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

AD894188

DDC FILE COPY

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

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		
CFSTI	WHITE SECTION ☑	
DDC	BUFF SECTION ☑	
UNANNOUNCED	☐	
JUSTIFICATION		
....		
BY		
DISTRIBUTION/AVAILABILITY CODES		
DIST.	AVAIL. and/or 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.

AIR FORCE: 30-3-72/50

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

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. Ii, C. J. Borland and J. R. Hogley. Mr. Lawrence J. Huttsell 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

LIST OF ILLUSTRATIONS

NOMENCLATURE

Mathematical Symbol	FORTRAN 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.
$C_{\bar{\nu}\bar{\mu}\bar{\lambda}}$	C	non-dimensional	Velocity potential spatial aerodynamic influence coefficient (AIC).
$C_{\bar{\nu}\bar{\mu}\lambda}^{\left(\frac{xy}{ab}\right)}$	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 or a $\begin{cases}\text{R-right}\\ \text{L-left}\end{cases}$ y or b $\begin{cases}\text{W-wing}\\ \text{T-tail}\end{cases}$
$C_{\bar{\nu}\bar{\mu}0}$, $C_{\bar{\nu}\bar{\mu}}$	PKERNL	non-dimensional	Velocity potential planar AIC
$C^{s}_{\bar{\nu}\bar{\mu}0}$	SKERNL	non-dimensional	Planar AIC defined for subdivided sending boxes.
$\Delta\bar{C}_{p_j}^{n,m}$	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
$C_{\ell_j}^{m}$	--	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
c_{rW}, c_{rT}	---	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
ij	---	force/length	Generalized stiffness
s	XKS	Non-dimensional	Reduced frequency based on leading planform semi-span, $k_s = \frac{\omega s}{U}$

NOMENCLATURE

Mathematical 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
$\bar{\bar{L}}_j^{n,m}$	BXL1FT	non-dimensional	Non-dimensional amplitude of box lift (program output) $L_j^{n,m} = \bar{L}_j^{n,m} e^{i\omega t} = q\left(\frac{b_1}{\beta}\right) \bar{\bar{L}}_j^{n,m} e^{i\omega t}$

NOMENCLATURE

Mathematical Symbol	FORTRAN Symbol	Dimension	Definition
$\overline{\overline{L}}_j^{\,m}$	SLIFT	non-dimensional	nondimensional amplitude of section lift (program output) $L_j^n = \bar{L}_j^n e^{i\omega t} = q\left(\frac{b_1}{\beta}\right)\overline{\overline{L}}_j^{\,n} e^{i\omega t}$
$\overline{\overline{L}}_j$	TLIFT	non-dimensional	Nondimensional amplitude of total lift (program output) $L_j = \bar{L}_j e^{i\omega t} = q\left(\frac{b_1}{\beta}\right)\overline{\overline{L}}_j e^{i\omega t}$
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
$N_{xyz}^{n,m}$	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; $x = \begin{cases}\text{R-right}\\ \text{L-left}\end{cases}$ $y = \begin{cases}\text{U-upper}\\ \text{L-lower}\end{cases}$ $z = \begin{cases}\text{W-wing}\\ \text{T-tail}\end{cases}$ e.g. $N_{RUW}^{n,m}$ means normal wash on the right upper wing at box n,m

NOMENCLATURE

Mathematical 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. $\hat{N}^{n,m}_{RUT\ LLW}$ means normal 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
$\bar{n}_c, \bar{m}_c, \bar{\ell}_c$	---	---	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

Mathematical Symbol	FORTRAN 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); $Q_{ij} = \bar{Q}_{ij}e^{i\omega t} = q\frac{b_1}{\beta}\bar{\bar{Q}}_{ij}e^{i\omega t}$
$\hat{Q}_{ij}$	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

Mathematical Symbol	FORTRAN 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.
$U_{\bar{\nu}\bar{\mu}\bar{\lambda}}$	---	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.
$V_{\bar{\nu}\bar{\mu}\bar{\lambda}}$	V	non-dimensional	Velocity spatial aerodynamic influence coefficient (AIC) for velocity normal to the free stream and parallel to the sending surface.
$V^{\binom{xy}{ab}}_{\bar{\nu}\bar{\mu}\bar{\lambda}}$	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).
$W_{\bar{\nu}\bar{\mu}\bar{\lambda}}$	W	non-dimensional	Velocity spatial aerodynamic influence coefficient (AIC) for velocity normal to the sending plane.

NOMENCLATURE

Mathematical Symbol	FORTRAN Symbol	Dimension	Definition
$w^{\binom{xy}{ab}}_{\overline{\nu\mu\lambda}}$	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	FORTRAN 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.
$\begin{Bmatrix} X_F \\ X_c \end{Bmatrix}$	$\begin{Bmatrix} \text{XEDGE} \\ \text{XCENTR} \end{Bmatrix}$	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
$\begin{Bmatrix} Y_{WLE} \\ Y_{TLE} \end{Bmatrix}$	$\begin{Bmatrix} \text{YWLE} \\ \text{YTLE} \end{Bmatrix}$	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)
$\bar{\bar{Z}}_\tau$	--	non-dimensional	Thickness slope piston theory correction; $\bar{\bar{Z}}_\tau = 1 + \frac{\gamma+1}{2} M \frac{\partial Z_\tau}{\partial x}$
$\alpha^{n,m}$	ALPHA	non-dimensional	Edge box area ratio for box(n,m).
β	BETA	non-dimensional	$\sqrt{M^2 - 1}$

NOMENCLATURE

Mathematical Symbol	FORTRAN Symbol	Dimension	Definition
b_1/β	BIBETA	length	Spanwise dimension of the Mach box.
b_{1_s}/β	BIBTAS	length	Spanwise dimension of a subdivided 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	$\sin^{-1}\frac{\bar{\eta}}{\tau} = \sin^{-1}\frac{\bar{\eta}}{\sqrt{\bar{\xi}^2 - \bar{\lambda}^2}}$
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	FORTRAN 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 ot X_T coordinate.
$\bar{\xi}$	XIBAR	non-dim 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 $\frac{\partial \phi}{\partial x_i}$ is velocity, positive in positive x_1 direction, where $x_i = X, Y$, 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: $\Delta\phi(x,y,t) = \phi_{upper} - \phi_{lower}$

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
$\Delta\bar{\bar{\phi}}_j(x,y)$, $\Delta\bar{\bar{\phi}}_j^{n,m}$	DELPHI	non-dimensional	Non-dimensional velocity potential difference due to the unit j^{th} generalized coordinate (program output); $\Delta \phi_j^{n,m} = \overline{\Delta\phi}_j^{n,m} e^{i\omega t} = \frac{U}{b_1}\left(\frac{b_1}{\beta}\right)\Delta\bar{\bar{\phi}}_j^{n,m} e^{i\omega t}$
$\Delta\bar{\bar{\phi}}_j(X_{TE},y)$, $\Delta\bar{\bar{\phi}}_{j\,TE}^{m}$	TVP	non-dimensional	Trailing edge velocity potential difference.
$\begin{Bmatrix}\Psi_W\\ \Psi_T\end{Bmatrix}$	$\begin{Bmatrix}\text{PSIW}\\ \text{PSIT}\end{Bmatrix}$	degrees	Dihedral angle of $\begin{Bmatrix}\text{wing}\\ \text{tail}\end{Bmatrix}$, 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. $C_{\bar{\nu}\bar{\mu}\chi}$, $V_{\bar{\nu}\bar{\mu}\chi}$, $W_{\bar{\nu}\bar{\mu}\chi}$.

GLOSSARY OF TERMINOLOGY

Sub-box - A member of the array of boxes formed when the grid of sending 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);

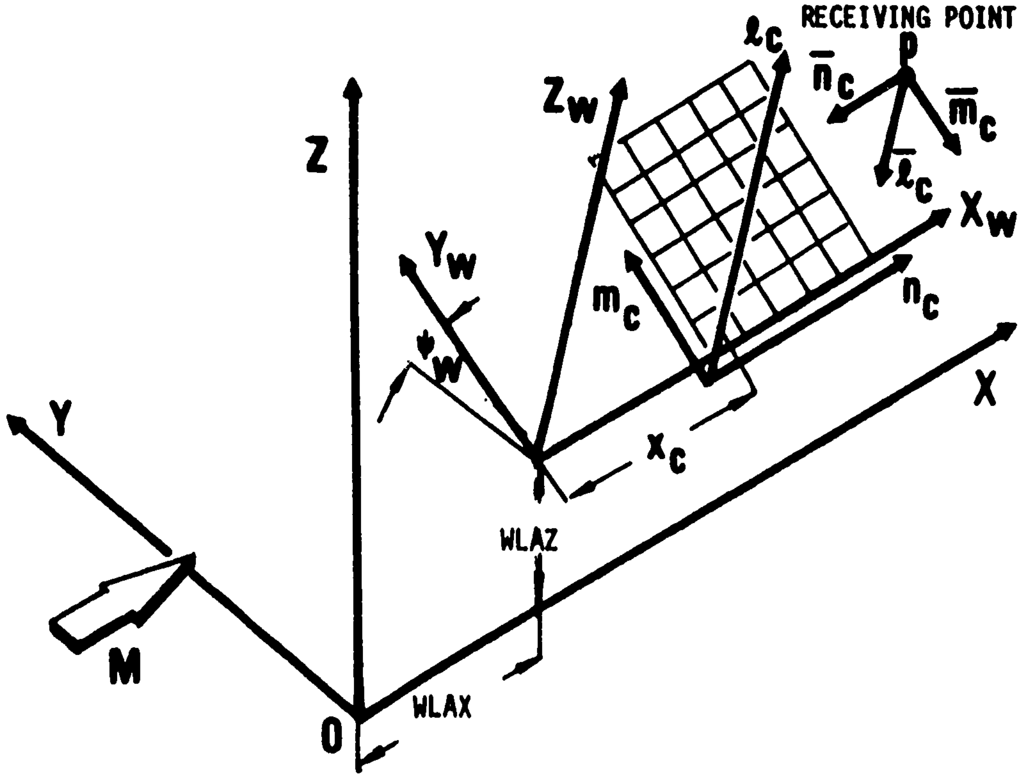


Figure 1 Coordinate Systems For A Right Wing

Symbol	Transformation	Definition	Dimension
X Y Z	Global or Reference Coordinate System. X positive Aft, Y positive Right, Z positive Upward		Length
X_w Y_w Z_w	$X - WLAX$ $Y\cos\phi_w + (Z - WLAZ)\sin\phi_w$ $(Z - WLAZ)\cos\phi_w - Y\sin\phi_w$	Wing Local Coordinate System, used to define wing leading and trailing edges. X_T, Y_T, Z_T are similarly defined for the tail local axes	Length
n_c m_c ℓc	$(X_w - X_c)/b_1 + 1$ $Y_w/(b_1/\beta) + 1/2$ $Z_w/(b_1/\beta)$	Sending Surface Coordinate System used to define box grid. The (n_c, m_c) plane lies within the plane of the sending surface, in this case the right wing.	Non-dimensional

Figure 1 (Cont'd)

Symbol	Transformation	Definition	Dimension
$\overline{n}_c$ $\overline{m}_c$ $\overline{l}_c$	$-(n_c-n)$ $-(m_c-m)$ $-(l_c-l)$	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-dimensional

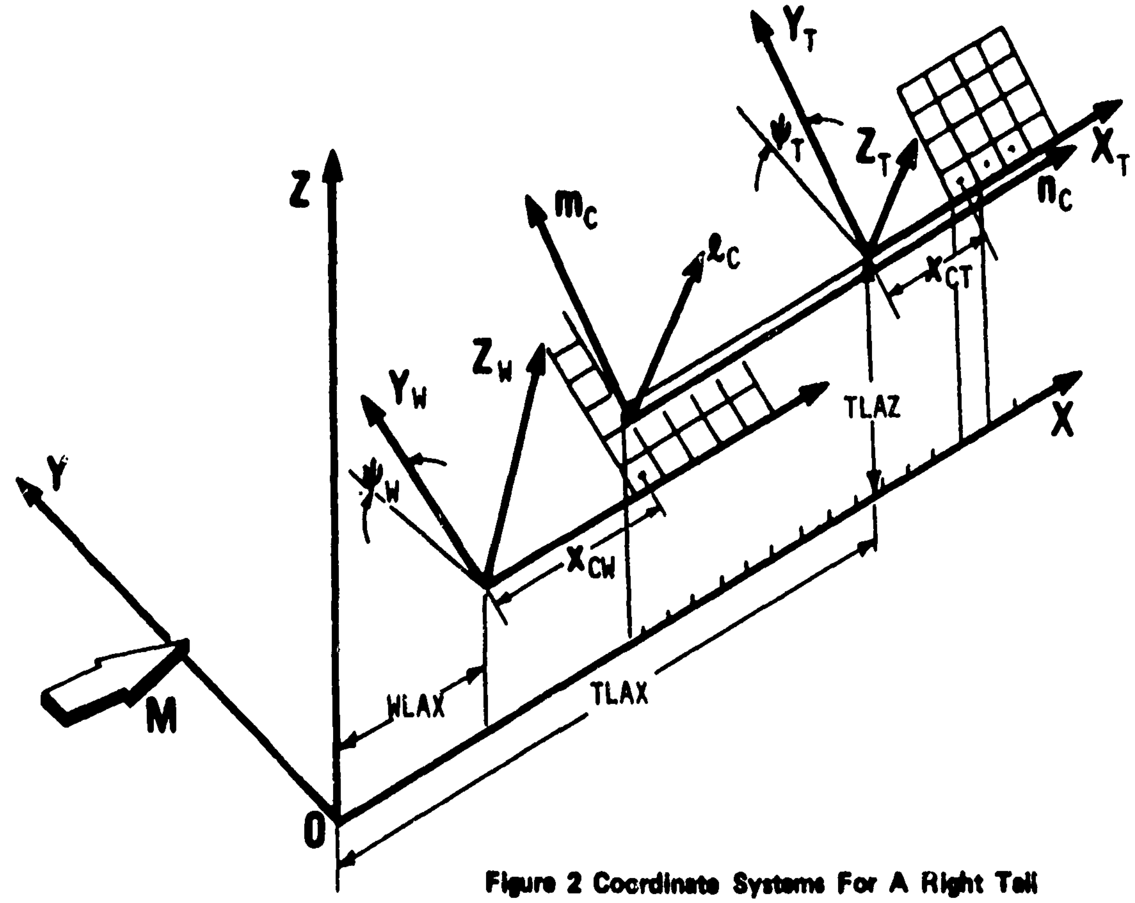


Figure 2 Coordinate Systems For A Right Tail

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 ℓ_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) Subdivision 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.

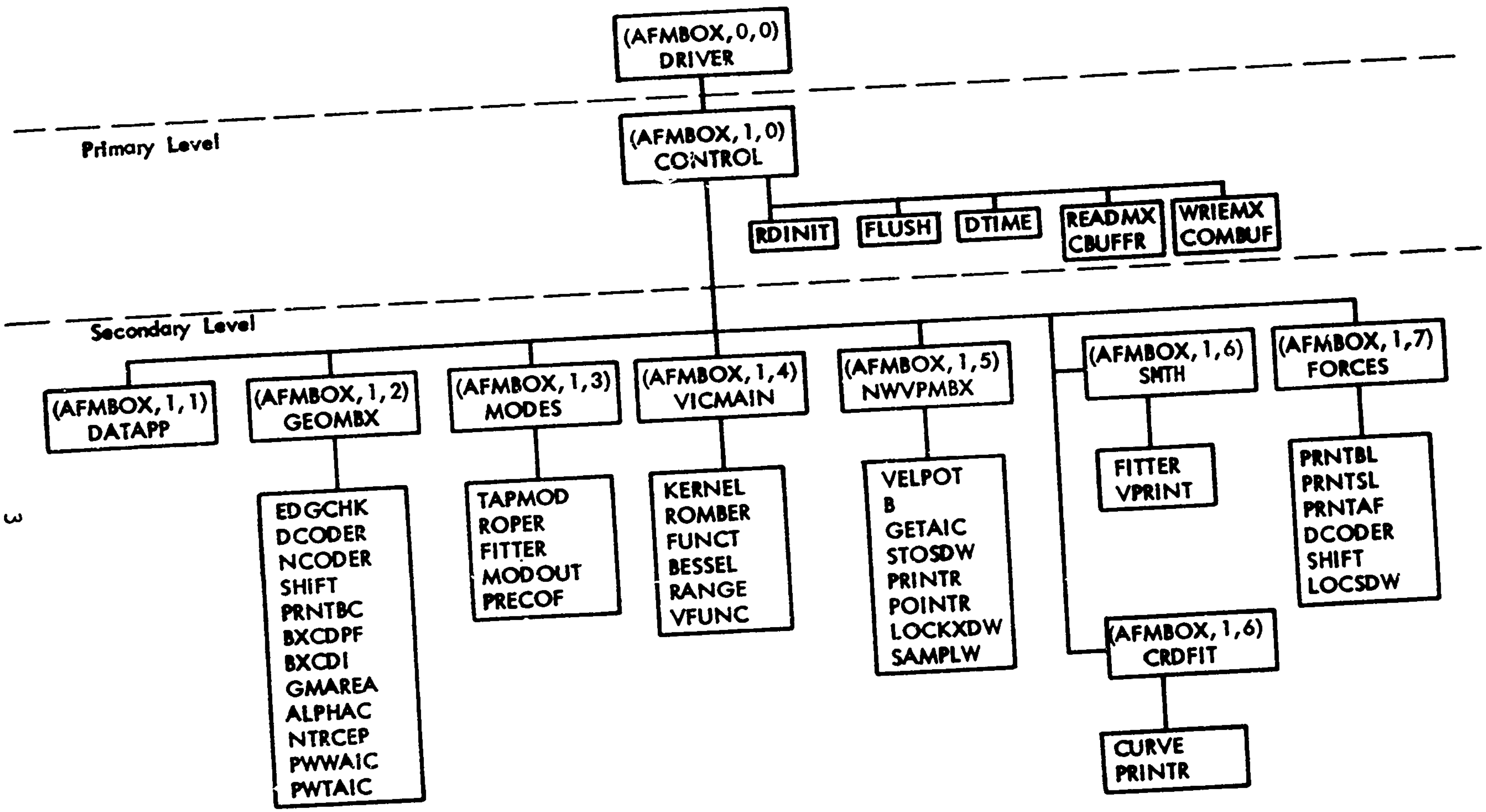


Figure 3 Program Overlay Structure

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

Fortran Subprogram CBUFFR

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 ·ll 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	
B1BETA	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 contributing 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:

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

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

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

LSROWS = Maximum number of subdivided rows allowed.

