - TAIL TOTAL LIFT	PRNTBL 00104
C		BCSFRB 00031
C	IF IMODE IS NEGATIVE THE PROGRAM WILL OUTPUT SECTION MOMENTS	BCSFRB 00032
C		BCSFRB 00033
C		PRNTBL 00105
	COMMON /PROBLM/ XMACH,IMODES,NTSLOP,NKVALS,SMOOTH,NDEG,CRDFIT,	PROBLM 00002
1	EXAIC,SUBDV,PLYWOOD	PROBLM 00003
	LOGICAL SMOOTH,CRDFIT,EXAIC,SUBDV,PLYWOOD	PROBLM 00004
	COMMON /CONTRL/ PREVEX,OMACH, TITLE(8), PRVGEOM,PRVMODE,DIHW,DIHT,	CONTRL 00002
1	DEFAULT	CONTRL 00003
	LOGICAL PRVGEOM,PRVMODE,DIHW,DIHT,DEFAULT	CONTRL 00004
	COMMON /KVAL / IKVAL,XKVAL(20), XKS(20)	KVAL 00002
	COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NBFAIC,NOUTP,	FILES 00002
1	IQUFSP,MODESC,IVPSC,IQEOSC,IWTFSC,IAICSC	FILES 00003
C		PRNTBL 00109
	COMPLEX SLIFT(1)	PRNTBL 00110
	COMPLEX TLIFT1,TLIFT2,TLIFT	PRNTBL 00111
	LOGICAL PRCH	BCSFRB 00035
	DATA BLANK/1H /	PRNTBL 00112
	DATA XINIT / -1.0 /	PRNTBL 00113
C		PRNTBL 00114
	IF (IMODE.LT.0) GO TO 100	BCSFRB 00036
	PRCH = .FALSE.	BCSFRB 00037
	GO TO 200	BCSFRB 00038
100	CONTINUE	BCSFRB 00039
	PRCH = .TRUE.	BCSFRB 00040
	IMODE = -IMODE	BCSFRB 00041
200	CONTINUE	BCSFRB 00042
C		PRNTBL 00115
	XVL = XVAL(IKVAL)	PRNTBL 00116
	IF(XKS(IKVAL).NE.XINIT) XVL = XKS(IKVAL)	PRNTBL 00117
	IF(PRCH) GO TO 300	BCSFRB 00043
	WRITE (NT6,6010) XMACH,XVL,IMODE	PRNTBL 00118
	GO TO 400	BCSFRB 00044
300	CONTINUE	BCSFRB 00045
	WRITE (NT6,9010) XMACH,XVL,IMODE	BCSFRB 00046
400	CONTINUE	BCSFRB 00047
	WRITE (NT6,6008)	PRNTBL 00119
	WRITE (NT6,6005)	PRNTBL 00120
	WRITE (NT6,6020)	PRNTBL 00121
	WRITE (NT6,6007) BLANK,BLANK,BLANK,BLANK	PRNTBL 00122
	DO 600 I=1,MYBW,4	PRNTBL 00123
	II = I + 3	PRNTBL 00124
	IF (II.GT. MYBW) II = MYBW	PRNTBL 00125
	WRITE (NT6,6030) I,(SLIFT(IJ),IJ=I,II)	PRNTBL 00126
600	CONTINUE	PRNTBL 00127
	IF(PRCH) GO TO 650	BCSFRB 00048
	WRITE (NT6,6023) TLIFT1	PRNTBL 00128
	IF(MYDT.EQ.0) GO TO 800	BCSFRB 00049
C		PRNTBL 00129