In/Out Parameters:

$\left.\begin{matrix}\text{IXBW}\\ \text{IXBT}\end{matrix}\right\}$ = 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:

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

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

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

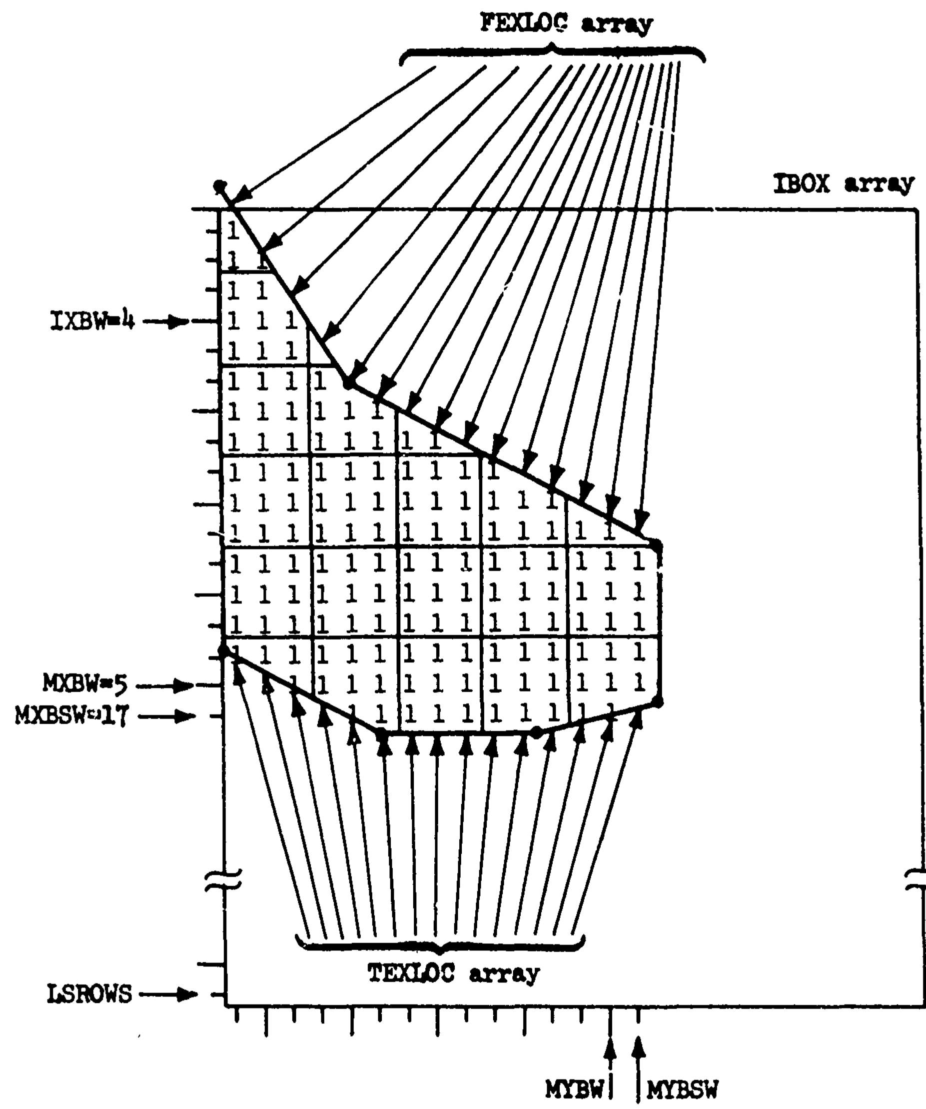


FIGURE 4 Output from BXCDPF, Called for a Wing (NSUBDV=3)

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

$\left.\begin{matrix} \text{FEXLOC} \\ \text{TEXLOC} \end{matrix}\right\} = \begin{Bmatrix} \text{Leading} \\ \text{Trailing} \end{Bmatrix}$ 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

$\left.\begin{matrix}\text{MXBBSW}\\ \text{MXBBST}\end{matrix}\right\}$	Maximum X extension of the subdivided $\left\{\begin{matrix}\text{wing}\\ \text{tail}\end{matrix}\right\}$ box pattern, including diaphragm
IXBST	X-location of the first subdivided tail row
$\left.\begin{matrix}\text{MYBSW}\\ \text{MYBST}\end{matrix}\right\}$	Maximum Y extension of the subdivided planform $\left\{\begin{matrix}\text{wing}\\ \text{tail}\end{matrix}\right\}$ pattern
$\left.\begin{matrix}\text{MYBBSW}\\ \text{MYBBST}\end{matrix}\right\}$	Maximum Y extension of the subdivided $\left\{\begin{matrix}\text{wing}\\ \text{tail}\end{matrix}\right\}$ pattern, including diaphragm
NSUBDV	Number of subdivisions

Common Output

$\left.\begin{matrix}\text{MYBBSW}\\ \text{MYBBST}\end{matrix}\right\}$	Modified, if necessary
$\left.\begin{matrix}\text{MYBBW}\\ \text{MYBBT}\end{matrix}\right\}$	Maximum Y extension of the unsubdivided $\left\{\begin{matrix}\text{wing}\\ \text{tail}\end{matrix}\right\}$ pattern, including diaphragm

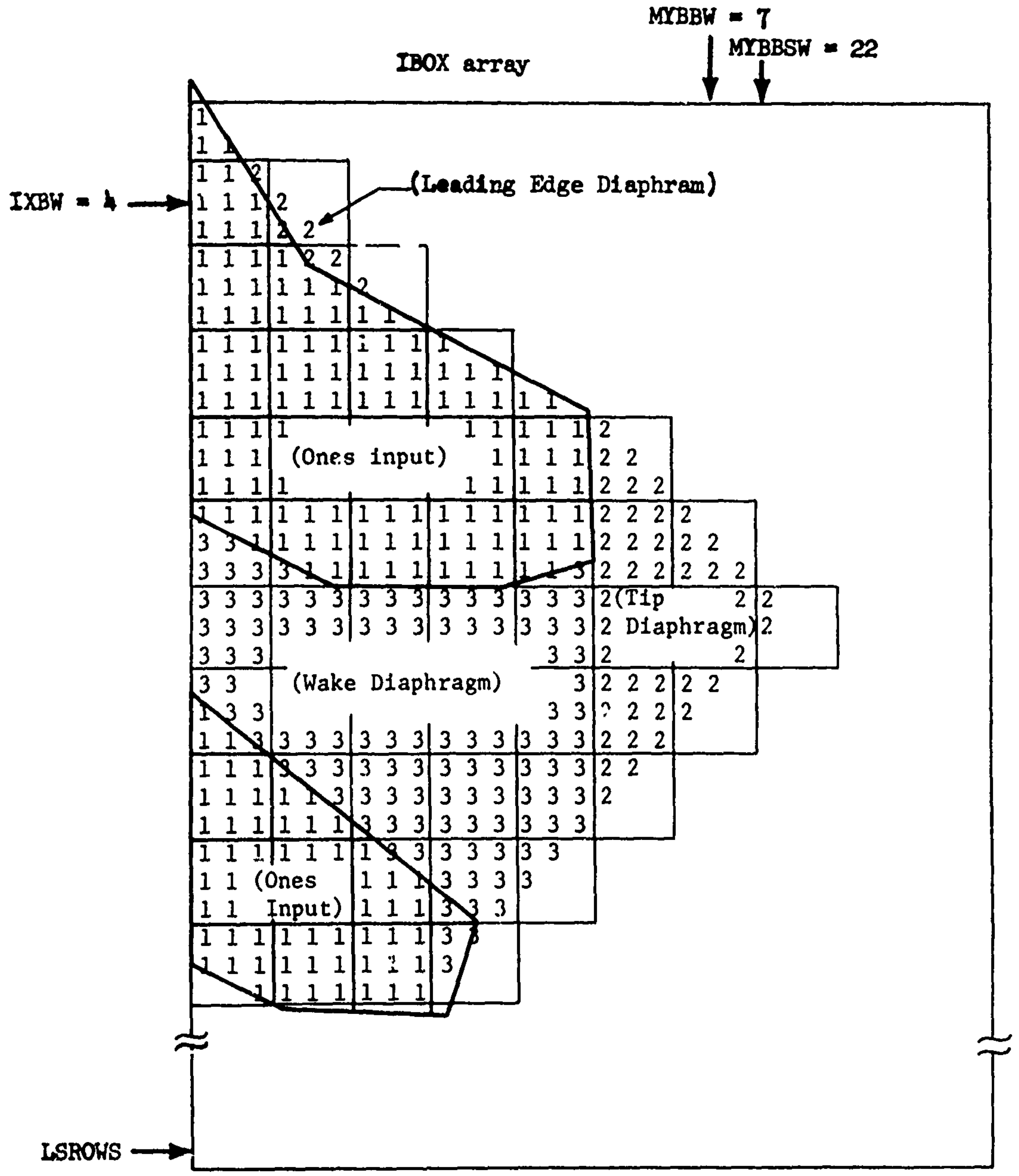


Figure 5 Output from BXCDI, called for a Coplanar Wing and Tail (NSUBDV = 3)

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 enencountered.

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.

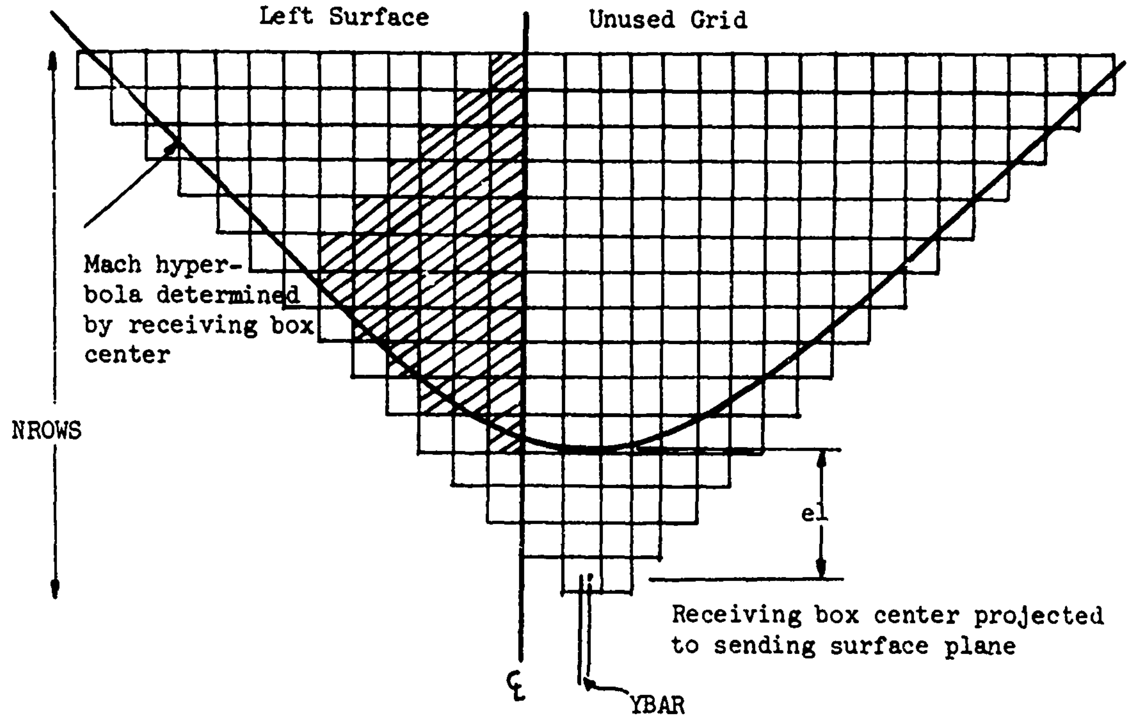


FIGURE 6. Possible Arrangement of Sending Boxes, Left Surface to Right
(Actual sending boxes shaded.)

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 (MUAIC 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 vox patterns, any sending boxes are indicated in the MUAIC 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 $= \phi_T - \phi_w$
NSUBDV
PSIW
MYBBW
MYBSW
IXBW
PSIT
FEXLOC
TEXLOC
SYM

Output Parameters

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

$\begin{Bmatrix} SURF \\ SURFL \end{Bmatrix}$ = .T., Contributing boxes were found on the $\begin{Bmatrix} right \\ left \end{Bmatrix}$ wing

= .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 paralled 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

$\left.\begin{matrix}\text{YEDG}\\ \text{XEDG}\end{matrix}\right\}$ = Arrays of $\left\{\begin{matrix}Y\\ X\end{matrix}\right\}$ locations of the edge definition points

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 partically 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 Paramters:

X = X coordinate of box center

XLED, YLED = Planform leading edge definition points

XTED, YTED = Planform trailing edge definition points

L1, C1, R1 = Left, Center, Right Chord edge intersections with the planform leading edge

NBK1 = Number of planform leading edge definition points within the chord

K1 = First leading edge definition point within chord

L2, C2, R2, NBK2, K2 Same as above for trailing edge

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.

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

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.

$\bar{\bar{Z}}_{\tau}(x,Y)$ 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:

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

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.

$$\text{Slope} = \sum_{i=0}^{\#\text{ of degrees}} d\left(a_{i0} x^i y^0 + \ldots + a_{0i} x^0 y^i \right) / dx \qquad (3)$$

$$= \sum_{i=1}^{\#\text{ of degrees}} \left(i a_{i0} x^{i-1} y^0 + \cdots + a_{1i} y^i \right)$$

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 arithmatic 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(IDEG, A, IFR)

Input:

Labeled common block /FILES/

IDEG - 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:

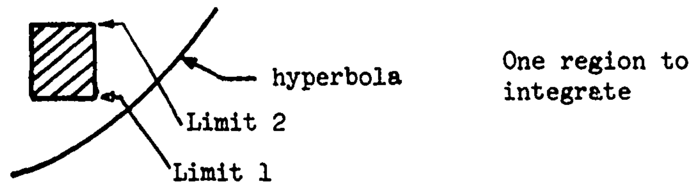


FIGURE 8 AIC Integration, Full Box

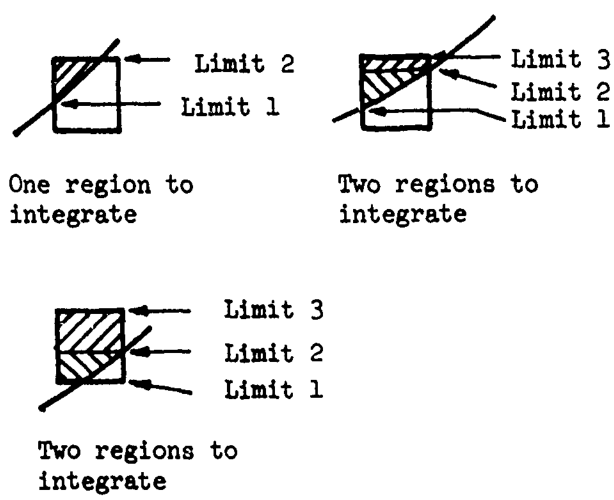


FIGURE 9 AIC Integration Edge Boxes

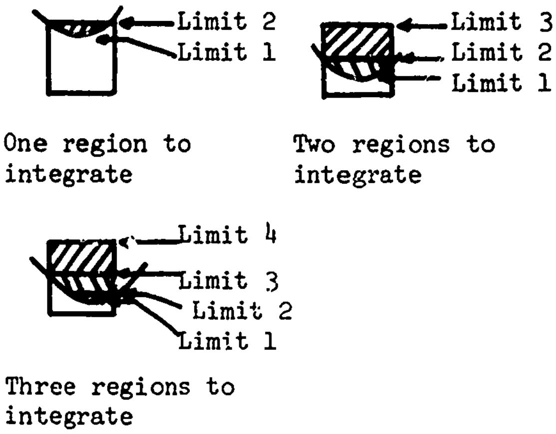


FIGURE 10 AIC Integration Apex Boxes

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 ($C_{\bar\nu\bar\mu\lambda}$), the upwash aerodynamic influence coefficients ($W_{\bar\nu\bar\mu\lambda}$), and the sidewash aerodynamic influence coefficients ($V_{\bar\nu,\bar\mu,\lambda}$).

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

$$C_{\bar\nu\bar\mu\lambda} = -\frac{1}{\pi}\int_{\bar\xi_l}^{\bar\xi_u} e^{-i\bar k_1\bar\xi}\left\{J_0\left(\frac{\bar k_1}{M}\sqrt{\bar\xi^2-\ell^2}\right)\left[\sin^{-1}\left(\frac{\bar\eta_u}{\sqrt{\bar\xi^2-\ell^2}}\right)\right.\right.$$

$$\left.-\sin^{-1}\left(\frac{\bar\eta_L}{\sqrt{\bar\xi^2-\ell^2}}\right)\right]+\sum_{r=1}^{\infty}\frac{(-1)^r}{r}J_{2r}\left(\frac{\bar k_1}{M}\sqrt{\bar\xi^2-\ell^2}\right)$$

$$\left.\left[\sin\left(2r\sin^{-1}\left(\frac{\bar\eta_u}{\sqrt{\bar\xi^2-\ell^2}}\right)\right)-\sin\left(2r\sin^{-1}\left(\frac{\bar\eta_L}{\sqrt{\bar\xi^2-\ell^2}}\right)\right)\right]\right\}d\bar\xi$$

$$W_{\bar{\nu}\bar{\mu}\bar{\lambda}} = \frac{\ell}{\pi}\Bigg[\int_{\bar{\xi}_\ell}^{\bar{\xi}_u} e^{-i\bar{k}_1\xi}\Bigg\{J_0\left(\frac{\bar{k}_1}{M}\sqrt{\bar{\xi}^2-\ell^2}\right)\Bigg[\sin^{-1}\left(\frac{\bar{\eta}_u}{\sqrt{\bar{\xi}^2-\ell^2}}\right) -$$

$$\sin^{-1}\left(\frac{\bar{\eta}_L}{\sqrt{\bar{\xi}^2-\ell^2}}\right)\Bigg] + \sum_{r=1}^{\infty}\frac{(-1)^r}{r}J_{2r}\left(\frac{\bar{k}_1}{M}\sqrt{\bar{\xi}^2-\ell^2}\right)\cdot$$

$$\left[\sin\left(2r\sin^{-1}\left(\frac{\bar{\eta}_u}{\sqrt{\bar{\xi}^2-\ell^2}}\right)\right) - \sin\left(2r\sin^{-1}\left(\frac{\bar{\eta}_L}{\sqrt{\bar{\xi}^2-\ell^2}}\right)\right)\right]\Bigg\}\frac{1+i\bar{k}_1\bar{\xi}}{\bar{\xi}^2}\,d\bar{\xi}$$

$$+\Bigg|\frac{e^{-ik_1\bar{\xi}}}{\bar{\xi}}\Bigg\{J_0\left(\frac{\bar{k}_1}{M}\sqrt{\bar{\xi}^2-\ell^2}\right)\left[\sin^{-1}\left(\frac{\bar{\eta}_u}{\sqrt{\bar{\xi}^2-\ell^2}}\right)-\sin^{-1}\left(\frac{\bar{\eta}_L}{\sqrt{\bar{\xi}^2-\ell^2}}\right)\right]$$

$$+\sum_{r=1}^{\infty}\frac{(-1)^r}{r}J_{2r}\left(\frac{\bar{k}_1}{M}\sqrt{\bar{\xi}^2-\ell^2}\right)\left[\sin\left(2r\sin^{-1}\left(\frac{\bar{\eta}_u}{\sqrt{\bar{\xi}^2-\ell^2}}\right)\right)\right.$$

$$\left.-\sin\left(2r\sin^{-1}\left(\frac{\bar{\eta}_L}{\sqrt{\bar{\xi}^2-\ell^2}}\right)\right)\right]\Bigg\}\Bigg[_{\bar{\xi}_\ell}^{\bar{\xi}_u} + e^{-i\bar{k}_1\bar{\xi}}\frac{\pi}{\ell}\Bigg] \tag{5}$$

The $C_{\bar{\nu}\bar{\mu}\bar{\lambda}}$ equations is valid for all types of boxes, which are shown in Figures 8, 9, and 10. The $W_{\bar{\nu}\bar{\mu}\bar{\lambda}}$ 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 $V_{\bar{\nu}\bar{\mu}\bar{\lambda}}$ is considerably different and holds for all regions to be integrated. The evaluation of the integral is done by subroutine VFUNC. The equation is:

$$V_{\bar{\nu}\bar{\mu}\lambda} = -\frac{M}{\pi \bar{k}_1}\left[\left| \frac{e^{-i\bar{k}_1\bar{\xi}}}{\bar{\xi}} \quad \frac{1+i\bar{k}_1\bar{\xi}}{\bar{\xi}^2}\left(\sin\left(\frac{\bar{k}_1}{M}\sqrt{\bar{\xi}^2-\bar{\eta}_u^2-\ell^2}\right)\right.\right.\right.$$

$$\left.\left.-\sin\left(\frac{\bar{k}_1}{M}\sqrt{\bar{\xi}^2-\bar{\eta}_L^2-\ell^2}\right)\right)\right]_{\bar{\xi}_L}^{\bar{\xi}_u} + \int_{\bar{\xi}_\ell}^{\bar{\xi}_u}\left[e^{-i\bar{k}_1\bar{\xi}}\,\frac{1+i\bar{k}_1\bar{\xi}}{\bar{\xi}^2}\right.$$

$$\left.\left(\sin\left(\frac{\bar{k}_1}{M}\sqrt{\bar{\xi}^2-\bar{\eta}_u^2-\ell^2}\right)-\sin\left(\frac{\bar{k}_1}{M}\sqrt{\bar{\xi}^2-\bar{\eta}_L^2-\ell^2}\right)\right)\right]\,d\bar{\xi} \qquad (6)$$

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

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

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

<u>Input</u>:

XMACH	Mach number
K1	Reduced frequency
ERR	Convergence criteria (relative not absolute)

Labeled Common Block /VICPAR/

<u>Output</u>:

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, $C_{\bar{\nu}\bar{\mu}\lambda}$

W - Upwash aerodynamic influence coefficient, $W_{\bar{\nu}\bar{\mu}\lambda}$

V - Sidewash aerodynamic influence coefficient, $V_{\bar{\nu}\bar{\mu}\lambda}$

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, K1BAR, EL, YMUBAR, IUC, XMACH, BESSY)

Input:

K - Number of functions to evaluate
XI - Array of independent variables
IFLAG - Indicator of real or imaginary parts
= 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 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 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 NWVLPT

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 IGEOSC. The mode shape and velocity potential pointer array IPNTRM is read from scratch file MODESC, 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 MODESC into array DEFSL. Subroutine VELPOT is called for the wing to compute N_{RUW}, N_{RLW}, and $\Delta\bar{\bar{\phi}}$ at box centers, and trailing edge $\Delta\bar{\bar{\phi}}$ 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 $\Delta\bar{\bar{\phi}}$ is done in routine PRINTR.

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

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

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

CALL OVERLAY (6HAFMBØX, 1, 5, 0)

Input: Uses labelled common blocks

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

Uses scratch files

IGEOSC
IMODESC

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,i) = the location of the first box value for row i, and IPNTR (2, i) = the chord number of the first box value for row i.

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
ITPE = 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:

NWWK NTTK NRWTK NLWTK	= Number of AIC arrays available for influence.	wing-wing tail-tail right wing-tail left wing-tail

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_{---}
W = $W^{\nu\mu\omega}_{---}$
V = $V^{\nu\mu\omega}_{---}$
$\nu\mu\omega$

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:

$$\frac{D\bar{f}_j^{n,m}}{Dt} = \left[i k_1 f_j^{n,m} + b_1 \frac{\partial f_j^{n,m}}{\partial x} \right] \tag{8}$$

(1) For the wing:

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

where

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

and

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

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

(2) For leading edge or tip diaphragm boxes:

$$\left(N_{RUS} - N_{RLS}\right)^{n,m} = \frac{1}{C_{000}} \Big[-\sum_{\substack{\text{right wing} \\ +\text{diaphragm}}} C_{\bar{\nu}\bar{\mu}0} \left(N_{RUW} - N_{RLW}\right)^{\nu\mu}$$

$$+ SYM * \sum_{\substack{\text{left wing} \\ +\text{diaphragm}}} C_{\bar{\nu}\bar{\mu}\bar{\lambda}} \left(N_{RUW}^{\nu\mu} - N_{RLW}^{\nu\mu}\right) \Big] \tag{12}$$

(3) For wake diaphragm boxes:

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

$$+\ SYM*\sum_{\substack{\text{left wing}\\ \text{+diaphragm}}} C_{\nu\mu\lambda}\left(N_{RUW}^{\nu\mu}-N_{RLW}^{\nu\mu}\right)\Bigg] \qquad (13)$$

where

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

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

(4) For the tail:

$$\left(N_{RUT}-N_{RLT}\right)^{n,m}=2\left(\frac{D\bar{f}}{Dt}^{n,m}\right)+\left(\hat{N}_{\substack{RLT\\ LLT}}^{n,m}-\hat{N}_{\substack{RUT\\ LUT}}^{n,m}\right)$$

$$-2\hat{N}_{\substack{RUT\\ RW}}^{n,m}-2\hat{N}_{\substack{RUT\\ LW}}^{n,m} \qquad (16)$$

where $\hat{N}_{\substack{RUT\\ LUT}}$ and $\hat{N}_{\substack{RLT\\ LLT}}$ are computed as in Equation (11) and (12).

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

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

(5) Velocity potentials:

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

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.

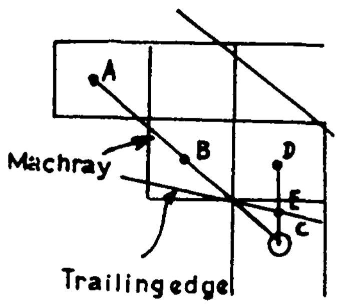
For each box, the $\hat{N}$ terms are first zeroed out. If spatial contribution is present, subroutine GETAIC is called to get the necessary AIC arrays from scratch file IAICSC, and the proper summation is computed over the forward Mach hyperbola. This is first done for $\hat{N}_{\substack{RUS\\LUS}}$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

If the surface is a tail, a similar procedure of getting AIC arrays and computing the proper summation for $\hat{N}_{\substack{RUT\\RW}}$ and $\hat{N}_{\substack{RUT\\LW}}$ 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 $\Delta\bar{\bar{\phi}}$ 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 ex= trapolated 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 $\Delta\bar{\bar{\phi}}$ 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.

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

USAGE: 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) $C_{\bar{\nu}\bar{\mu}0}$ array.
SKERNL	Subdivided $C^{K_1/N_s}_{\bar{\nu}\bar{\mu}0}$ 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, FEXLOC}	{leading, trailing} edge X-locations at chord centers
	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	$\Delta\bar{\bar{\phi}}$ array
	TVP	$\Delta\bar{\bar{\phi}}_{TE}$ array
/NWASHES/	ENRUS	N_{RUW} or N_{RUT}
	ENRLS	N_{RLW} or N_{RLT}
	IOVLAPN	Measure of tail overlap of wing diaphragm, for normal washes.
/SNWASH/	IPNTSD	Pointer array for subdivided normal washes
	ENSUBD	$N^{(S)}_{RUS}$ and $N^{(S)}_{RLS}$
	IPNTIN, IPNTOT, IPNTLS	End-around pointers for array IPNTSD

Fortran Function B

AUTHOR: G. D. Kramer

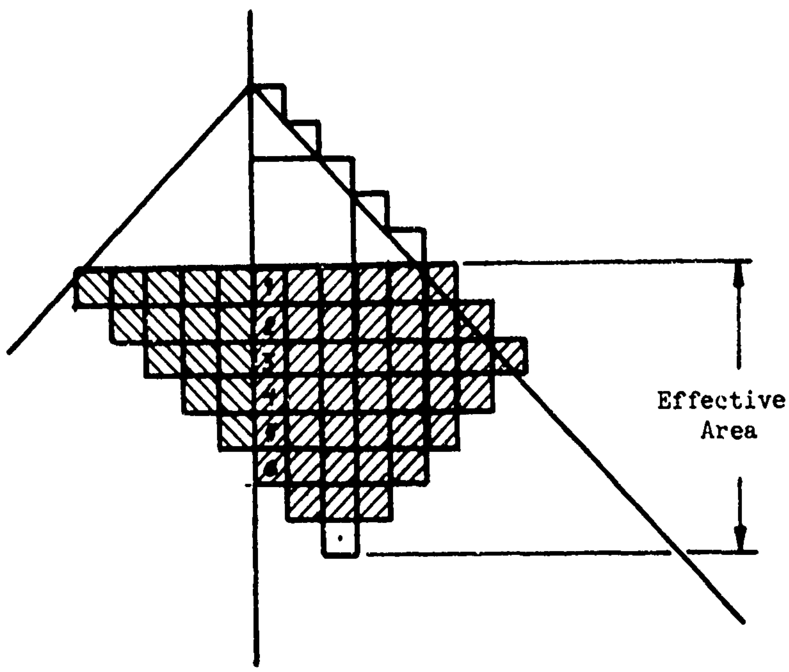
PURPOSE: Given the location of a Mach box, to compute the planar contribution of the rest of the surface to the velocity potential difference for the box.

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

$$B_s = \sum_{\bar{\nu}} \sum_{\bar{\mu}} C^{K_1/N_s}_{\bar{\nu}\bar{\mu}0} \left(N^{(s)}_{RUs} - N^{(s)}_{RLs} \right) \tag{21}$$

where the summation limits are as shown in Figure 12.



Always contributes to the summation
Contributes if the dihedral angle γ = 0

FIGURE 12 Subdivided "Effective Area"

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

$$\mathcal{B} = \mathcal{B}_S + \sum \sum C_{\bar{\nu}\bar{\mu}o} \left(N_{RUS} - N_{RLS} \right) \quad (22)$$

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.

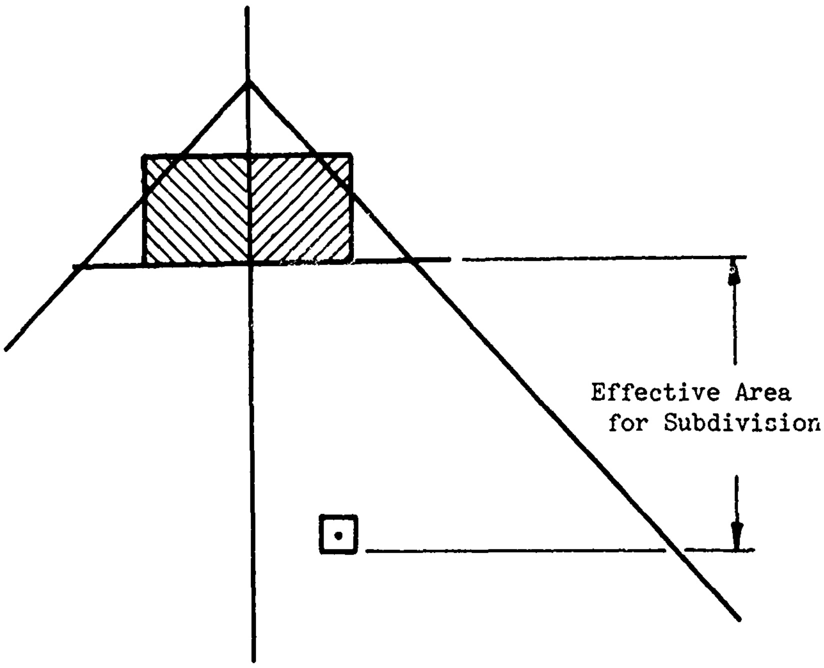


FIGURE 13. Unsubdivided Boxes Outside the "Effective Area"

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) $C_{\bar{\nu}\bar{\mu}0}$ array
SKERNL	Subdivided $C_{\bar{\nu}\bar{\mu}0}^{K_1/N_s}$ 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

$$\mathrm{WSUM} = \sum_{\text{rt. wing}} W_{\bar\nu\bar\mu\bar\lambda} * N_{RW}^{\nu\mu} \tag{23}$$

$$\mathrm{VSUM} = \sum_{\text{rt wing}} V_{\bar\nu\bar\mu\bar\lambda} * N_{RW}^{\nu\mu} \tag{24}$$

$$\mathrm{PHISUM} = \sum_{\text{rt wing}} C_{\bar\nu\bar\mu\bar\lambda} * N_{RW}^{\nu\mu} \tag{25}$$

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:

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

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

$$PHI_R = \mathrm{PHISUM}$$

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

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

$$SW_{\text{complete}} = SW_R + \left(1/b_1\right) * \left(\cos\Psi_W * \mathrm{VSUM} - \sin\Psi_W * \mathrm{WSUM}\right) \times \mathrm{SYM} \tag{29}$$

$$PHI_{complete} = PHI_R + PHISUM * SYM \qquad (30)$$

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

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

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.

FORTRAN 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 though 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 $\Delta\bar{\bar{\phi}}$ array is read into DELPHI and $\Delta\bar{\bar{\phi}}_{TE}$ 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:

$$\bar{\bar{L}}_j^{nm} = BXLIFT(IDC) = \frac{2}{\beta}\left[\Delta\bar{\bar{\phi}}_{j\,TE}^{nm} - \Delta\bar{\bar{\phi}}_{j\,LE}^{nm} + i\alpha^{nm} k_1 \Delta\bar{\bar{\phi}}_j^{nm}\right]\bar{\bar{Z}}_\tau^{nm} \qquad (32)$$

$$\Delta C_{p_j}^{nm} = DELCP(IDC) = \bar{\bar{L}}_j^{nm} / (\alpha^{nm} * b_1) \qquad (33)$$

$$\bar{\bar{Q}}_{ij}^{nm}\,\beta/2 = \left[\left(f_{TE}^{nm} * \Delta\bar{\bar{\phi}}_{TE}^{nm} - f_{LE}^{nm} * \Delta\bar{\bar{\phi}}_{LE}^{nm}\right) - b_1 \alpha^{nm} \frac{\partial f^{nm}}{\partial x} \Delta\bar{\bar{\phi}}^{nm} + i\alpha^{nm} k_1 f^{nm} \Delta\bar{\bar{\phi}}_j^{nm}\right] \bar{\bar{Z}}_\tau^{nm} \qquad (34)$$

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

$$AFROW(JVP) = \sum_m \sum_n Q_{ij}^{nm}\,\beta/2 \qquad (35)$$

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:

$$\bar{\bar{L}}_j^{m} = SLIFT(JCOL) = \sum_n \bar{\bar{L}}_j^{nm} \qquad (36)$$

$$\bar{L}_j = TLIFT = \sum_m \bar{\bar{L}}_j^{m} \qquad (37)$$

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

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

$$Q'_{ij} = -b_1/s^3\beta \ \text{Re}\left[\bar{\bar{Q}}_{ij}\right] \tag{39}$$

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

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,

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

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

and similarly for the box trailing edge.

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

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

$$\Delta\phi_{LE}^{nm} = \begin{cases} 0 & \text{for wing or spatial tail} \\ \Delta\bar{\bar{\phi}}_{\substack{TE\ of \\ wing}} * e^{\frac{-i\left(x^m_{LE\,tail} - X^m_{TE\,wing}\right)k_1}{b_1}} \end{cases} \tag{44}$$

(3) Box is cut by the planform trailing edge

$$f_{TE}^{nm} = f^{nm} + \left(x_{TE} - x_n\right) \cdot \frac{\partial f^{nm}}{\partial x} \qquad (45)$$