WRITE (NT6,6009)	PRINTBL 00130
GO TO 675	BCSFRB 00050
650 CONTINUE	BCSFRB 00051
IF(MYBT.EQ.0) GO TO 800	BCSFRB 00052
WRITE(NT6,9009)	BCSFRB 00053
675 CONTINUE	BCSFRB 00054
WRITE (NT6,6020)	PRINTBL 00131
WRITE (NT6,6007) BLANK,BLANK,BLANK,BLANK	PRINTBL 00132
DO 700 I=1,MYBT,4	PRINTBL 00133
II = I + 3	PRINTBL 00134
IF (II .GT. MYBT) II = MYBT	PRINTBL 00135
I2 = I + MYBW	PRINTBL 00136
II2 = II + MYBW	PRINTBL 00137
WRITE (NT6,6030) I,SLIFT(IJ),IJ=I2,II2)	PRINTBL 00138
700 CONTINUE	PRINTBL 00139
C	PRINTBL 00140
IF(PRCM) GO TO 800	BCSFRB 00055
WRITE (NT6,6024) TLIFT2	PRINTBL 00141
TLIFT = TLIFT1 + TLIFT2	PRINTBL 00142
WRITE (NT6,6025) TLIFT	PRINTBL 00143
800 CONTINUE	BCSFRB 00056
RETURN	PRINTBL 00144
C	PRINTBL 00145
C	PRINTBL 00146
6005 FORMAT(44X,32(1H-),30X,A10,A4)	PRINTBL 00147
6007 FORMAT(3X, 4(A1,9X,4HREAL,8X,9HIMAGINARY))	PRINTBL 00148
6008 FORMAT(58X,*WING*)	PRINTBL 00149
6009 FORMAT(1HD, // 53X, *SECTION LIFTS* / 58X, *TAIL* , / 44X, 32(1H-) /)	PRINTBL 00150
6010 FORMAT(1HD, 52X, 14HSECTION LIFTS / 44X, *(MACH *, F5.3, 5X,	PRINTBL 00151
1 *RED. FREQ. == F8.5, *) * / 52X, *MODE SHAPE*, I3)	PRINTBL 00152
6020 FORMAT(6HCHORD)	PRINTBL 00153
6023 FORMAT(1HD, 44X, * TOTAL LIFT - WING * / 1HD 40X, 2E16.8)	PRINTBL 00154
6024 FORMAT(1HD, 44X, * TOTAL LIFT - TAIL * / 1HD 40X, 2E16.8)	PRINTBL 00155
6025 FORMAT(1HD, 53X, * TOTAL LIFT * / 1HD, 40X, 2E16.8)	PRINTBL 00156
6030 FORMAT(I4, 8E16.8)	PRINTBL 00157
9009 FORMAT(1HD // 45X, *SECTIONAL MOMENT COEFFICIENTS* / 58X, *TAIL* /	BCSFRB 00057
1 44X, 32(1H-) /)	BCSFRB 00058
9010 FORMAT(1HD // // 45X, *SECTIONAL MOMENT COEFFICIENTS* / 44X, *(MACH *,	BCSFRB 00059
1 F5.3, 5X, *RED. FREQ. == F8.5, *) * / 52X, * MODE SHAPE*, I3)	BCSFRB 00060
END	PRINTBL 00158