$\Delta\bar{\bar{\phi}}_{TE}$ 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 $\Delta\bar{\bar{\phi}}$ 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)
```

<u>Input</u>:

Uses labeled common blocks:

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

Uses the following files:

MODESC
IPNTRM
IVPSC
ITSLSC

<u>Output</u>:

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.

/CONTRL/ PREVEX, OMACH, TITLE(8), PRVGEOM, PRVMODE, DIHW, DIHT, DEFAULT

		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 / DIHT	.T. {Wing / Tail} dihedral is to be used computing influence on itself .F. The {wing / tail} is to be considered flat in computing influence on itself, but dihedral will be used in wing/tail calculations	.F.
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

		Default
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, XCENTR

/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	b_1/β, box width, = YWLE(NWLE)/MYBW	--
B1S B1BTAS	Subdivided box $\left\{\begin{array}{l}\text{length = B1/XSUBDV}\\ \text{width = B1BETA/XSUBDV}\end{array}\right\}$	-- --
WLAX TLAX	$\left\{\begin{array}{l}\text{Wing}\\ \text{Tail}\end{array}\right\}$ local axis location, in global X co-ordinate	0. 0.
WLAZ TLAZ	$\left\{\begin{array}{l}\text{Wing}\\ \text{Tail}\end{array}\right\}$ local axis location, in global Z co-ordinate	0. 0.
PSIW PSIT	$\left\{\begin{array}{l}\text{Wing}\\ \text{Tail}\end{array}\right\}$ dihedral angle, input in degrees but immediately changed to radians.	0. 0.
MXBW MXBT	Number of rows to aftmost portion of the $\left\{\begin{array}{l}\text{wing}\\ \text{tail}\end{array}\right\}$ measured in the n_c coordinate	-- --
MXBBW	Number of rows to aftmost wing diaphragm box, n_c coordinate	--
MYBW MYBT	Number of chords on the $\left\{\begin{array}{l}\text{wing}\\ \text{tail}\end{array}\right\}$, m_c coordinate	=NCHRDS --
MYBBW MYBBT	Number of $\left\{\begin{array}{l}\text{wing}\\ \text{tail}\end{array}\right\}$ chords, including tip diaphragm	-- --
MXBSW MXBST	Subdivided MXB $\left\{\begin{array}{l}\text{W}\\ \text{T}\end{array}\right\}$ count	-- --

		Default
MYBSW MYBST	Subdivided MYB $\left\{\begin{matrix} W \\ T \end{matrix}\right\}$ count	-- --
MYBBSW MYBBST	Subdivided MYBB $\left\{\begin{matrix} W \\ T \end{matrix}\right\}$ count	-- --
IXBW IXBT	Subdivided grid X-location of the first unsubdivided $\left\{\begin{matrix} \text{wing} \\ \text{tail} \end{matrix}\right\}$ box center	-- --
IXBST	Subdivided grid X-location of the first subdivided tail box	--
XCENTR	X_w location of the center of the first box on the wing	No default
CAPL	Non-dimensionalized vertical distance between centerlines of the wing and tail	0.

/KERN/ ERR, MXSKRN, IPKERN, NPLKRN, NSPATK, NRØWEA

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 $C_{\nu\bar{\mu}o}$ array.	1
NPLKRN	Size of the planar AIC array (number of rows)	--
NSPATK	Number of spatial AIC arrays necessary	0
NRØWEA	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, NOUTP, IOUFSP, MODESC, IVPSC, IGEOSC, IWTFSC, IAICSC

NT5	Card file (INPUT)	5
NT6	Print file (OUTPUT)	6
INTAPE	Binary input tape number, If 0 or 5 card input will be used	0

		Default
INFSP	Initial file spacing on the input tape	0
NPLAIC NSPAIC	Tape number for the $\left\{\begin{array}{l}\text{planar}\\ \text{spatial}\end{array}\right\}$ AIC arrays	0 0
NOUTP	Binary output tape number. If 0, none written	0
IOUFSP	Initial file spacing on tape NOUTP	0
MODESC, IVPSC IGEOSC, IWTFSC IAICSC	Internal scratch files	

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

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

		Default
PRSL	.T., Print section lifts, $\bar{\bar{L}}_j^m$	.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, NMS	File / Matrix spacing	0 0
LS, NMR	Not used	
ID	ID array for the matrix	
NID	Number of words in the ID array on tape	1
ITYPE	Matrix type - MIXED, COMPLEX	
LRS, LWS	Not used	
M, N	Matrix dimensions	-- --
PARM	Numerical parameters for the matrix	--
IRR	Error return	-0-

/MODES/ SYM, SYMT, MTYPEW, MTYPET

SYM	1, Symmetric modes -1, Antisymmetric modes 0, Left surface contribution will be ignored	1
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, Deflections at arbitrary locations will be read 3, Box center values will be read	2
MTYPET	1, Same as above for the tail 2, 3,	2

/ARRAYS/ KBXCDW, LBXCDW, LBOXC, KBXCDT, LBXCDT, KJALPH,
LJALPH, KALPHA, KKERNL, LKERNL, KPNTRM, LPNTRM, KDEFSL,
KELPHI, LMODES, KPNTSD, LPNTSD, KSDW, LSDW, KPNTDW,
LPNTDW, KDW, LDW, KTVP, LTVP

Locations and limits for arrays:

Variable	Array affected	Value
KBXCDW	IBOXW (LBXCDW, LBOXC)	Not used
LBXCDW		150
LBOXC		8
KBXCDT	IBOXT (LBXCDT, LBOXC)	Not used
LBXCDT		90
KJALPH	IJALPH (LJALPH)	Not used
LJALPH		200
KALPHA	ALPHA (LJALPH)	Not used
KKERNL	SKERNL (LKERNL), PKERNL	1
LKERNL		1640
KPNTRM	IPNTRM (2, LPNTRM)	Not used
LPNTRM		100
KDEFSL	DEFSL (2, LMODES)	Not used
KELPHI	DELPHI (LMODES), complex	Not used
LMODES		500
KPNTSD	IPNTSD (2, LPNTSD)	Not used
LPNTSD		50
KSDW	ENSUBD (2, LSDW)	Not used
LSDW		600
KPNTDW	IPNTDW (2, LPNTDW)	Not used
LPNTDW		100
KDW	ENRUS (LDW), ENRLS (LDW)	Not used
LDW		1275
KTVP	TVP (LTVP), TEXLOC (LTVP), FEXLOC (LTVP)	Not used
LTVP		250

Default

/RWBUFF/ BFCODE, IBFCNT, BUFF (3280)

BFCODE = Code word		8HBUFFSIZE
IBFCNT = Size of buffer		3280
BUFF = Buffer array for use by READMX and WRTEMX		

/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 internally to correspond to wing coordinates	0.

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

NWLE, NWTE	Number of wing {leading edge / trailing edge} definition points
NTLE, NTTE	Number of tail {leading edge / trailing edge} definition points
XWLE, YWLE	Wing leading edge definition points
XWTE, YWTE	Wing trailing edge definition points
XTLE, YTLE	Tail leading edge definition points
XTTE, YTTE	Tail trailing edge definition points

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"

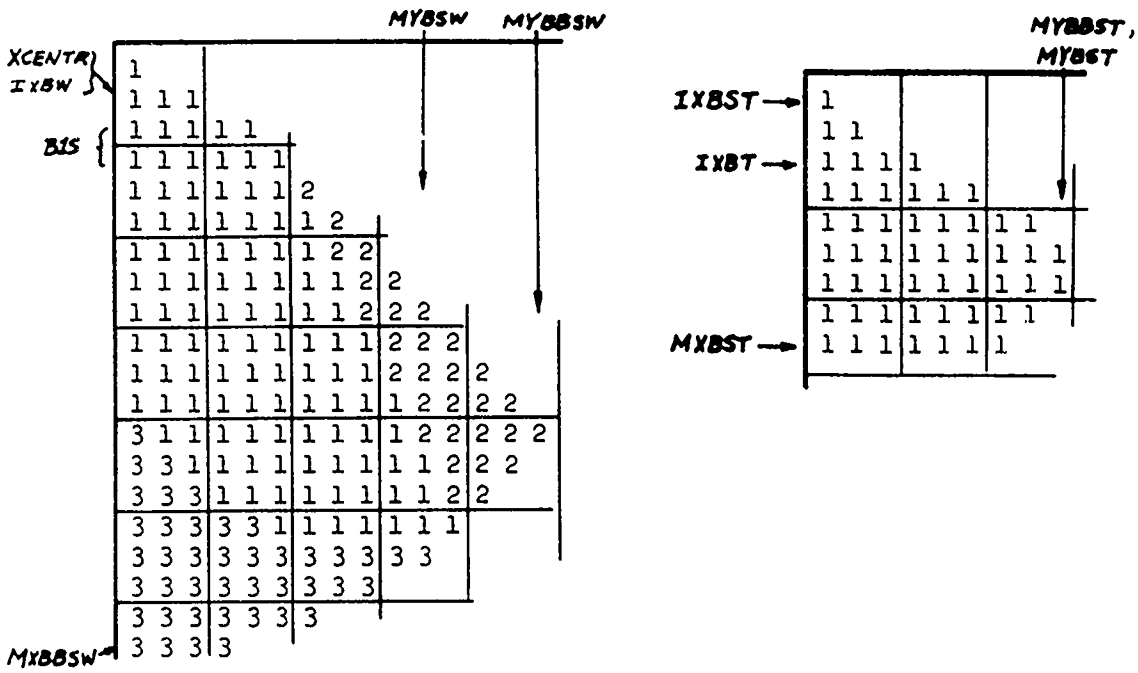
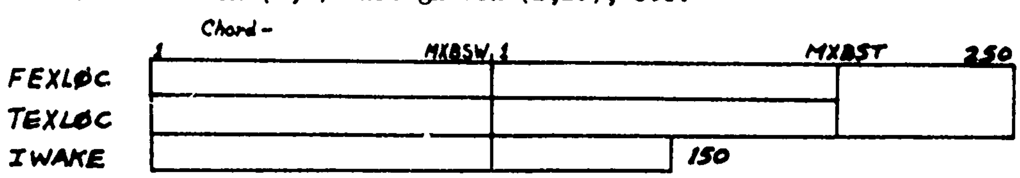


FIGURE 14. Box Code Arrays

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.

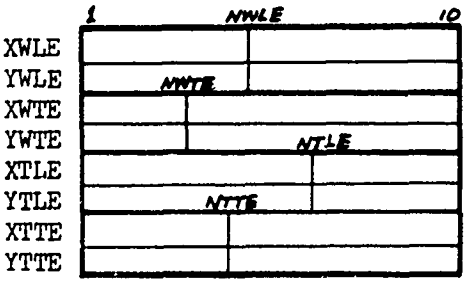


FEXLOC(I) = The location of the leading edge at chord I, normalized to B1S with 1.0 corresponding to the center of the 1st (subdivided) row.

TEXLOC(I) = Same for trailing edge.

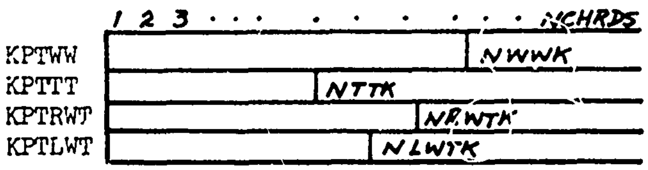
IWAKE(I) = Aftmost subdivided wing wake box needed by the tail.

FIGURE 15. Leading and Trailing Edge Arrays

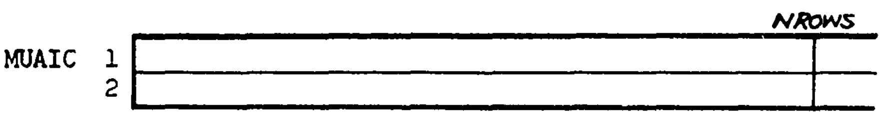


Planform edge definition, Non-dimensionalized and shifted after reading.

FIGURE 16. Planform Edge Definitions



For each AIC array needed:



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 IAICSC 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 $C_{\bar{\nu}\bar{\mu}\lambda}$, $W_{\bar{\nu}\bar{\mu}\lambda}$ and $V_{\bar{\nu}\bar{\mu}\lambda}$ arrays are stored in a one dimensional matrix. For planar AIC's the $W_{\bar{\nu}\bar{\mu}\lambda}$ and $V_{\bar{\nu}\bar{\mu}\lambda}$ are not computed and the $C_{\bar{\nu}\bar{\mu}\lambda}$ 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 unsubdivided array.

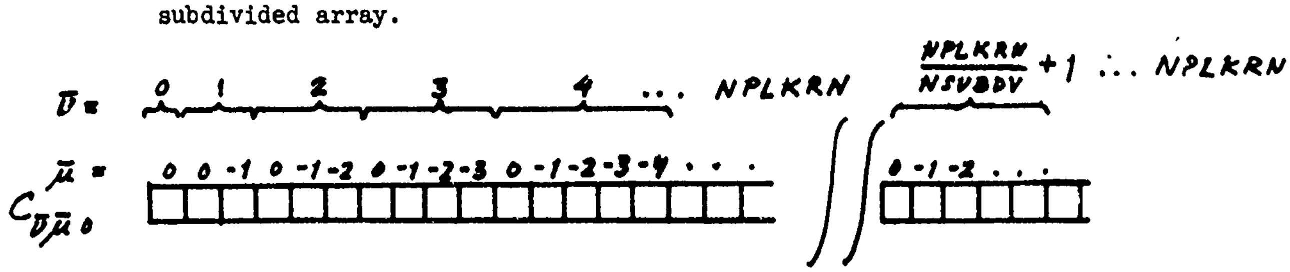


FIGURE 19. Planar AIC

For unsubdivided cases the array ends when $\bar{\nu}$ = NPLKRN. For subdivided cases when $\bar{\nu}$ = NPLKRN the array contains the subdivided AIC calculated at $k_1=k_1/\text{NSUBDV}$. $\bar{\nu}$ 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.

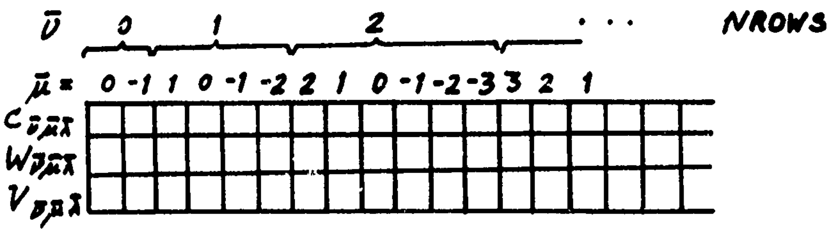


FIGURE 20. Spatial AIC's

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 - 16_{10} 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		
10		User parameters, array PARM
11		
to		
16		

Matrix Record - variable length (word 6 above)

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

b. Geometry Scratch File IGEOSC

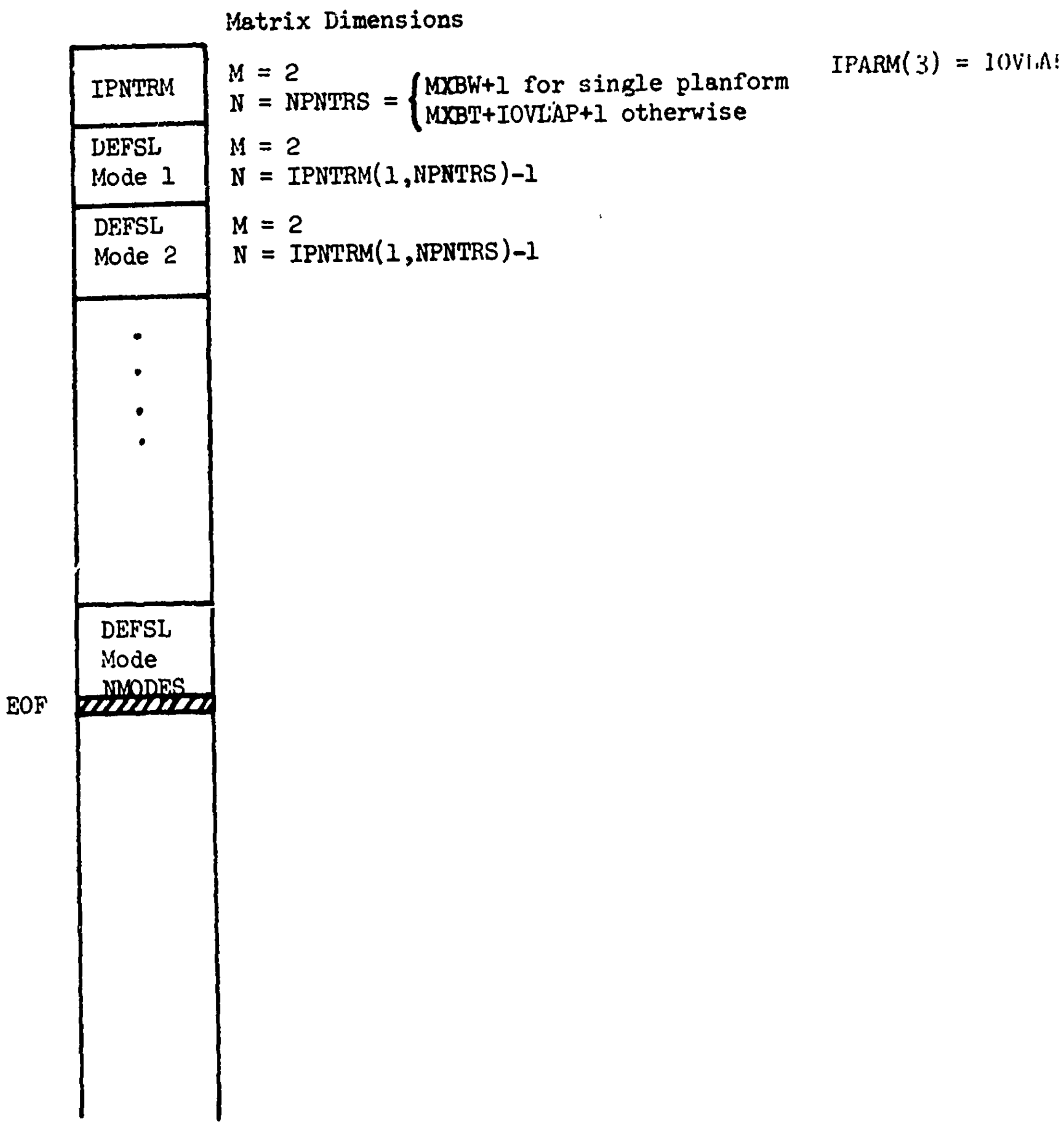
This file is generated in the geometry processor and contains all large geometry arrays. The space after the two geometry files is used for temporary scratch during mode shape processing.

Notes	Record	Matrix Dimensions	Parameter Array
	IBOXW	M = MXBBW*NSUBDV N = (MYBBSW-1)/NBWRD+1	PARM(1) = 0. PARM(2) = XMACH
Present only if NSURF=2 & COPLAN=.F.	IBOXT	M = MXBBST-IXBST+1 N = (MYBBST-1)/NBWRD+1	
	FEXLOC	M = 1 N = MYBSW+MYBST	
	TEXLOC	M = 1 N = MYBSW+MYBST	
	ALPHA	M = 1 N = NAL	IPARM(3) = NALPHW
	IJALPH	M = 1 N = NAL	
Present only if NSPATK 0	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			
NSPATK matrices	MUAIC	M = 2 N = NROWS	PARM(4) = YBAR PARM(5) = EL IPARM(6) = 0; C,W,V needed 1; W,V needed 2; V needed
	MUAIC	M = 2 N = NROWS	
	MUAIC	M = 2 N = NROWS	
EOF			

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

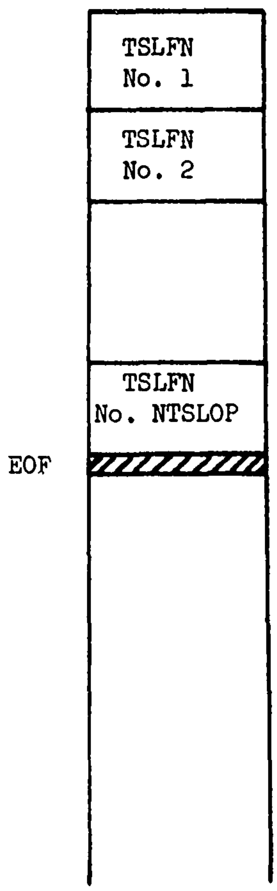
c. Modes Scratch File MODESC

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



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 $\frac{\partial z_\tau}{\partial x} = 0$.



Matrix Dimensions

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

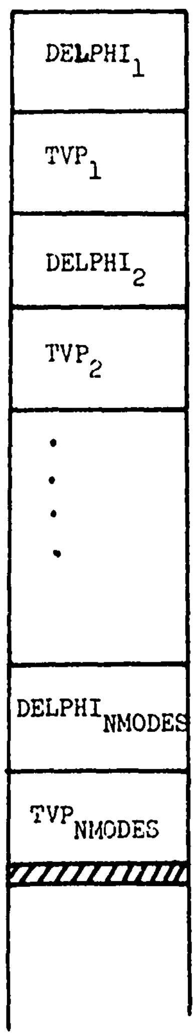
e. Spatial AIC Scratch File IAICSC

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 PARM(4) = $YBAR_1$ PARM(5) = EL_1
$C_{\bar{\nu}\bar{\mu}\lambda}$	M = 2 N = $(NROWS_1+1)(NROWS_1/2)$	
$W_{\bar{\nu}\bar{\mu}\lambda}$		
$V_{\bar{\nu}\bar{\mu}\lambda}$		
$MUAIC_2$	M = 2 N = $NROWS_2$	PARM(4) = $YBAR_2$ PARM(5) = EL_2
$C_{\bar{\nu}\bar{\mu}\lambda}$	M = 2 N = $(NROWS_2+1)(NROWS_2/2)$	
$W_{\bar{\nu}\bar{\mu}\lambda}$		
$V_{\bar{\nu}\bar{\mu}\lambda}$		
$MUAIC_{NSPATK}$		
$C_{\bar{\nu}\bar{\mu}\lambda}$		
$W_{\bar{\nu}\bar{\mu}\lambda}$		
$V_{\bar{\nu}\bar{\mu}\lambda}$		
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 $\Delta\phi_{TE}$ arrays for each mode.



Matrix Dimensions

M = 2
N = IPNTRM(1,NPNTRS)-1

M = 2
N = $\begin{cases} \text{MYBSW if wing only} \\ \text{MYBSW+MYBST otherwise} \end{cases}$

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 airforces matrices on for flutter or dynamic loads analyses. It is written optionally in the forces section of the program.

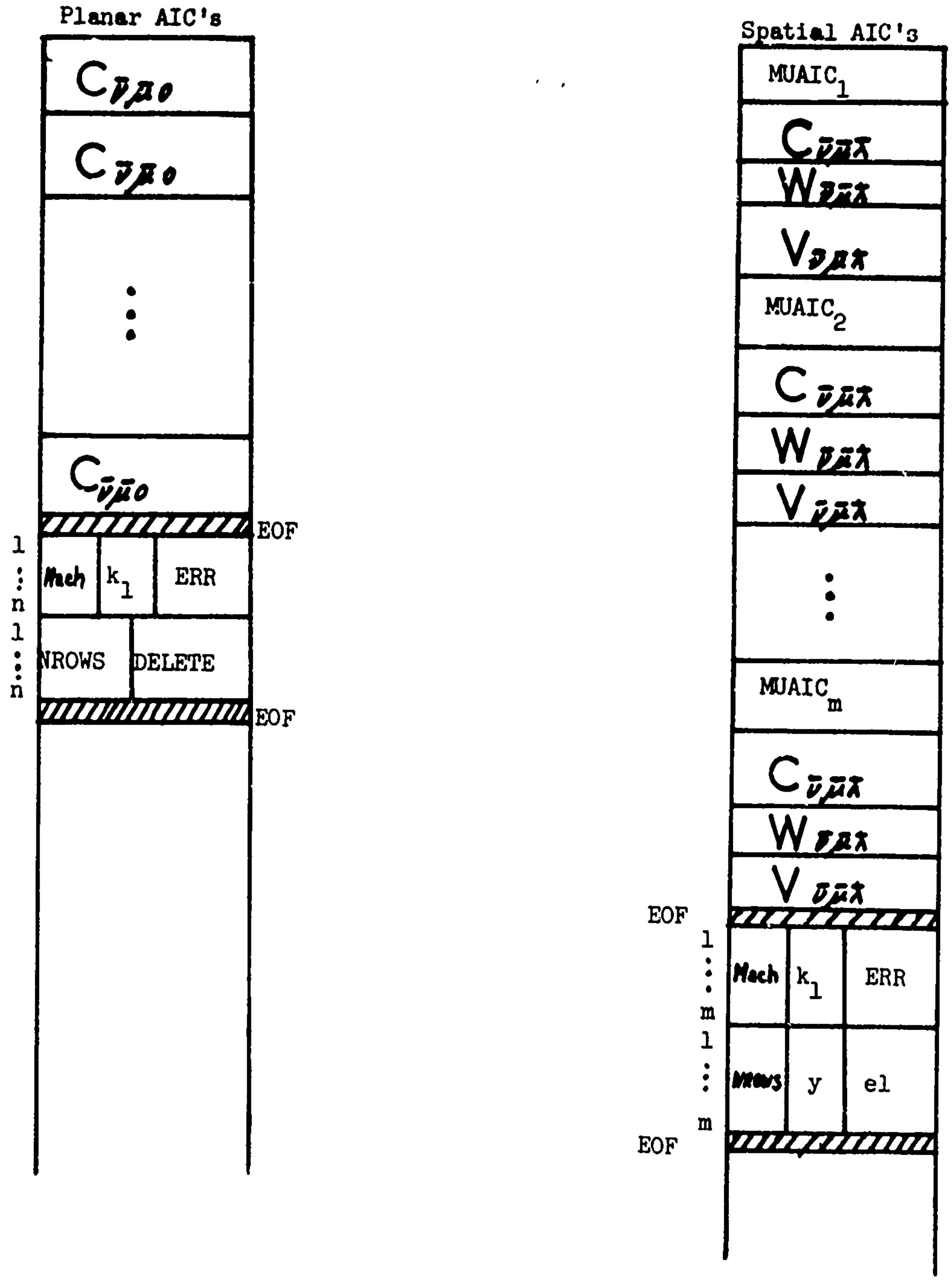


FIGURE 21. Tape Storage of AIC Arrays

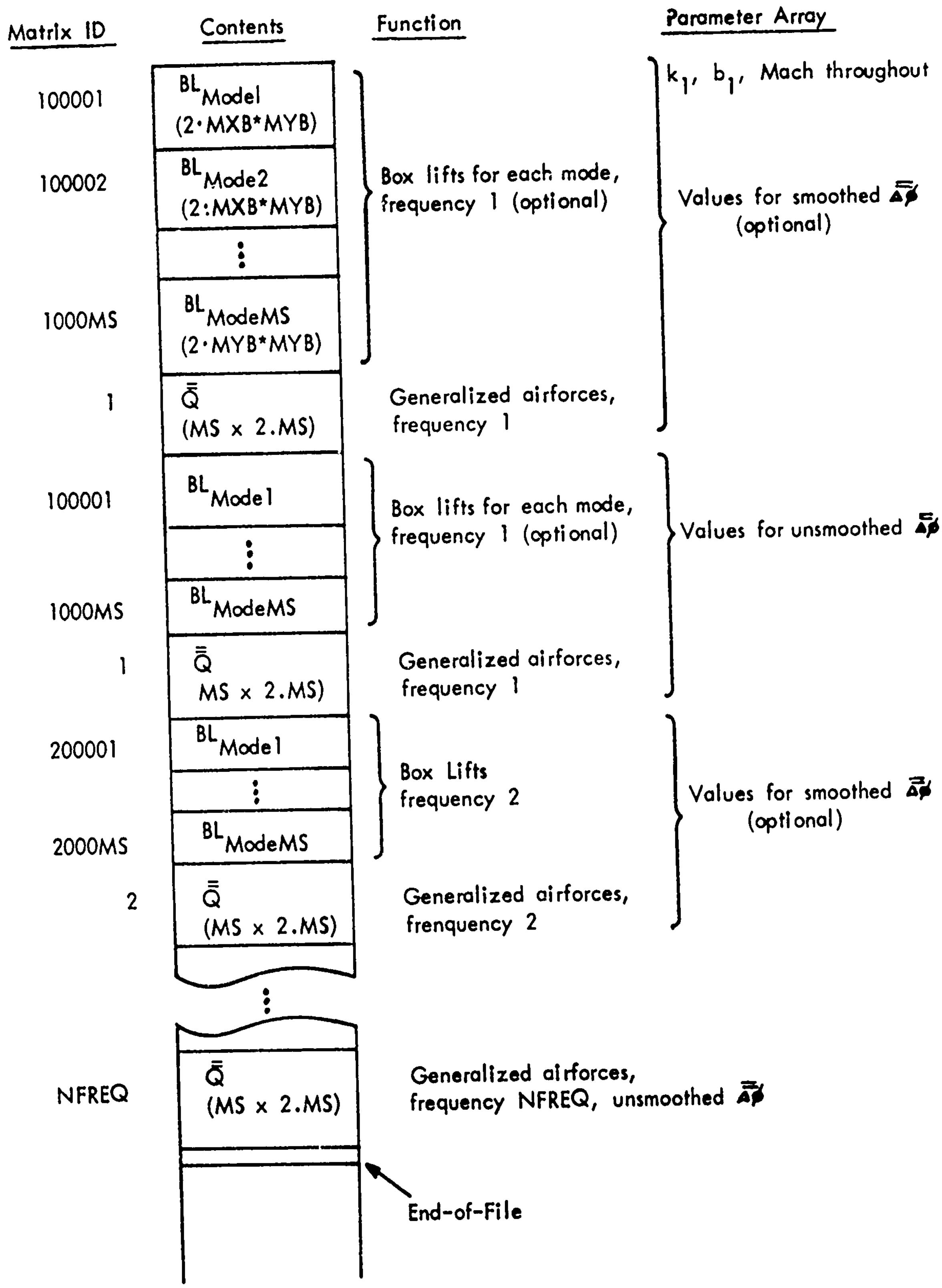


FIGURE 22. TAPE MAP OF FORCES OUTPUT TAPE

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.

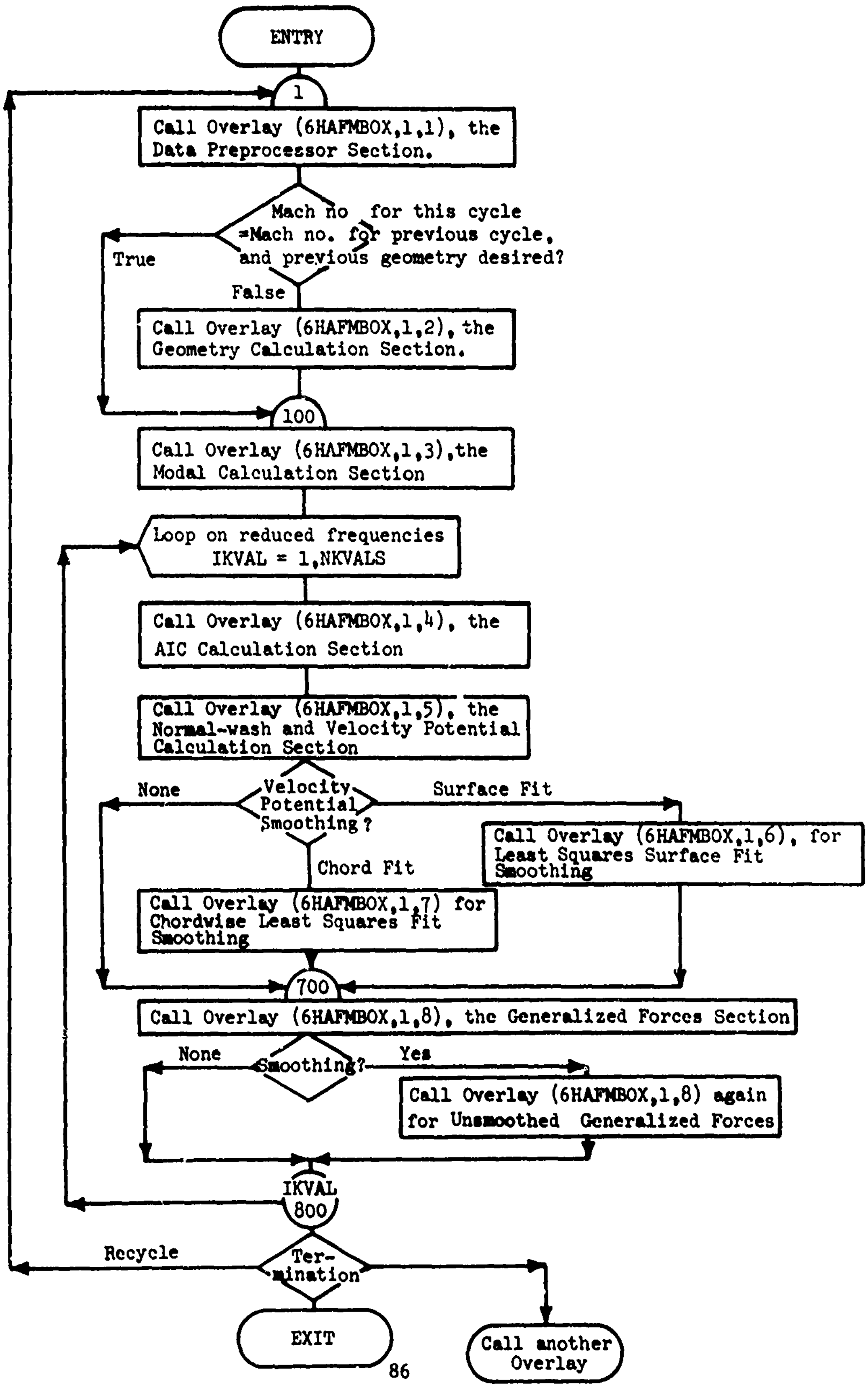
c. Debugging

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.

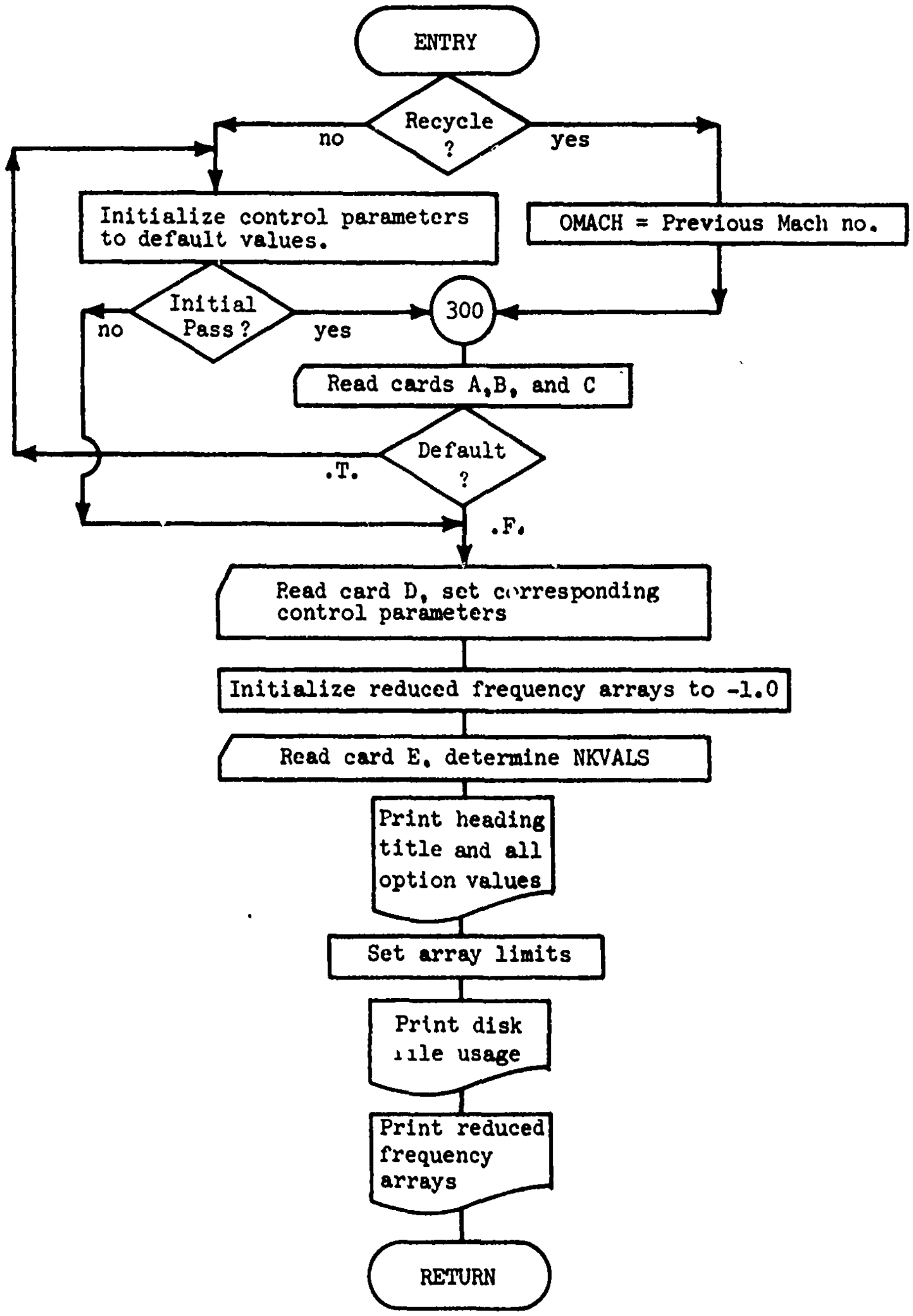
In the event the program fails "hard" (mode error, time limit, etc.), standard use of listings, load maps and dumps will usually pinpoint the cause.

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.

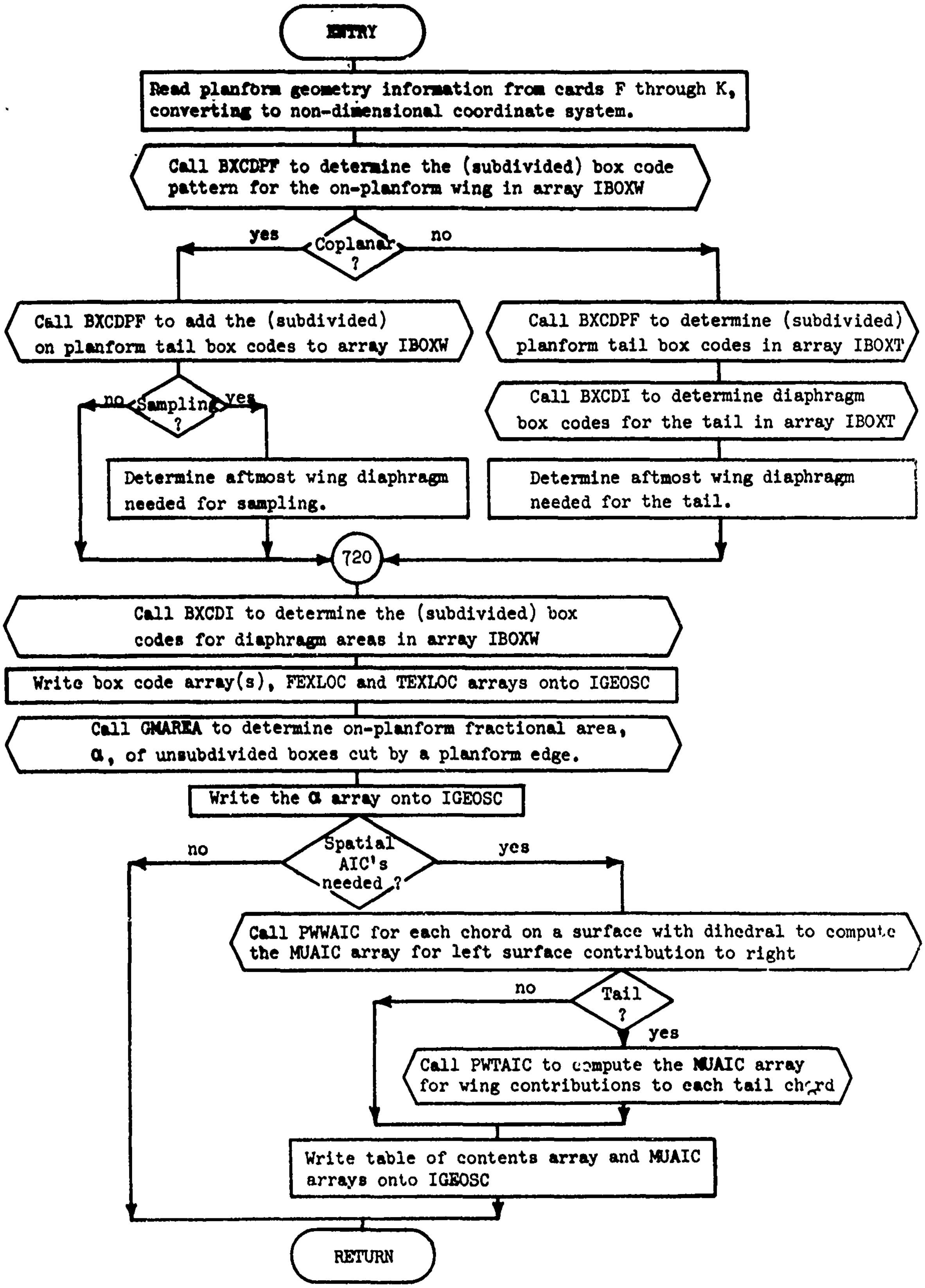
Program CONTROL — Primary level overlay which controls the program flow



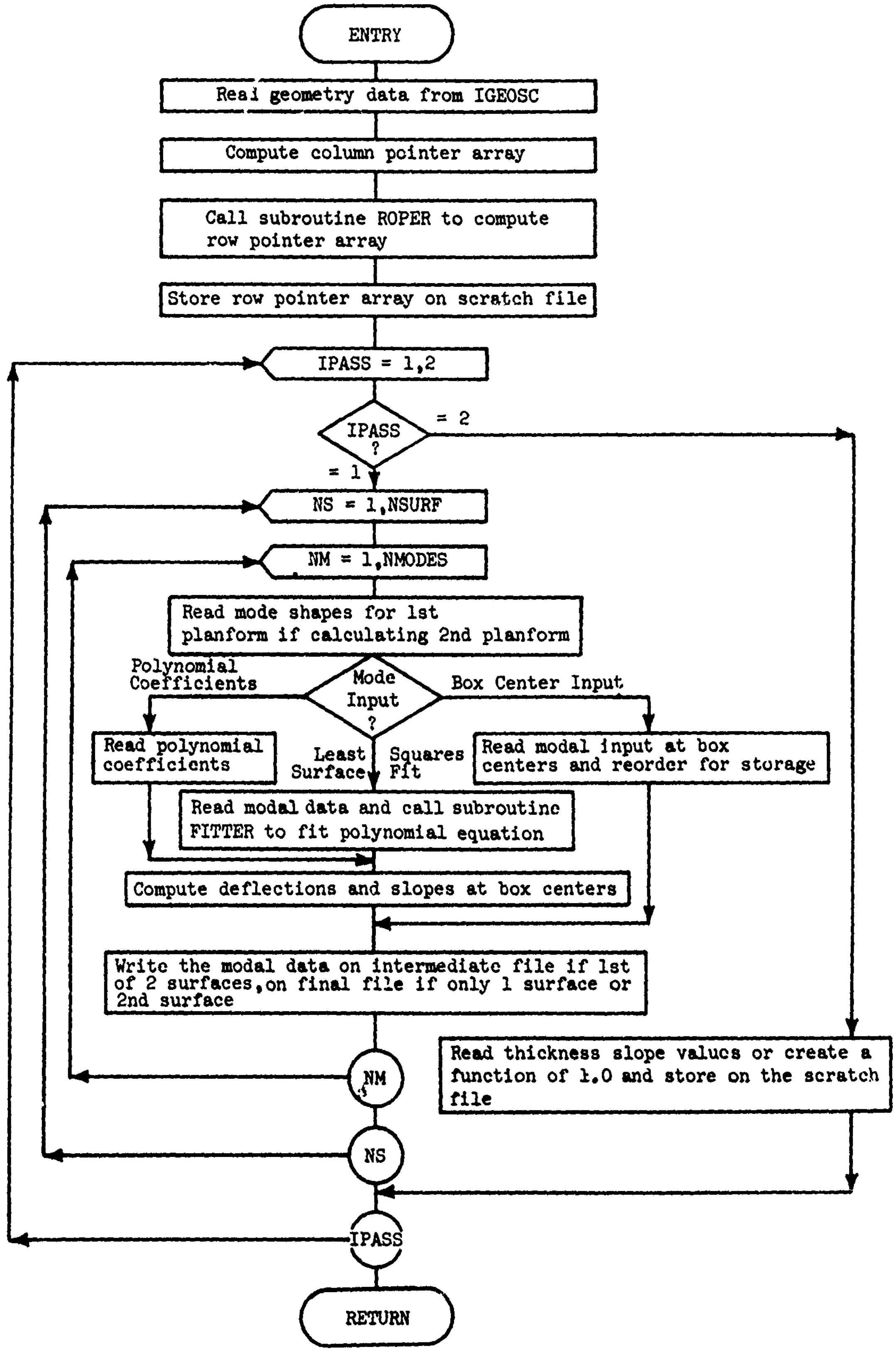
Program DATAPP — Secondary overlay which initiates control parameters as a function of defaults or card data.



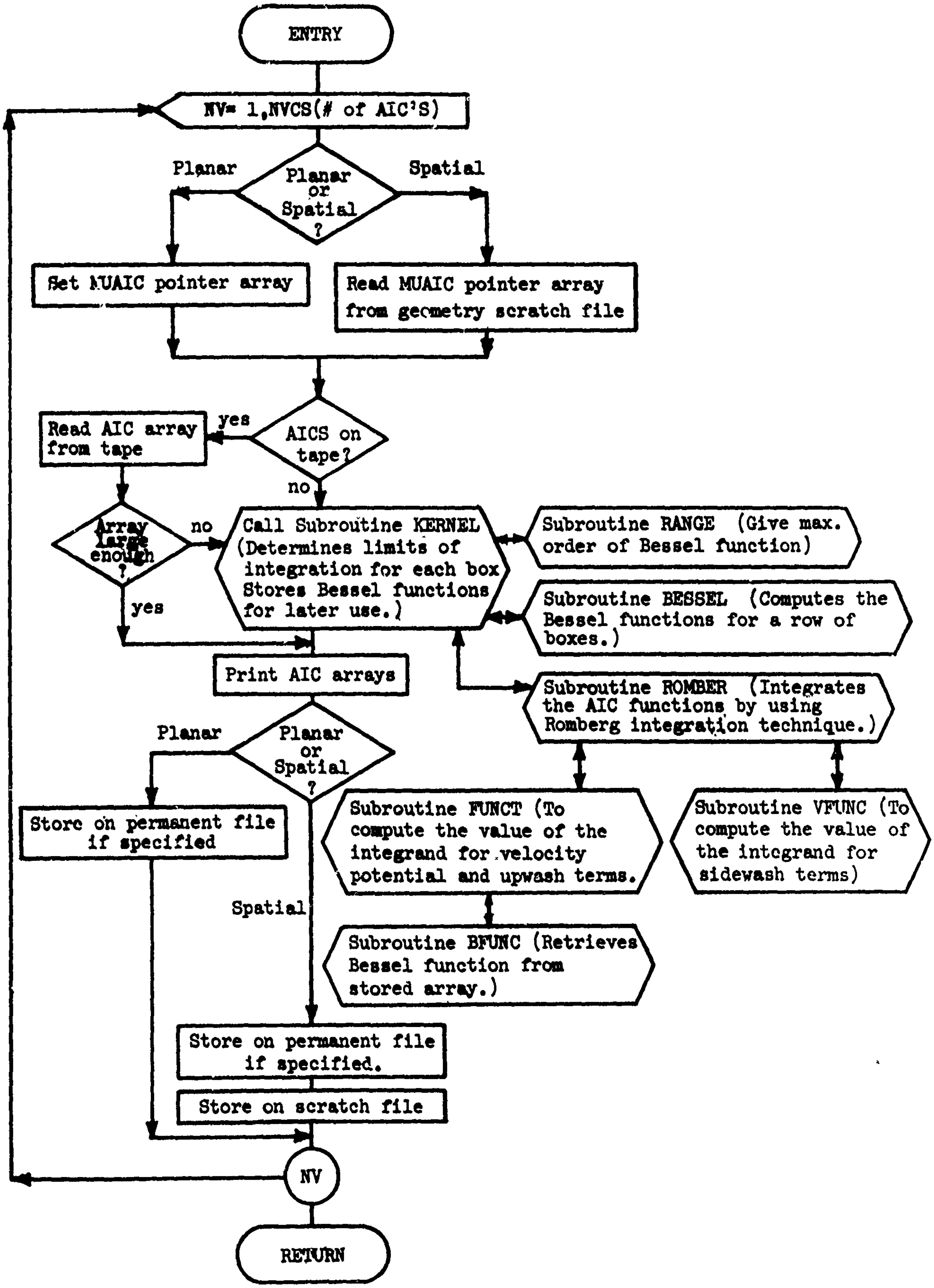
Program GEOMBK — Secondary level overlay which computes planform geometry.



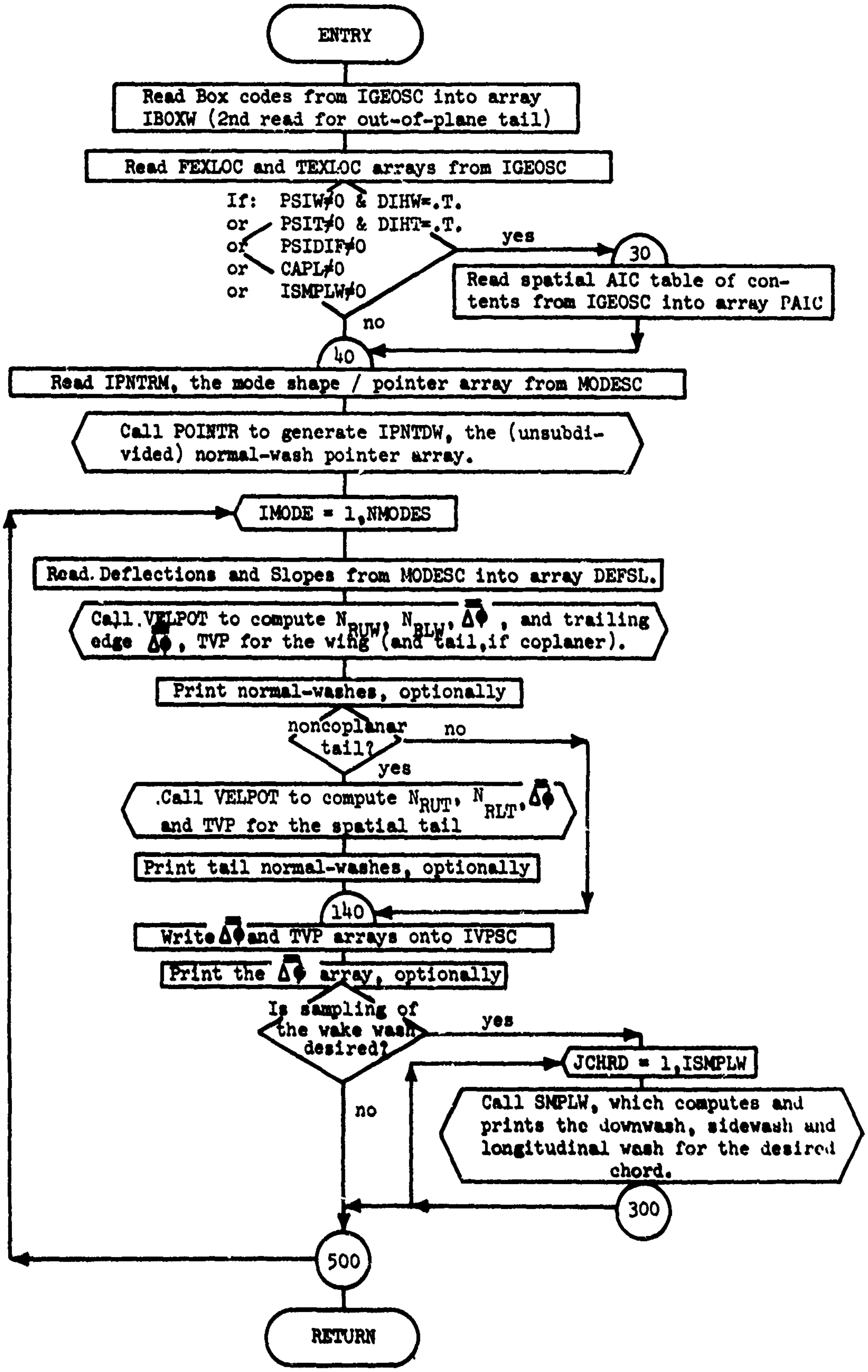
Program MODES — Secondary level overlay which reads mode shapes and thickness slope functions



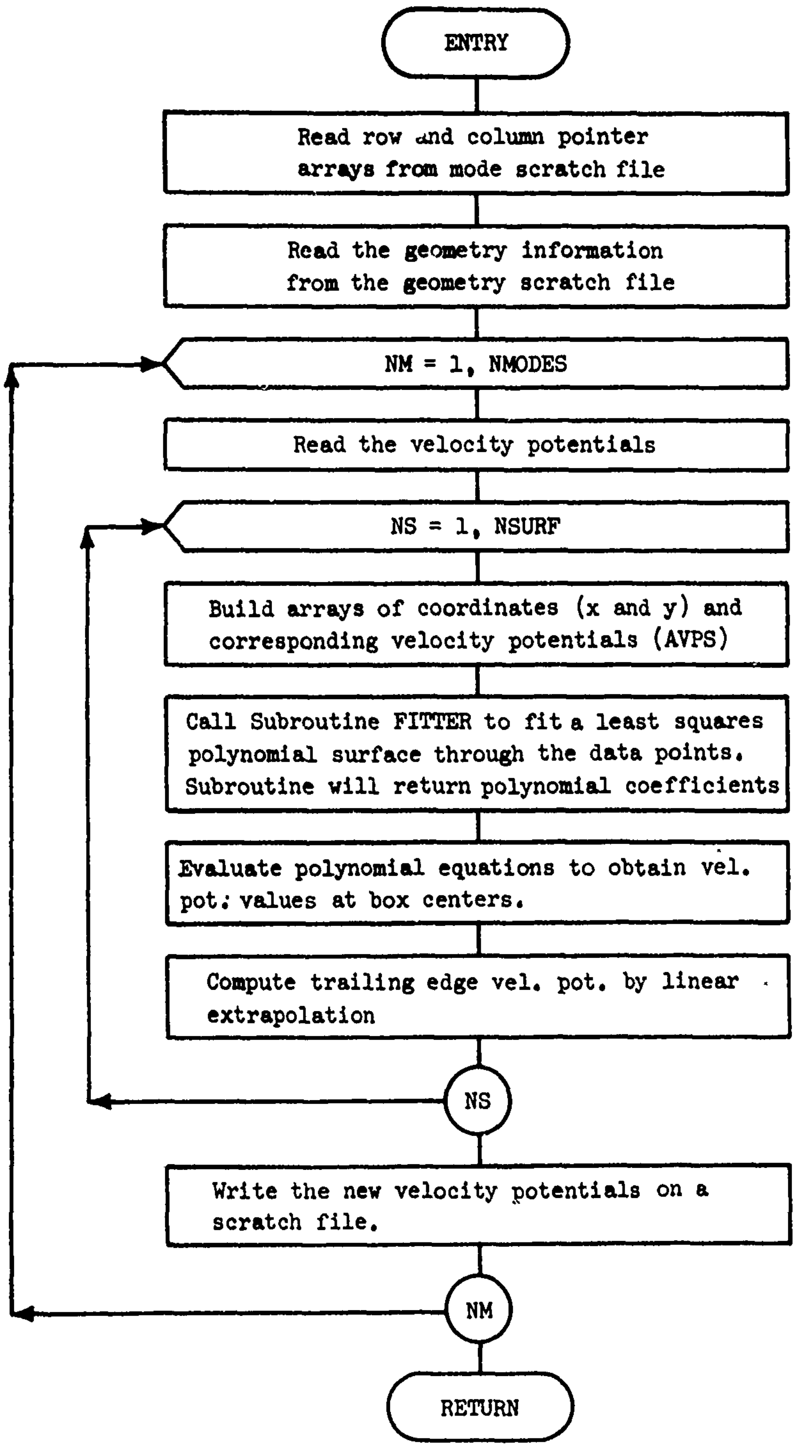
Program VICMAIN — Secondary level overlay which calculates all Aerodynamic Influence Coefficients



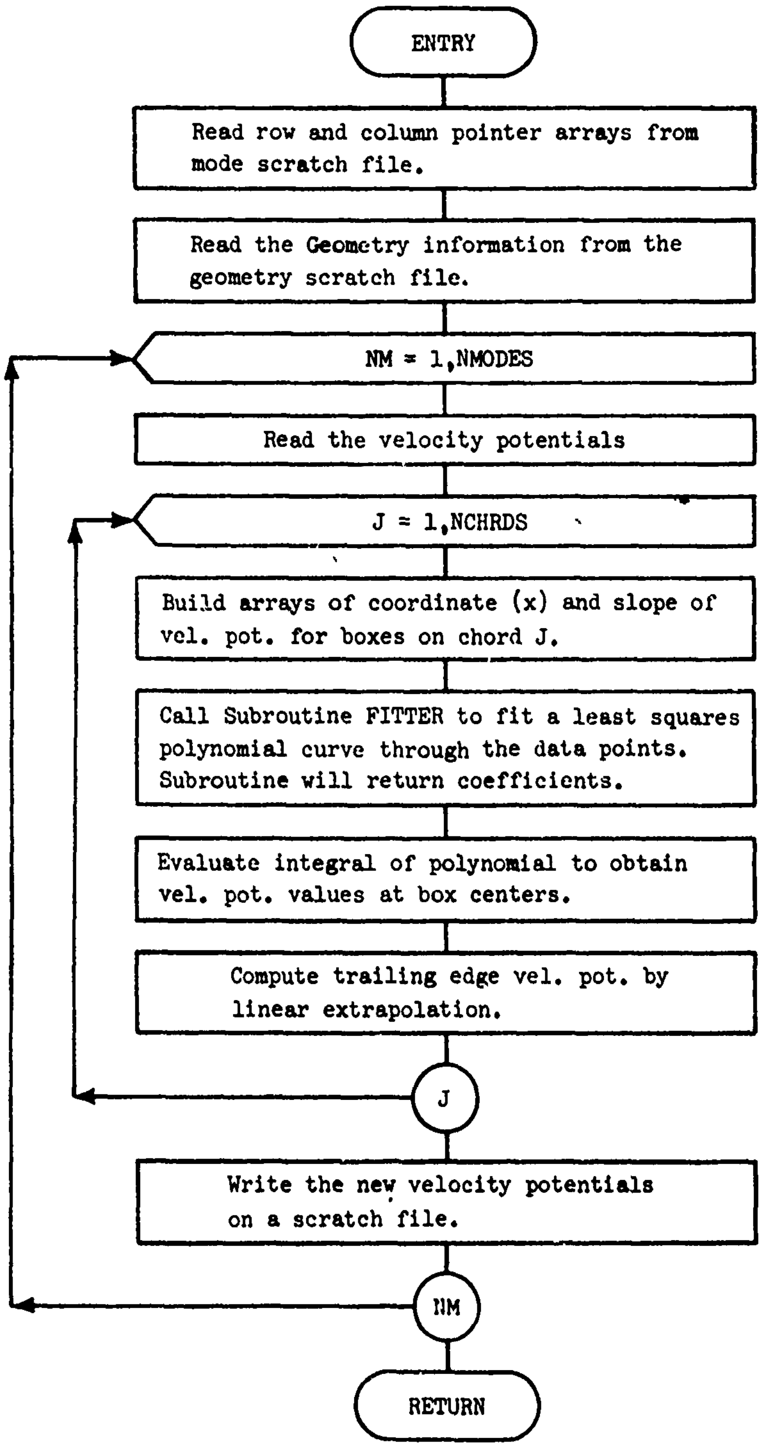
Program NWVLPT   Secondary overlay which calculates normal washes and velocity potentials.



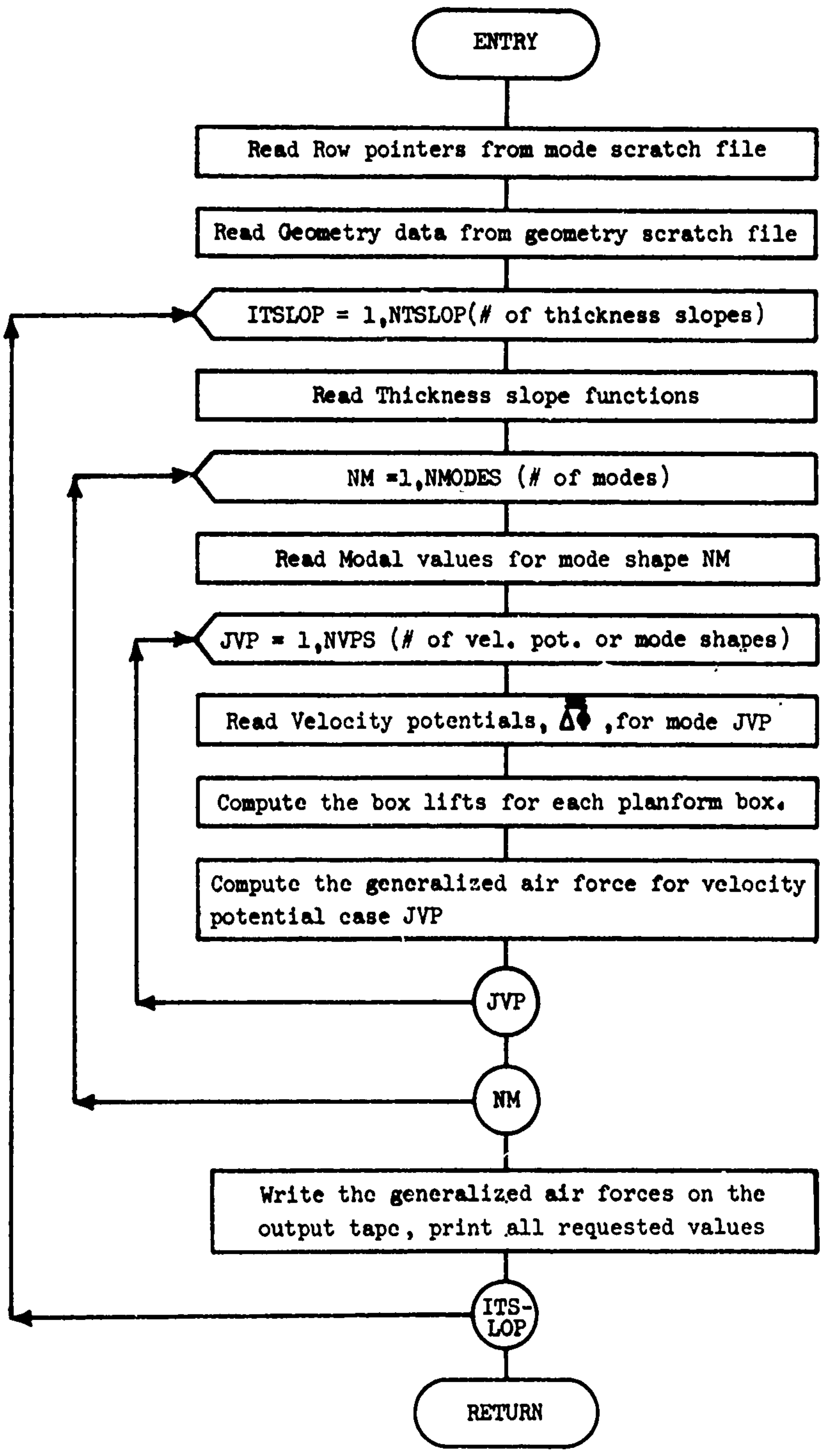
Program SMTH   Secondary overlay which smooths velocity potentials by fitting a least squares surface through them



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.

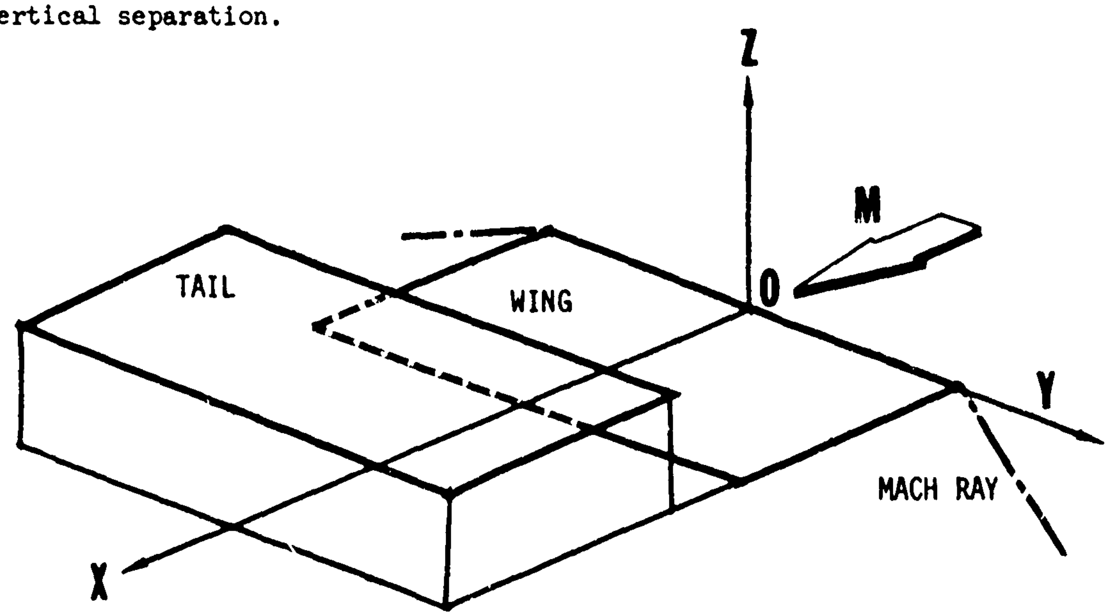


FIGURE 23 SAMPLE PROBLEM CONFIGURATION

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

```
AFMBOX                  1           0
SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
 $CARDB XMACH=1.20              $
 $CARDC SYM=1.0,MTYPEW=1,MTYPET=1,NSURF=2,WTGNAF=.T.,WTDL=.T.,
 PRGNAC=.T., PRDCP=.T.,
      PLYWOOD = .T.,
    CRDFIT=.T.,NDEG=4,
  PRGNAF=.T.,PRDL=.T.,PRSL=.T.,PRPAIC=.T.,PRSAIC=.T.,PRCOEF=.T.,
       PRBOX=.T.,PRVP=.T.,EXAIC=.F., PRNW=.T.,    PRMODS=.T.  $
 $CARDD         $
 $CARDE XKS(1) =       .50              $
 $CARDF TLAX=1.20, TLAZ=.40  $
 $CARDG NCHRDS=10, XEDGE=0.00           $
    2    2    2    2                                                      CARD H
0.          0.          0.          1.0                                   CARD I
1.0         0.0         1.0         1.0                                   CARD J
0.          0.          0.          1.0                                   CARD K
1.0         0.0         1.0         1.0                                   CARD L
 $CARDH NMODES=2 $
    2
1.          0.          0.          0.          0.          0.            WING-1
    2
0.          1.          0.          0.          0.          0.            WING-2
    1
0.          0.          0.                                                TAIL-1
    1
0.          0.          0.                                                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