SUBROUTINE PRNTAF (ARRAY, PRGNAC, GFAFC, GPPAFC)	PRNTAF 00002
C PRINTS COMPLEX GENERALIZED AIRFORCES, FROM COMPACT FORTRAN	PRNTAF 00003
C STORAGE	PRNTAF 00004
C	PRNTAF 00005
C ARRAY - ARRAY OF GENERALIZED AIR FORCES	PRNTAF 00006
C PRGNAC - LOGICAL FLAG FOR PRINT OPTION	PRNTAF 00007
C GFAFC - AGARD GENERALIZED AERODYNAMIC COEFFICIENT	PRNTAF 00008
C GPPAFC - AGARD GENERALIZED AERODYNAMIC COEFFICIENT	PRNTAF 00009
C	PRNTAF 00010
COMPLEX ARRAY (1)	PRNTAF 00011
DIMENSION GFAFC (1), GPPAFC (1)	PRNTAF 00012
LOGICAL PRGNAC	PRNTAF 00013
C	PRNTAF 00014
DIMENSION PC (2)	PRNTAF 00015
COMMON /PROBLM/ XMACH, NMODES, NTSLOP, NKVALS, SMOOTH, NDEG, CRDFIT,	PROBLM 00002
1 EXAIC, SUBDV, PLYWOOD	PROBLM 00003
LOGICAL SMOOTH, CRDFIT, EXAIC, SUBDV, PLYWOOD	PROBLM 00004
COMMON /KVAL / IKVAL, KVAL (20), KXS (20)	KVAL 00002
COMMON /FILES / NT5, NT6, INTA'E, INFSP, HPLAIC, NSPAIC, NOUTP,	FILES 00002
1 IOUFSP, MODESC, VPSC, IGEOBC, IWFSC, IAICSC	FILES 00003
INTEGER PAGE	PRNTAF 00022
DATA PC / 10HPAGE CONTI, 4HNUEJ /	FTNDI 00087
DATA BLANK / 1H /	FTNDI 00088
DATA XINIT / -1.0 /	FTNDI 00089
KVL = KVAL (IKVAL)	PRNTAF 00023
IF (KXS (IKVAL).NE. XINIT) KVL = KXS (IKVAL)	PRNTAF 00024
LINEMX = 50	PRNTAF 00025
PAGE = 0	PRNTAF 00026
J1 = 1	PRNTAF 00027
J2 = 4	PRNTAF 00028
IF (NMODES .LT. J2) J2 = NMODES	PRNTAF 00029
IJ1 = 1	PRNTAF 00030
IJ2 = (J2-1)*NMODES + 1	PRNTAF 00031
C	PRNTAF 00032
100 LINE = LINEMX + 10	PRNTAF 00033
110 DO 200 I = 1, NMODES	PRNTAF 00034
IF (LINE .LE. LINEMX) GO TO 170	PRNTAF 00035
PAGE = PAGE + 1	PRNTAF 00036
LINE = 0	PRNTAF 00037
WRITE (NT6, 6001) XMACH, KVL	PRNTAF 00038
IF (PAGE .EQ. 1) GO TO 150	PRNTAF 00039
WRITE (NT6, 6005) PC	PRNTAF 00040
GO TO 100	PRNTAF 00041
150 WRITE (NT6, 6005)	PRNTAF 00042
160 CONTINUE	PRNTAF 00043
C	PRNTAF 00044
WRITE (NT6, 6006) (BLANK, J, J=J1, J2)	PRNTAF 00045
WRITE (NT6, 6007) (BLANK, J=J1, J2)	PRNTAF 00046
C	PRNTAF 00047
170 CONTINUE	PRNTAF 00048
WRITE (NT6, 6010) I, (ARRAY (IJ), IJ= IJ1, IJ2, NMODES)	PRNTAF 00049
LINE = LINE + 1	PRNTAF 00050
IJ1 = IJ1 + 1	PRNTAF 00051
IJ2 = IJ2 + 1	PRNTAF 00052
200 CONTINUE	PRNTAF 00053
C	PRNTAF 00054
J1 = J1 + 4	PRNTAF 00055

```

      J2 = J2 + 4
      IF (J1 .GT. NMODES) GO TO 300
      IF (J2 .GT. NMODES) J2 = NMODES
      IJ1 = IJ1 + 3*NMODES
      IJ2 = IJ1 + (J2-J1) * NMODES
      IF (LINE .GT. LINEMX - 6) GO TO 100
      WRITE (NT6,6006) (BLANK,J, J = J1,J2)
      LINE = LINE+3
      GO TO 110

C
300 CONTINUE

C
      PRINT THE GENERALIZED AERODYNAMIC COEFFICIENTS
C      IF DESIRED.
C
      IF(.NOT.PRGMAC) GO TO 1400
      PAGE = 0
      DO 1300 IPR = 1,2
      J1 = 1
      J2 = 8
      IF (NMODES.LT.J2) J2 = NMODES
      IJ1 = 1
      IJ2 = (J2-1) * NMODES + 1

C
1100 LINE = LINEMX + 10
1110 DO 1200 I=1,NMODES
      IF(LINE.LE.LINEMX) GO TO 1170
      PAGE = PAGE + 1
      LINE = 8
      WRITE (NT6,7001) XMACH,XVL
      IF(IPR.EQ.2) GO TO 1140
      IF(PAGE.EQ.1) GO TO 1130
      WRITE (NT6,7005) PC
      GO TO 1100
1130 CONTINUE
      WRITE(NT6,7005)
      GO TO 1100
1140 CONTINUE
      IF (PAGE.EQ.1) GO TO 1130
      WRITE (NT6,7015) PC
      GO TO 1100
1150 WRITE(NT6,7015)
1160 CONTINUE

C
      WRITE (NT6,7006)
      WRITE (NT6,7007) (J,J=J1,J2)

C
1170 CONTINUE
      IF(IPR.EQ.2) GO TO 1180
      WRITE(NT6,8010) I, (GPAFC(IJ),IJ=IJ1,IJ2,NMODES)
      GO TO 1190
1180 CONTINUE
      WRITE (NT6,8010) I, (GPPAFC(IJ),IJ=IJ1,IJ2,NMODES)
1190 CONTINUE
      LINE = LINE + 1
      IJ1 = IJ1 + 1
      IJ2 = IJ2 + 1