                    THE FOLLOWING OPTIONS ARE REQUESTED -

                         MACH NUMBER = 1.200000
                         SYMMETRIC ANALYSIS
                         PLYWOOD OPTION IS USED. (PLANFORM BOUNDARY DETERMINED BY BOX PATTERN.)
                         BASIC (UNSUBDIVIDED) ANALYSIS WILL BE USED
                         ANALYSIS FOR 2 PLANFORMS
                         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 BOX PATTERN
                         PRINT MODE SHAPE POLYNOMIAL COEFFICIENTS, IF AVAILABLE
                         PRINT MODE SHAPES USED
                         PRINT THE PLANAR AIC ARRAYS USED
                         PRINT THE SPATIAL AIC ARRAYS USED
                         PRINT NORMAL WASHES
                         PRINT THE VELOCITY POTENTIALS
                         PRINT THE BOX 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,

                    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  - - - -

CARDF -LOCAL AXES DEFINITION-    X-LOCATION    Z-LOCATION    DIHEDRAL ANGLE (PSI)
                         WING      0.000         0.000          0.00 DEGREES
                         TAIL      1.200          .400          0.00 DEGREES

CARDG -BOX PATTERN DEFINITION-    NCHRDS          XCENTR     OR    XEDGE
                                    10          *0000.0000         0.0000

CARDH -PLANFORM DEFINITION POINT COUNTS-     LEADING EDGE    TRAILING EDGE
                                      WING        2                2
                                      TAIL        2                2

CARDI TO CARDL -PLANFORM DEFINITIONS-       X          Y      (LOCAL AXES)
                         WING L.E.        0.000      0.000
                                          0.000      1.000
```

WING T.E.	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/DETA (WIDTH) = 1.00000000E-01

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

 19  1 1 1 1 1 1 1 1 1 1
 20  1 1 1 1 1 1 1 1 1 1 2
 21  1 1 1 1 1 1 1 1 1 1 2 2
 22  1 1 1 1 1 1 1 1 1 1 2 2 2
 23  1 1 1 1 1 1 1 1 1 1 2 2 2 2
 24  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2
 25  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2
 26  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2
 27  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2
 28  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2
 29  1 1 1 1 1 1 1 1 1 1 2 2 2 2
 30  1 1 1 1 1 1 1 1 1 1 2 2 2
 31  1 1 1 1 1 1 1 1 1 1 2 2
 32  1 1 1 1 1 1 1 1 1 1 2
 33  1 1 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 1 1 1
 2  1 1 1 1 1 1 1 1 1 1 2
 3  1 1 1 1 1 1 1 1 1 1 2 2
 4  1 1 1 1 1 1 1 1 1 1 2 2 2
 5  1 1 1 1 1 1 1 1 1 1 2 2 2 2
 6  1 _ 1 1 1 1 1 1 1 1 2 2 2 2 2
 7  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2
 8  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2
 9  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
10  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2
11  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
12  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
13  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
14  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2
15  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
16  3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
17  3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
18  3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
19  3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2
20  3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2
21  3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2
22  3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2
23  3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2
24  3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2
25  3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2
26  3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2
27  3 3 3 3 3 3 3 3 3 3 2 2 2 2 2
28  3 3 3 3 3 3 3 3 3 3 2 2 2 2
29  3 3 3 3 3 3 3 3 3 3 2 2
```

ENTERING PROGRAM MODES   CURRENT ELAPSED TIME IS   CP =   1.894, FF = 40.123

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

MODE SHAPE NUMBER 2
MACH NUMBER = 1.200000

MODAL POLYNOMIAL COEFFICIENTS FROM CARD INPUT

CONSTANT
0.

X **1Y **0	X **0Y **1
1.0000E+00	0.

X **2Y **0	X **1Y **1	X **0Y **2
0.	0.	0.

MODAL POLYNOMIAL COEFFICIENTS FROM CARD INPUT

CONSTANT
0.

X **1Y **0	X **0Y **1
0.	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.[illegible]22	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	[illegible].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	[illegible].638	5.638	5.[illegible]38	[illegible].638	5.638	5.638	5.638	5.638

10	6.302	6.302	6.302	6.302	6.302	6.302	6.302	6.302	6.302	6.302
11	6.965	6.965	6.965	6.965	6.965	6.965	6.965	6.965	6.965	6.965
12	7.628	7.628	7.628	7.628	7.628	7.628	7.628	7.628	7.628	7.628
13	8.292	8.292	8.292	8.292	8.292	8.292	8.292	8.292	8.292	8.292
14	8.955	8.955	8.955	8.955	8.955	8.955	8.955	8.955	8.955	8.955
15	9.618	9.618	9.618	9.618	9.618	9.618	9.618	9.618	9.618	9.618
19	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
20	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
21	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
22	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
24	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
26	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
27	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
28	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
29	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
30	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
31	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
33	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

SLOPES X 1.0E 0

	1	2	3	4	5	6	7	8	9	10
1	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
5	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
6	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
7	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
8	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
9	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
11	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
12	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
13	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
14	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
15	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
19	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

20	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
21	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
22	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
24	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
26	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
27	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
28	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
29	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
30	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
31	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
33	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

ENTERING PROGRAM AIC   CURRENT ELAPSED TIME IS CP = 2.555, PP = 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 VELOCITY POTENTIAL COEFFICIENT		W UPWASH COEFFICIENT		V SIDEWASH COEFFICIENT	
0	0	0.	0.	0.	0.	0.	0.
1	1	0.	0.	0.	0.	0.	0.
1	0	0.	0.	0.	0.	0.	0.
1	-1	0.	0.	0.	0.	0.	0.
2	2	0.	0.	0.	0.	0.	0.
2	1	0.	0.	0.	0.	0.	0.
2	0	0.	0.	0.	0.	0.	0.
2	-1	0.	0.	0.	0.	0.	0.
2	-2	0.	0.	0.	0.	0.	0.
3	3	0.	0.	0.	0.	0.	0.
3	2	0.	0.	0.	0.	0.	0.
3	1	0.	0.	0.	0.	0.	0.
3	0	0.	0.	0.	0.	0.	0.
3	-1	0.	0.	0.	0.	0.	0.
3	-2	0.	0.	0.	0.	0.	0.
3	-3	0.	0.	0.	0.	0.	0.
4	4	0.	0.	0.	0.	0.	0.
4	3	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	1	-1.17760029E-01	5.85388468E-02	1.72703458E-01	-7.56330976E-02	4.39228951E-02	-1.94380883E-02
4	0	-1.39653064E-01	6.82985240E-02	1.59648216E-01	-6.50348287E-02	0.	0.
4	-1	-1.17760029E-01	5.85388468E-02	1.72703458E-01	-7.56330976E-02	-4.39228951E-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	-3	0.	0.	0.	0.	0.	0.
4	-4	0.	0.	0.	0.	0.	0.
5	5	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
5	3	-1.16088951E-01	7.25678775E-02	1.10641214E-01	-6.02286065E-02	8.38256078E-02	-4.59949061E-02
5	2	-1.45526451E-01	8.55703207E-02	-9.83107500E-02	5.93665069E-02	-4.25224207E-02	2.62689346E-02
5	1	-9.96538896E-02	5.92204335E-02	-6.88337551E-02	3.97750794E-02	-1.80453789E-02	1.04152589E-02
5	0	-9.18644239E-02	5.46812863E-02	-5.23885439E-02	3.04815076E-02	0.	0.
5	-1	-9.96538896E-02	5.92204335E-02	-6.88337551E-02	3.97750794E-02	1.80453789E-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	-6.02286065E-02	-8.38256078E-02	4.59949061E-02
5	-4	-1.26580111E-02	8.45618922E-03	7.57086924E-02	-4.99682786E-02	-6.96685870E-02	4.60082949E-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	-5.45262136E-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	-9.31703898E-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	-9.20008406E-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	-3.83083684E-02	3.61961147E-02	-6.86176794E-03	6.43842195E-03	3.45857670E-03	-3.24490770E-03
7	-3	-4.33217879E-02	4.08912504E-02	-9.20008406E-03	8.60658871E-03	6.97408521E-03	-6.52337914E-03
7	-4	-5.41942395E-02	5.10205307E-02	-1.66531945E-02	1.54504087E-02	1.69471710E-02	-1.57190079E-02
7	-5	-9.31703898E-02	8.69860745E-02	-3.18547641E-02	3.11225707E-02	3.78738988E-02	-3.73390401E-02
7	-6	-5.56430474E-02	5.54322101E-02	6.75641236E-02	-6.43952032E-02	-1.02001591E-01	9.75864803E-02
7	-7	0.	0.	0.	0.	0.	0.
8	8	0.	0.	0.	0.	0.	0.
8	7	-6.15398979E-02	7.51517780E-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	-2.96375479E-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	-3.22017048E-02	3.78151778E-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	1.16230536E-02	-1.34631701E-02
8	-6	-6.73784527E-02	7.80656965E-02	-3.25854394E-02	3.74499354E-02	4.95508943E-02	-5.69967196E-02
8	-7	-6.15398979E-02	7.51517780E-02	5.40298073E-02	-6.23899846E-02	-9.63065278E-02	1.117347[illegible]7E-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	-4.70533895E-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	-4.70533895E-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	7.62974673E-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	-5.38313642E-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	5.38313642E-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	-2.28652507E-02	-2.62746278E-02	5.50987357E-02
11	11	-4.18639975E-03	1.20663134E-02	8.56562552E-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	-6.08235328E-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	-1.23991352E-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	-1.06807888E-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	1.06807888E-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	-1.23991352E-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	-6.08235328E-03	1.47178448E-02	1.38752164E-02	-3.35630000E-02
11	-10	-3.78975536E-02	9.62892728E-02	3.27966726E-03	-4.93045524E-03	-9.24313571E-03	1.47902829E-02
11	-11	-4.18639975E-03	1.20663134E-02	8.56562552E-03	-2.40799525E-02	-2.28776961E-02	6.43417919E-02
12	12	-3.72776720E-03	1.60634916E-02	5.82061141E-03	-2.40293415E-02	1.70533830E-02	-7.04581208E-02
12	11	-2.62917539E-02	9.55891131E-02	1.02580352E-03	4.09961357E-05	3.50152909E-03	-2.07692808E-03
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	4.74458787E-03	-3.02684968E-03	1.07305559E-02
12	8	-8.05661541E-03	2.89904861E-02	-8.46758430E-04	3.02522663E-03	-1.69955143E-03	6.07168542E-03
12	7	-6.68576228E-03	2.40828451E-02	-6.30077270E-04	2.25950416E-03	-1.10566569E-03	3.96486125E-03
12	6	-5.76180714E-03	2.07657237E-02	-5.11573400E-04	1.83850727E-03	-7.69132814E-04	2.76406283E-03
12	5	-5.10897888E-03	1.84179437E-02	-4.39299702E-04	1.58092128E-03	-5.50263143E-04	1.98021174E-03
12	4	-4.64105922E-03	1.67332808E-02	-3.92716140E-04	1.41454784E-03	-3.93483559E-04	1.41729147E-03
12	3	-4.31064695E-03	1.55428127E-02	-3.62239121E-04	1.30554646E-03	-2.72196355E-04	9.81011826E-04
12	2	-4.09014224E-03	1.47479566E-02	-3.42948235E-04	1.23648864E-03	-1.71797790E-04	6.19405321E-04
12	1	-3.96345727E-03	1.42911603E-02	-3.32229523E-04	1.19809594E-03	-8.32140693E-05	3.00086475E-04
12	0	-3.92213375E-03	1.41420631E-02	-3.28781103E-04	1.18576433E-03	0.	0.
12	-1	-3.96345727E-03	1.42911603E-02	-3.32229523E-04	1.19809594E-03	8.32140693E-05	-3.00086475E-04
12	-2	-4.09014224E-03	1.47479566E-02	-3.42948235E-04	1.23648864E-03	1.71797790E-04	-6.19405321E-04
12	-3	-4.31064695E-03	1.55428127E-02	-3.62239121E-04	1.30554646E-03	2.72196355E-04	-9.81011826E-04
12	-4	-4.64105922E-03	1.67332808E-02	-3.92716140E-04	1.41454784E-03	3.93483559E-04	-1.41729147E-03
12	-5	-5.10897888E-03	1.84179437E-02	-4.39299702E-04	1.58092128E-03	5.50263143E-04	-1.98021174E-03
12	-6	-5.76180714E-03	2.07657237E-02	-5.11573400E-04	1.83850727E-03	7.69132814E-04	-2.76406283E-03
12	-7	-6.68576228E-03	2.40828451E-02	-6.30077270E-04	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	8.55318683E-03	-2.95400203E-02
12	-11	-2.62917539E-02	9.55831131E-02	1.02580352E-03	4.09961357E-05	-3.50152909E-03	2.07692808E-03
12	-12	-3.72776720E-03	1.60634916E-02	5.82061141E-03	-2.40293415E-02	-1.70533830E-02	7.04581208E-02
13	13	-2.37982443E-03	1.94553051E-02	3.11216511E-03	-2.32790623E-02	9.91317750E-03	-7.42776469E-02
13	12	-1.52050771E-02	9.33755139E-02	3.40777951E-04	3.01418639E-03	1.40306647E-03	7.16239221E-03
13	11	-8.47075744E-03	5.08828967E-02	-1.67259381E-03	9.62191041E-03	-4.64421110E-03	2.67048472E-02
13	10	-5.71152866E-03	3.46965701E-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	1.86812756E-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	1.22417056E-02	-1.70025667E-04	1.04124164E-03	-1.27728500E-04	7.82204177E-04
13	2	-1.89954413E-03	1.16053736E-02	-1.62451508E-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	-1.81974877E-03	1.11164365E-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	-1.62451508E-04	9.95292673E-04	8.13619251E-05	-4.98476564E-04
13	-3	-2.00343382E-03	1.22417056E-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	1.86812756E-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	3.46965701E-02	-6.88665739E-04	4.10942874E-03	1.72968514E-03	-1.03199772E-02
13	-11	-8.47075744E-03	5.08828967E-02	-1.67259381E-03	9.62191041E-03	4.64421110E-03	-2.67048472E-02
13	-12	-1.52050771E-02	9.33755139E-02	3.40777951E-04	3.01418639E-03	-1.40306647E-03	-7.16239221E-03
13	-13	-2.37982443E-03	1.94553051E-02	3.11216511E-03	-2.32790623E-02	-9.91317750E-03	7.42776469E-02
14	14	-3.37797820E-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	4.75299637E-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	-8.97576116E-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	1.05526558E-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	8.88083239E-04	-4.73719298E-05	8.89327316E-04
14	3	-4.98856938E-04	9.21954634E-03	-4.46777593E-05	8.39990809E-04	-3.35588745E-05	6.30935449E-04

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

AIC CALCULATIONS

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

NU	MU	C VELOCITY POTENTIAL COEFFICIENT		W UPWASH COEFFICIENT		V SIDEWASH COEFFICIENT	
0	0	-4.99669441E-01	1.35612131E-02	0.	0.	0.	0.
1	0	-3.83780888E-01	3.72653793E-02	0.	0.	0.	0.
1	-1	-3.02537716E-01	3.51774045E-02	0.	0.	0.	0.
2	0	-1.58274582E-01	3.41050490E-02	0.	0.	0.	0.
2	-1	-1.98775746E-01	4.21384433E-02	0.	0.	0.	0.
2	-2	-2.06053452E-01	4.75173856E-02	0.	0.	0.	0.
3	0	-9.83703960E-02	3.28543730E-02	0.	0.	0.	0.
3	-1	-1.05489007E-01	3.51833148E-02	0.	0.	0.	0.
3	-2	-1.47785247E-01	4.86337200E-02	0.	0.	0.	0.
3	-3	-1.62295376E-01	5.66261105E-02	0.	0.	0.	0.
4	0	-6.81525261E-02	3.13856404E-02	0.	0.	0.	0.
4	-1	-7.08290748E-02	3.26038898E-02	0.	0.	0.	0.
4	-2	-8.09774972E-02	3.72153085E-02	0.	0.	0.	0.
4	-3	-1.18703126E-01	5.39547333E-02	0.	0.	0.	0.
4	-4	-1.34046042E-01	6.38424648E-02	0.	0.	0.	0.
5	0	-4.93500001E-02	2.96149940E-02	0.	0.	0.	0.
5	-1	-5.06323716E-02	3.03790933E-02	0.	0.	0.	0.
5	-2	-5.50235203E-02	3.29920612E-02	0.	0.	0.	0.
5	-3	-6.50871190E-02	3.89671498E-02	0.	0.	0.	0.
5	-4	-9.81522309E-02	5.81981438E-02	0.	0.	0.	0.
5	-5	-1.12751838E-01	6.96207606E-02	0.	0.	0.	0.
6	0	-3.63050688E-02	2.75563287E-02	0.	0.	0.	0.
6	-1	-3.70060878E-02	2.80858986E-02	0.	0.	0.	0.
6	-2	-3.92959019E-02	2.98145968E-02	0.	0.	0.	0.
6	-3	-4.39152690E-02	3.32964611E-02	0.	0.	0.	0.
6	-4	-5.31898899E-02	4.02714174E-02	0.	0.	0.	0.
6	-5	-8.19406102E-02	6.14878343E-02	0.	0.	0.	0.
6	-6	-9.52385406E-02	7.41890253E-02	0.	0.	0.	0.
7	0	-2.66803249E-02	2.52527061E-02	0.	0.	0.	0.
7	-1	-2.75944500E-02	2.56432429E-02	0.	0.	0.	0.
7	-2	-2.84124874E-02	2.68859576E-02	0.	0.	0.	0.

7	-3	-3.09101106E-02	2.92388864E-02	0.	0.	0.	0.
7	-4	-3.52822140E-02	3.33509956E-02	0.	0.	0.	0.
7	-5	-4.35481149E-02	4.11045928E-02	0.	0.	0.	0.
7	-6	-6.82881173E-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	2.27776282E-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	-1.38130848E-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	-4.55880825E-02	6.64147809E-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	-9.63939934E-03	1.81891284E-02	0.	0.	0.	0.
10	-3	-1.01853736E-02	1.92178725E-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	-3.57973860E-02	6.66196905E-02	0.	0.	0.	0.
10	-10	-4.24590939E-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	-6.08983020E-03	1.52958524E-02	0.	0.	0.	0.
11	-3	-6.41970067E-03	1.61239079E-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	-8.62033143E-03	2.16410619E-02	0.	0.	0.	0.
11	-7	-1.00533793E-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	-3.16568299E-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	1.20050222E-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	-3.18510175E-03	7.79147557E-02	0.	0.	0.	0.
15	0	2.39857385E-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.[illegible]5214217E-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	4.04138196E-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	1.161[illegible]60E-03	7.09745539E-03	0.	0.	0.	0.
16	-9	1.4[illegible]855805E-03	8.71031390E-03	0.	0.	0.	0.
16	-10	1.7[illegible]56494E-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	1.72261700E-02	0.	0.	0.	0.
16	-13	3.7183638[illegible]-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	7.15600868E-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	2.83984684E-04	1.06903946E-03	0.	0.	0.	0.
17	-4	3.85675563E-04	1.43250587E-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	9.22720363E-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 REAL	CHORD 1 IMAGINARY	CHORD 2 REAL	CHORD 2 IMAGINARY	CHORD 3 REAL	CHORD 3 IMAGINARY	CHORD 4 REAL	CHORD 4 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	[illegible]63324958E-02	3.19000000E-02	6.63324958E-02	3.19000000E-02
16	4.89641580E-02	-1.30749482E-02	4.93038417E-02	-1.22854610E-02	5.00166418E-02	-1.10011154E-02	5.13430376E-02	-1.08606884E-02
17	5.34053895E-02	-1.25650430E-02	5.35634354E-02	-1.14413488E-02	5.38681729E-02	-8.06011302E-03	5.40440732E-02	-9.10498770E-03
18	5.41381779E-02	-2.02172665E-02	5.43168889E-02	-1.95243635E-02	5.44424617E-02	-1.89244049E-02	5.46292480E-02	-2.12867906E-02
19	5.79542051E-02	-1.96828840E-02	5.78454023E-02	-1.88978027E-02	5.76530897E-02	-1.69356306E-02	5.69733825E-02	-1.89204435E-02
20	5.73734614E-02	-2.71512149E-02	5.71352697E-02	-2.71388378E-02	5.68188521E-02	-2.65568648E-02	5.62169272E-02	-2.88472801E-02
21	6.04595983E-02	-2.70321600E-02	6.02605942E-02	-2.62106415E-02	5.97677831E-02	-2.45199788E-02	5.84133933E-02	-2.65929915E-02
22	5.85434725E-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	6.03843890E-02	-3.12862399E-02	5.84167806E-02	-3.33057775E-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	6.05806479E-02	-4.00119142E-02	6.01789947E-02	-3.92066531E-02	5.93362001E-02	-3.75892667E-02	5.69321679E-02	-3.95670355E-02
26	5.66229702E-02	-4.59595304E-02	5.60746605E-02	-4.56029133E-02	5.52333135E-02	-4.44969255E-02	5.30926973E-02	-4.64212[illegible]E-02
27	5.85342587E-02	-4.54480721E-02	5.81445607E-02	-4.47221836E-02	5.73011295E-02	-4.31999606E-02	5.46365432E-02	-4.51101685E-02
28	5.37129457E-02	-5.07683932E-02	5.32926932E-02	-5.03777030E-02	5.25698835E-02	-4.92341998E-02	5.02389327E-02	-5.10742794E-02
29	5.55032801E-02	-5.02873669E-02	5.51823218E-02	-4.96013106E-02	5.44977244E-02	-4.81416667E-02	5.17346519E-02	-4.99633399E-02

ROW	CHORD 5 REAL	CHORD 5 IMAGINARY	CHORD 6 REAL	CHORD 6 IMAGINARY	CHORD 7 REAL	CHORD 7 IMAGINARY	CHORD 8 REAL	CHORD 8 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	9.90000000
6	6.63324958E-02	1.21000000E-02	6.63324958E-02	1.21000000E-02	6.63324958E-02	1.21000000E-02	6.63324958E-02	1.21000000
7	6.63324958E-02	1.43000000E-02	6.63324958E-02	1.43000000E-02	6.63324958E-02	1.43000000E-02	6.63324958E-02	1.43000000
8	6.63324958E-02	1.65000000E-02	6.63324958E-02	1.65000000E-02	6.63324958E-02	1.65000000E-02	6.63324958E-02	1.65000000
9	6.63324958E-02	1.87000000E-02	6.63324958E-02	1.87000000E-02	6.63324958E-02	1.87000000E-02	6.63324958E-02	1.87000000
10	6.63324958E-02	2.09000000E-02	6.63324958E-02	2.09000000E-02	6.63324958E-02	2.09000000E-02	6.63324958E-02	2.09000000
11	6.63324958E-02	2.31000000E-02	6.63324958E-02	2.31000000E-02	6.63324958E-02	2.31000000E-02	6.63324958E-02	2.31000000
12	6.63324958E-02	2.53000000E-02	6.63324958E-02	2.53000000E-02	6.63324958E-02	2.53000000E-02	6.63324958E-02	2.53000000
13	6.63324958E-02	2.75000000E-02	6.63324958E-02	2.75000000E-02	6.63324958E-02	2.75000000E-02	6.63324958E-02	2.75000000
14	6.63324958E-02	2.97000000E-02	6.63324958E-02	2.97000000E-02	6.63324958E-02	2.97000000E-02	6.63324958E-02	2.97000000
15	6.63324958E-02	3.19000000E-02	6.63324958E-02	3.19000000E-02	6.63324958E-02	3.19000000E-02	6.63324958E-02	3.19000000

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 5 REAL	CHORD 5 IMAGINARY	CHORD 6 REAL	CHORD 6 IMAGINARY	CHORD 7 REAL	CHORD 7 IMAGINARY	CHORD 8 REAL	CHORD 8 IMAGINARY
16	5.30176346E-02	-9.91932790E-04	5.50613145E-02	1.04159983E-02	5.69535875E-02	1.23276743E-02	5.88563377E-02	1.49505673E-02
17	5.43792077E-02	-5.14658827E-03	5.64545688E-02	6.87301296E-03	5.88168279E-02	1.24604221E-02	6.05004025E-02	1.52164635E-02
18	5.55270913E-02	-1.44806783E-02	5.75290193E-02	-1.04166120E-03	5.99711433E-02	7.29512421E-03	6.20595209E-02	1.32708146E-02
19	5.67210504E-02	-1.47246991E-02	5.82865856E-02	-3.40550016E-03	6.00788565E-02	3.32387786E-03	6.20850019E-02	1.00460554E-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	5.77773679E-02	-2.21106111E-02	5.90086164E-02	-1.10086512E-02	5.99448603E-02	-4.79230985E-03	6.10546534E-02	1.06655474E-03
22	5.65311153E-02	-2.93813307E-02	5.83963492E-02	-1.68161852E-02	5.9527644 [illegible]-02	-9.65812058E-03	6.05020749E-02	-3.49853951E-03
23	5.73788943E-02	-2.87875354E-02	5.86483818E-02	-1.78121969E-02	5.93340972E-02	-1.17494930E-02	6.01318867E-02	-6.13287008E-03
24	5.52344872E-02	-3.54524593E-02	5.74448841E-02	-2.32323246E-02	5.85694276E-02	-1.62468669E-02	5.94703856E-02	-1.01681791E-02
25	5.60419323E-02	-3.48533143E-02	5.77589677E-02	-2.36914847E-02	5.84255983E-02	-1.79039209E-02	5.91504993E-02	-1.23946730E-02
26	5.31226009E-02	-4.09267681E-02	5.59590667E-02	-2.89583669E-02	5.73138244E-02	-2.20966980E-02	5.82993476E-02	-1.60938672E-02
27	5.39440323E-02	-4.03352086E-02	5.62390469E-02	-2.94608082E-02	5.71513511E-02	-2.34842761E-02	5.80062583E-02	-1.80323749E-02
28	5.04753262E-02	-4.57299853E-02	5.39317467E-02	-3.41047884E-02	5.55889660E-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	-2.85786082E-02	5.64938414E-02	-2.32117324E-02

ROW	CHORD 9 REAL	CHORD 9 IMAGINARY	CHORD 10 REAL	CHORD 10 IMAGINARY	CHORD 11 REAL	CHORD 11 IMAGINARY	CHORD 12 REAL	CHORD 12 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	-1.03366260E-01	1.57205476E-02	-1.65889770E-02	5.53256845E-03
6	6.63324958E-02	1.21000000E-02	6.63324958E-02	1.21000000E-02	-1.20361173E-01	2.09651249E-02	-2.42536793E-02	8.65275973E-03
7	6.63324958E-02	1.43000000E-02	6.63324958E-02	1.43000000E-02	-1.31800755E-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	-4.12400257E-02	2.05096602E-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	4.55466611E-02	-4.54944213E-02	2.42402706E-02
14	6.63324958E-02	2.97000000E-02	6.63324958E-02	2.97000000E-02	-1.83251245E-01	4.70548004E-02	-4.78900387E-02	2.63266740E-02
15	6.63324958E-02	3.19000000E-02	6.63324958E-02	3.19000000E-02	-1.86493052E-01	4.69535457E-02	-4.85573078E-02	2.67945501E-02
16	6.09179906E-02	1.84679076E-02	6.31184786E-02	2.39654324E-02	-1.90119958E-01	4.68089760E-02	-5.03699424E-02	2.82128903E-02
17	6.23204447E-02	1.85005614E-02	6.33756342E-02	2.26240652E-02	-1.90328165E-01	5.10933937E-02	-5.09283631E-02	2.80540906E-02
18	6.30009000E-02	1.64886655E-02	6.33470944E-02	2.00508602E-02	-1.90092157E-01	5.68633914E-02	-5.20635292E-02	2.92283809E-02
19	6.34150301E-02	1.47885497E-02	6.37584871E-02	1.80803960E-02	-1.88250356E-01	6.30070751E-02	-5.14282272E-02	3.03414323E-02