```

```

PRNTAF 00056
PRNTAF 00057
PRNTAF 00058
PRNTAF 00059
PRNTAF 00060
PRNTAF 00061
PRNTAF 00062
PRNTAF 00063
PRNTAF 00064
PRNTAF 00065
PRNTAF 00066
PRNTAF 00067
PRNTAF 00068
PRNTAF 00069
PRNTAF 00070
PRNTAF 00071
PRNTAF 00072
PRNTAF 00073
PRNTAF 00074
PRNTAF 00075
PRNTAF 00076
PRNTAF 00077
PRNTAF 00078
PRNTAF 00079
PRNTAF 00080
PRNTAF 00081
PRNTAF 00082
PRNTAF 00083
PRNTAF 00084
PRNTAF 00085
PRNTAF 00086
PRNTAF 00087
PRNTAF 00088
PRNTAF 00089
PRNTAF 00090
PRNTAF 00091
PRNTAF 00092
PRNTAF 00093
PRNTAF 00094
PRNTAF 00095
PRNTAF 00096
PRNTAF 00097
PRNTAF 00098
PRNTAF 00099
PRNTAF 00100
PRNTAF 00101
PRNTAF 00102
PRNTAF 00103
PRNTAF 00104
PRNTAF 00105
PRNTAF 00106
PRNTAF 00107
PRNTAF 00108
PRNTAF 00109
PRNTAF 00110
PRNTAF 00111
PRNTAF 00112

```

1200 CONTINUE	PRNTAF 00113
C	PRNTAF 00114
J1 = J1 + 8	PRNTAF 00115
J2 = J2 + 8	PRNTAF 00116
IF (J1.GT.NMODES) GO TO 1300	PRNTAF 00117
IF (J2.GT.NMODES) J2 = NMODES	PRNTAF 00118
IJ1 = IJ1 + 3*NMODES	PRNTAF 00119
IJ2 = IJ1 + (J2-J1)* NMODES	PRNTAF 00120
IF (LINE.GT.LINEMX-6) GO TO 1100	PRNTAF 00121
WRITE (INT6,7006)	PRNTAF 00122
LINE = LINE + 3	PRNTAF 00123
GO TO 1110	PRNTAF 00124
C	PRNTAF 00125
1300 CONTINUE	PRNTAF 00126
1400 CONTINUE	PRNTAF 00127
RETURN	PRNTAF 00128
C	PRNTAF 00129
8001 FORMAT(1H1,51X,18HGENERALIZED FORCES /44X,6H(MACH ,F5.3,5X,*RED. F	PRNTAF 00130
1REQ. =* F8.5,1H))	PRNTAF 00131
8005 FORMAT(35X,50(1H-),20X,A10,A4)	PRNTAF 00132
8006 FORMAT(5H0 WT. / 6H FUNCT, 4(A2,23HVELOCITY POTENTIAL MODE,I3,3X))	PRNTAF 00133
8007 FORMAT(2X,4(A10,4HREAL,8X, 9HIMAGINARY))	PRNTAF 00134
8010 FORMAT(14, 8E16.8)	PRNTAF 00135
7001 FORMAT(1H1,44X,*GENERALIZED AERODYNAMIC COEFFICIENTS*, /	PRNTAF 00136
1 44X,6H(MACH , F5.3,5X,12HRED.FREQ. = F8.5,1H))	PRNTAF 00137
7006 FORMAT(5H0 WT. / 6H FUNCT,27X,*VELOCITY POTENTIAL MODES*,/ 1	PRNTAF 00138
7007 FORMAT(113,7I16)	PRNTAF 00139
7005 FORMAT(50X,*REAL PART*,/35X,50(1H-),20X,A10,A4)	PRNTAF 00140
7015 FORMAT(50X,*IMAGINARY PART*,/35X,50(1H-),20X,A10,A4	PRNTAF 00141
END	PRNTAF 00142

FUNCTION LOCSDW(IROW,JCOL,IPNTSD,IPNTIN,IPNTOT,IPNTLS)	LOCSDW 00002
C	LOCSDW 00003
C RETURNS THE LOCATION OF THE WORD IN THE END-AROUND SUBDIVIDED	LOCSDW 00004
C DOWNWASH ARRAY CORRESPONDING TO BOX(IROW,JCOL) OF THE SUB-	LOCSDW 00005
C DIVIDED BOX ARRAY	LOCSDW 00006
C	LOCSDW 00007
C IROW = BOX CHORDWISE LOCATION	LOCSDW 00008
C JCOL = BOX SPANWISE LOCATION	LOCSDW 00009
C IPNTSD = ARRAY OF POINTERS	LOCSDW 00010
C IPNTIN = NEXT AVAILABLE (UNUSED) CELL IN IPNTSD (END-	LOCSDW 00011
C AROUND)	LOCSDW 00012
C IPNTOT = FIRST CURRENTLY AVAILABLE CELL IN IPNTSD	LOCSDW 00013
C IPNTLS = LAST CELL OF IPNTSD (LENGTH OF ARRAY)	LOCSDW 00014
C RETURN -	LOCSDW 00015
C LOCSDW = LOCATION OF DESIRED DOWNWASH, IF SUCCESSFUL	LOCSDW 00016
C = 0, IF LOCPNT LIES OUTSIDE THE DEFINED AREA.	LOCSDW 00017
C	LOCSDW 00018
C DIMENSION IPNTSD(2,IPNTLS)	LOCSDW 00019
C	LOCSDW 00020
C LOCPNT = MOD(IROW-1,IPNTLS) + 1	LOCSDW 00021
C LOCPNT = LOCATION OF CELL IN IPNTSD WHICH WAS OR IS TO BE	LOCSDW 00022
C USED	LOCSDW 00023
C IF(IPNTIN - IPNTOT) 100, 300, 200	LOCSDW 00024
C END AROUND HAS OCCURRED	LOCSDW 00025
C 100 IF (LOCPNT - IPNTIN) 400, 300, 150	LOCSDW 00026
C NOT IN UPPER PART. IS LOCPNT WITHIN BOTTOM PART -	LOCSDW 00027
C 150 IF (LOCPNT - IPNTOT) 300, 400, 400	LOCSDW 00028
C	LOCSDW 00029
C NO END AROUND, NORMAL SEQUENCE	LOCSDW 00030
C 200 IF (LOCPNT - IPNTIN) 250, 300, 300	LOCSDW 00031
C LESS THAN UPPER LIMIT. IS LOCPNT .GE. LOWER LIMIT -	LOCSDW 00032
C 250 IF (LOCPNT .GE. IPNTOT) GO TO 400	LOCSDW 00033
C	LOCSDW 00034
C ERROR OR INITIAL CONDITION ENCOUNTERED (SHOULD NEVER OCCUR)	LOCSDW 00035
C 300 LOCSDW = 0	LOCSDW 00036
C GO TO 300	LOCSDW 00037
C	LOCSDW 00038
C SUCCESSFUL, BOX HAS BEEN DEFINED	LOCSDW 00039
C 400 IFB = IPNTSD(2,LOCPNT)	LOCSDW 00040
C IF(JCOL.LT.IFB) GO TO 300	LOCSDW 00041
C LOCSDW = IPNTSD(1,LOCPNT) + JCOL-IFB	LOCSDW 00042
C	LOCSDW 00043
C 500 CONTINUE	LOCSDW 00044
C RETURN	LOCSDW 00045
C END	LOCSDW 00046

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D		
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1. ORIGINATING ACTIVITY (Corporate author) The Boeing Company Commercial Airplace Group		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED
		2b. GROUP
3. REPORT TITLE PREDICTION OF UNSTEADY AERODYNAMIC LOADINGS OF NON-PLANAR WINGS AND WING-TAIL CONFIGURATIONS IN SUPERSONIC FLOW, Part II Computer Program Description		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Final Technical Report, December 1969 to April 1971		
5. AUTHOR(S) (First name, middle initial, last name) Gordon D. Kramer George E. Keylon		
6. REPORT DATE March 1972	7a. TOTAL NO. OF PAGES 399	7b. NO. OF REFS 2
8a. CONTRACT OR GRANT NO. F33615-70-C-1126	9a. ORIGINATOR'S REPORT NUMBER(S) D6-24860	
b. PROJECT NO. 1370		
c. Task No. 137003	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFFDL-TR-71-108, Part II	
d.		
10. DISTRIBUTION STATEMENT Distributions limited to U.S. Government agencies only; test and evaluation; statement applied August 1971. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FY).		
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY Air Force Flight Dynamics Laboratory Wright-Patterson AFB, OH 45433
13. ABSTRACT The Mach box technique has been extended to include wing and tail with dihedral angles and vertical separation. A digital computer program, written in FORTRAN, is presented. The program provides for up to nine sweep angles of the leading and trailing edges of each surface. First order piston theory thickness correction is available as an option and two refinement procedures are provided, subdivision with averaging and velocity potential smoothing. For a maximum of twenty oscillatory mode shapes the program calculates normal washes, velocity potentials, lifts, pressures and generalized forces matrices. If only one surface is being analyzed, sampling of wake up-wash, side-wash and longitudinal wash is available. The methods described in this report are intended to be used by airplane designers to calculate with improved accuracy, the unsteady aerodynamic loads that act on a lifting surface being propelled at supersonic speeds. The new feature of these calculations is that the aerodynamic interference between the wing and tail has been taken into account. These calculations are an essential ingredient of flutter analyses and will improve the confidence level of such calculations in preventing wing-tail flutter. The general requirement for such calculations are contained in Military Specification MIL-A-8870A (USAF).		

DD FORM 1 NOV 65 1473

UNCLASSIFIED

Security Classification

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Unsteady Aerodynamics Aeroelasticity Flutter Supersonic Flow Aerodynamic Interference						

UNCLASSIFIED

Security Classification