20	6.30432416E-02	1.03561105E-02	6.39999152E-02	1.56041868E-02	-1.66882715E-01	6.90765780E-02	-5.14090667E-02	3.22391763E-02
21	6.21532067E-02	6.29835288E-03	6.36310965E-02	1.21088638E-02	-1.84303575E-01	7.50734456E-02	-5.02526410E-02	3.34852678E-02
22	6.14992208E-02	2.07468036E-03	6.24588453E-02	7.77605235E-03	-1.83175413E-01	7.98877033E-02	-4.97309900E-02	3.53380366E-02
23	6.09647776E-02	-9.99128631E-04	6.19944156E-02	4.46565443E-03	-1.81492997E-01	8.41865910E-02	-4.92946749E-02	3.55649950E-02
24	6.02810218E-02	-4.71283641E-03	6.11197429E-02	9.56613651E-04	-1.79884154E-01	8.90572969E-02	-4.98551296E-02	3.61484949E-02
25	5.98818112E-02	-7.32674558E-03	6.07745348E-02	-1.94795443E-03	-1.77350702E-01	9.38154949E-02	-4.93220213E-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	-4.83770563E-02	3.80963244E-02
28	5.74690969E-02	-1.61133529E-02	5.88382303E-02	-1.05127273E-02	-1.69066770E-01	1.08791922E-01	-4.78577853E-02	3.93204051E-02
29	5.74475102E-02	-1.82508756E-02	5.85207798E-02	-1.29608019E-02	-1.65445011E-01	1.13599786E-01	-4.67697855E-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	2.76973082E-03	-1.94322794E-03	9.14503279E-04	0.	0.	0.	0.
6	-1.14663490E-02	4.75141796E-03	-4.66356705E-03	2.37317247E-03	-1.33621092E-03	8.05363569E-04	0.	0.
7	-1.44401398E-02	6.72365208E-03	-6.94868535E-03	3.93003987E-03	-3.12042302E-03	2.05351456E-03	-9.40314423E-04	7.17394071E-04
8	-1.81664905E-02	9.05484645E-03	-9.47261486E-03	5.78202322E-03	-4.80292119E-03	3.43996838E-03	-2.20337043E-03	1.81824070E-03
9	-2.02131926E-02	1.09509815E-02	-1.12706190E-02	7.47864451E-03	-6.39605141E-03	4.98731558E-03	-3.37792331E-03	3.03931286E-03
10	-2.32317936E-02	1.33351588E-02	-1.33298884E-02	9.41936548E-03	-7.82872197E-03	6.54581970E-03	-4.53058056E-03	4.41260790E-03
11	-2.45529999E-02	1.49535110E-02	-1.45765125E-02	1.10071844E-02	-9.00767772E-03	8.09634408E-03	-5.48306080E-03	5.75757718E-03
12	-2.68835747E-02	1.71581298E-02	-1.61628975E-02	1.28415526E-02	-1.01029280E-02	9.62019278E-03	-6.35119173E-03	7.14683044E-03
13	-2.76730137E-02	1.83620450E-02	-1.69400700E-02	1.41529826E-02	-1.08807889E-02	1.09959437E-02	-7.01319218E-03	8.42304855E-03
14	-2.94323537E-02	2.02243594E-02	-1.81283837E-02	1.57496171E-02	-1.16728875E-02	1.23521301E-02	-7.60542406E-03	9.68264618E-03
15	-2.96841840E-02	2.09355359E-02	-1.85712314E-02	1.66833163E-02	-1.21382507E-02	1.34415927E-02	-8.01383648E-03	1.07675525E-02
16	-3.12328537E-02	2.23541072E-02	-1.94720907E-02	1.79558435E-02	-1.27091411E-02	1.45444449E-02	-8.39943021E-03	1.18087750E-02
17	-3.15438555E-02	2.25455495E-02	-1.97418835E-02	1.84647869E-02	-1.29896767E-02	1.52873042E-02	-8.64480352E-03	1.26291324E-02
18	-3.26370046E-02	2.34817603E-02	-2.04616947E-02	1.93748727E-02	-1.34213285E-02	1.60961994E-02	-8.89608598E-03	1.34087037E-02
19	-3.26359469E-02	2.36484018E-02	-2.07060065E-02	1.94660691E-02	-1.36409376E-02	1.64825469E-02	-9.07545007E-03	1.39392964E-02
20	-3.28643249E-02	2.50550314E-02	-2.11489786E-02	2.02693319E-02	-1.40545680E-02	1.69978187E-02	-9.30568956E-03	1.44530753E-02
21	-3.22014537E-02	2.55998631E-02	-2.08702972E-02	2.05794628E-02	-1.41270885E-02	1.72172288E-02	-9.50493914E-03	1.47044550E-02
22	-3.19347106E-02	2.70052876E-02	-2.07607923E-02	2.15422036E-02	-1.41230280E-02	1.78460528E-02	-9.64058979E-03	1.50880083E-02
23	-3.10061574E-02	2.75224811E-02	-2.01832499E-02	2.18353212E-02	-1.38141988E-02	1.81072346E-02	-9.52991494E-03	1.53505958E-02
24	-3.14155437E-02	2.80695849E-02	-1.98079084E-02	2.27253158E-02	-1.35438422E-02	1.87063391E-02	-9.35848176E-03	1.57512844E-02
25	-3.17087588E-02	2.76185349E-02	-2.00256866E-02	2.23575239E-02	-1.30835002E-02	1.88705713E-02	-9.07008221E-03	1.59323004E-02
26	-3.19525492E-02	2.82438447E-02	-2.07941214E-02	2.24824962E-02	-1.35689011E-02	1.89658124E-02	-8.76404768E-03	1.62611932E-02
27	-3.17216688E-02	2.81994459E-02	-2.09429577E-02	2.22170399E-02	-1.42786537E-02	1.85661266E-02	0.	0.
28	-3.15969918E-02	2.89886994E-02	-2.11453359E-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	6.46577963E-04	0.	0.	0.	0.	0.	0.
9	-1.56809326E-03	1.63111099E-03	-5.00370367E-04	5.90566475E-04	0.	0.	0.	0.
10	-2.43016860E-03	2.72071381E-03	-1.15025148E-03	1.47036362E-03	-3.58996464E-04	5.31856260E-04	0.	0.
11	-3.23673723E-03	3.93950114E-03	-1.75198522E-03	2.45469521E-03	-8.22442031E-04	1.32669563E-03	-2.56037860E-04	4.85044908E-04
12	-3.90668812E-03	5.13412743E-03	-2.31126880E-03	3.53736006E-03	-1.23591964E-03	2.20743583E-03	-5.74993281E-04	1.20117793E-03
13	-4.47622985E-03	6.34827040E-03	-2.75948579E-03	4.60117377E-03	-1.60958492E-03	3.17006585E-03	-8.47535685E-04	1.99452066E-03
14	-4.92596936E-03	7.48156522E-03	-3.12763024E-03	5.67425163E-03	-1.88873132E-03	4.12041401E-03	-1.07577483E-03	2.85952186E-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	-5.55821950E-03	9.52779218E-03	-3.58969376E-03	7.61387747E-03	-2.23425180E-03	5.94108762E-03	-1.32058094E-03	4.55659[illegible]E-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	6.02025724E-03
19	-5.99549060E-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	9.14655378E-03	-1.17650478E-03	7.67500306E-03
22	-6.39589685E-03	1.29395270E-02	-4.01009098E-03	1.11378211E-02	-2.30565947E-03	9.50465000E-03	-1.10604250E-03	8.06202447E-03
23	-6.46741106E-03	1.31740323E-02	-4.13444493E-03	1.13768915E-02	-2.35486728E-03	9.79888686E-03	-1.09550927E-03	8.37857866E-03
24	-6.38760843E-03	1.34638455E-02	-4.17073026E-03	1.15987044E-02	-2.42197068E-03	1.00047719E-02	0.	0.
25	-6.23331690E-03	1.36528040E-02	0.	0.	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

PAGE CONTINUED

ROW	CHORD 21 REAL	CHORD 21 IMAGINARY	CHORD 22 REAL	CHORD 22 IMAGINARY	CHORD 23 REAL	CHORD 23 IMAGINARY	CHORD 24 REAL	CHORD 24 IMAGINARY
12	-1.74717044E-04	4.41134579E-04	0.	0.	0.	0.	0.	0.
13	-3.82337016E-04	1.08787407E-03	-1.10360272E-04	4.00858990E-04	0.	0.	0.	0.
14	-5.42889730E-04	1.80198221E-03	-2.29793159E-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	-2.04470693E-05	3.28139270E-04
16	-7.10895882E-04	3.32035161E-03	-3.30188758E-04	2.31228605E-03	-1.07 [illegible]120[illegible]E-04	1.46500629E-03	-1.28717102E-05	7.97961850E-04
17	-7.21670133E-04	4.05775393E-03	-3.13242567E-04	2.97251953E-03	-7.64634981E-05	2.07068864E-03	2.68935821E-05	1.30830854E-03
18	-6.56274652E-04	4.72743578E-03	-2.24545485E-04	3.61911566E-03	3.07807933E-05	2.65147306E-03	1.43172505E-04	1.84630316E-03
19	-5.95249172E-04	5.35402309E-03	-1.38063385E-04	4.20650277E-03	1.36125841E-04	3.22025310E-03	2.56141846E-04	2.35903835E-03
20	-4.77520700E-04	5.89323467E-03	1.08203342E-05	4.74381945E-03	3.05080543E-04	3.72424214E-03	0.	0.
21	-3.88257274E-04	6.37059695E-03	1.35843501E-04	5.21282666E-03	0.	0.	0.	0.
22	-2.72809739E-04	6.76716545E-03	0.	0.	0.	0.	0.	0.

ROW	CHORD 25 REAL	CHORD 25 IMAGINARY	CHORD 26 REAL	CHORD 26 IMAGINARY
16	2.11855982E-05	2.95121951E-04	0.	0.
17	6.25336556E-05	7.14594872E-04	3.34829454E-05	2.63851131E-04
18	1.82843887E-04	1.16460852E-03	0.	0.

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

ROW	CHORD 1 REAL	CHORD 1 IMAGINARY	CHORD 2 REAL	CHORD 2 IMAGINARY	CHORD 3 REAL	CHORD 3 IMAGINARY	CHORD 4 REAL	CHORD 4 IMAGINARY
19	-5.91117621E-02	-2.47284538E-02	-5.84920791E-02	-2.46367453E-02	-5.72564692E-02	-2.44095793E-02	-5.51534895E-02	-2.42748960E-02
20	-2.48461090E-02	7.62604131E-03	-2.46804922E-02	7.66129327E-03	-2.53310010E-02	5.95447091E-03	-2.63939790E-02	2.21401863E-03
21	-3.06898118E-02	6.83011666E-03	-3.13032001E-02	4.88595339E-05	-3.08401114E-02	3.69555450E-03	-3.00202668E-02	2.03175787E-03
22	-2.96068597E-02	1.20060762E-02	-2.86532315E-02	1.26432369E-02	-2.85890768E-02	1.16241815E-02	-2.81264626E-02	9.75185104E-03
23	-3.39291768E-02	1.22802512E-02	-3.40362409E-02	1.10930708E-02	-3.32165604E-02	9.95689165E-03	-3.21064546E-02	8.10891405E-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	1.29580866E-02
26	-3.18115445E-02	2.23868962E-02	-3.12358647E-02	2.19958548E-02	-3.03487004E-02	2.09051624E-02	-2.93686729E-02	1.88435648E-02
27	-3.56790652E-02	2.21230891E-02	-3.53209339E-02	2.11576834E-02	-3.42314106E-02	1.96584595E-02	-3.28952760E-02	1.75115528E-02
28	-3.11340922E-02	2.60897632E-02	-3.04566532E-02	2.56364876E-02	-2.94349385E-02	2.45211349E-02	-2.84291956E-02	2.24151502E-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	2.91053778E-02	-2.88783485E-02	2.85903589E-02	-2.79622851E-02	2.74651740E-02	-2.70761231E-02	2.53743520E-02
31	-3.26952241E-02	2.89805176E-02	-3.23820660E-02	2.81842294E-02	-3.15038605E-02	2.67441671E-02	-3.03207664E-02	2.45837878E-02
32	-2.72462488E-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 REAL	CHORD 5 IMAGINARY	CHORD 6 REAL	CHORD 6 IMAGINARY	CHORD 7 REAL	CHORD 7 IMAGINARY	CHORD 8 REAL	CHORD 8 IMAGINARY
19	-5.21239318E-02	-2.39568863E-02	-4.79293884E-02	-2.34355827E-02	-4.23383163E-02	-2.26869769E-02	-3.49877890E-02	-2.12205154E-02
20	-2.65862529E-02	-1.01561359E-03	-2.58952780E-02	-4.16937027E-03	-2.45644304E-02	-7.96759246E-03	-2.16729183E-02	-1.07745919E-02
21	-2.92213121E-02	-5.47973131E-04	-2.73445926E-02	-2.77955362E-03	-2.48483538E-02	-5.32290548E-03	-2.16808426E-02	-8.90978012E-03
22	-2.73426720E-02	7.10127731E-03	-2.67000506E-02	2.89845759E-03	-2.49032017E-02	-1.75247646E-03	-2.05845508E-02	-5.28908729E-03
23	-3.04962510E-02	5.80160599E-03	-2.80354560E-02	3.08336269E-03	-2.45566608E-02	-3.51999370E-06	-2.00419223E-02	-3.50518960E-03
24	-2.83361607E-02	1.16844286E-02	-2.64446779E-02	7.89827473E-03	-2.36806037E-02	3.49285337E-03	-1.92051360E-02	-6.75563461E-04
25	-3.10470633E-02	1.05926293E-02	-2.79477869E-02	7.71316302E-03	-2.38625576E-02	4.38741259E-03	-1.87358939E-02	6.66927359E-04
26	-2.81670795E-02	1.57413325E-02	-2.61020249E-02	1.18385678E-02	-2.29706639E-02	7.38917661E-03	-1.80625267E-02	2.98958983E-03
27	-3.05765698E-02	1.48882223E-02	-2.75293657E-02	1.16549883E-02	-2.34166344E-02	7.92846070E-03	-1.81323607E-02	3.84939346E-03
28	-2.71305877E-02	1.93245717E-02	-2.51600203E-02	1.53230418E-02	-2.21926048E-02	1.06830386E-02	-1.73781630E-02	5.90809682E-03
29	-2.95788211E-02	1.85767642E-02	-2.66058760E-02	1.51246384E-02	-2.26697574E-02	1.10511189E-02	-1.75375588E-02	6.57607054E-03
30	-2.58358739E-02	2.22935932E-02	-2.39751045E-02	1.82512376E-02	-2.11095875E-02	1.34970067E-02	-1.65410037E-02	8.40865099E-03
31	-2.82885733E-02	2.17310618E-02	-2.54841924E-02	1.80687860E-02	-2.17556449E-02	1.37261493E-02	-1.67579133E-02	8.91770117E-03
32	-2.42778165E-02	2.47232123E-02	-2.26292002E-02	2.06877060E-02	-1.99585437E-02	1.58357227E-02	-1.56430476E-02	1.04966373E-02
33	-2.66638384E-02	2.43114805E-02	-2.41465345E-02	2.05405601E-02	-2.07147208E-02	1.60001373E-02	-1.59561423E-02	1.08696999E-02

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

19	-2.55259243E-02	-1.97606723E-02	-1.65726275E-02	-1.89298755E-02	0.	0.	0.	0.
20	-1.61412604E-02	-1.22913695E-02	-9.60053148E-03	-1.36362870E-02	1.10794454E-02	1.05955449E-02	0.	0.
21	-1.74744322E-02	-1.32599814E-02	-1.02335305E-02	-1.41645819E-02	1.78471129E-02	1.25946556E-02	1.16311944E-03	6.26503715E-04
22	-1.57567273E-02	-9.31559556E-03	-9.84003651E-03	-1.35098295E-02	2.81171848E-02	1.46408357E-02	4.61711867E-03	2.28436453E-03
23	-1.49396138E-02	-8.09635379E-03	-8.38649828E-03	-1.12506069E-02	3.60612125E-02	1.27059992E-02	6.58320169E-03	1.58140658E-03
24	-1.39520516E-02	-5.78332840E-03	-7.17765922E-03	-9.56439179E-03	4.17276699E-02	7.21638329E-03	1.04482331E-02	1.59428211E-03
25	-1.35229399E-02	-4.42223203E-03	-6.76934638E-03	-8.26180481E-03	4.41397703E-02	-7.34915422E-04	1.15768673E-02	-6.45210097E-04
26	-1.26624036E-02	-2.53040020E-03	-5.87722163E-03	-6.93507078E-03	4.69572737E-02	-8.37461989E-03	1.33856852E-02	-2.76251588E-03
27	-1.25910156E-02	-1.54149850E-03	-5.62035202E-03	-5.90032363E-03	4.63529653E-02	-1.83111331E-02	1.33836910E-02	-5.82981572E-03
28	-1.20677130E-02	-2.56646955E-05	-5.06859491E-03	-4.90644161E-03	4.51872967E-02	-2.74816020E-02	1.39189235E-02	-8.64355010E-03
29	-1.21529590E-02	7.74236549E-04	-5.20730714E-03	-4.10311505E-03	4.27370943E-02	-3.64239780E-02	1.24981795E-02	-1.18814964E-02

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 REAL	CHORD 9 IMAGINARY	CHORD 10 REAL	CHORD 10 IMAGINARY	CHORD 11 REAL	CHORD 11 IMAGINARY	CHORD 12 REAL	CHORD 12 IMAGINARY
30	-1.15607068E-02	2.09657463E-03	-4.75938727E-03	-3.29669287E-03	4.09917565E-02	-4.48305773E-02	1.20979134E-02	-1.46844504E-02
31	-1.16434588E-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	3.87724276E-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 REAL	CHORD 13 IMAGINARY	CHORD 14 REAL	CHORD 14 IMAGINARY	CHORD 15 REAL	CHORD 15 IMAGINARY	CHORD 16 REAL	CHORD 16 IMAGINARY
22	1.18454473E-03	6.89342320E-04	0.	0.	0.	0.	0.	0.
23	2.75826523E-03	9.38935130E-04	7.51679787E-04	3.13609382E-04	0.	0.	0.	0.
24	4.83053318E-03	8.93452950E-04	1.94020972E-03	4.52327671E-04	5.66505863E-04	1.70718127E-04	0.	0.
25	6.59107323E-03	1.00952742E-04	3.19889254E-03	1.39905863E-04	1.41981786E-03	1.59598195E-04	4.41015186E-04	8.17972168E-05
26	8.03644877E-03	-1.11508997E-03	4.65586578E-03	-4.09409200E-04	2.38709161E-03	-1.50783780E-04	1.09673289E-03	2.23414303E-06
27	8.40911009E-03	-3.04584920E-03	5.33423333E-03	-1.53639139E-03	3.38667695E-03	-7.25777348E-04	1.82619159E-03	-3.29137268E-04
28	9.03934072E-03	-4.89832448E-03	5.84244144E-03	-2.82022265E-03	3.92398929E-03	-1.60534939E-03	0.	0.
29	8.44797231E-03	-7.14273830E-03	5.81398097E-03	-4.34614532E-03	0.	0.	0.	0.
30	8.02748330E-03	-9.13928362E-03	0.	0.	0.	0.	0.	0.

ROW	CHORD 17 REAL	CHORD 17 IMAGINARY
26	3.54947168E-04	2.67341355E-05

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

WING VELOCITY POTENTIALS

(MACH 1.200 RED. FREQ.= .50000)

MODE SHAPE 2

ROW	CHORD 1 REAL	CHORD 1 IMAGINARY	CHORD 2 REAL	CHORD 2 IMAGINARY	CHORD 3 REAL	CHORD 3 IMAGINARY	CHORD 4 REAL	CHORD 4 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	-8.78809499E-01	2.36376541E-01
9	-9.77902331E-01	3.05162316E-01	-9.77902331E-01	3.05162316E-01	-9.68906048E-01	2.94662062E-01	-9.44152688E-01	2.69584305E-01
10	-1.05679945E+00	3.65581916E-01	-1.04945283E+00	3.54839787E-01	-1.02901545E+00	3.29063251E-01	-9.97376643E-01	2.96151547E-01
11	-1.12041939E+00	4.15341562E-01	-1.10393729E+00	3.89161759E-01	-1.07795893E+00	3.55521572E-01	-1.04147231E+00	3.16894320E-01
12	-1.16482969E+00	4.37437154E-01	-1.14839260E+00	4.14327181E-01	-1.11856163E+00	3.75043558E-01	-1.07837117E+00	3.31908232E-01
13	-1.19683799E+00	4.34075395E-01	-1.18244992E+00	4.20748755E-01	-1.15273527E+00	3.87880205E-01	-1.10961148E+00	3.41279931E-01
14	-1.22347731E+00	4.14853563E-01	-1.20910451E+00	4.04809431E-01	-1.18025898E+00	3.83665452E-01	-1.13651259E+00	3.45050063E-01
15	-1.24801429E+00	3.83210227E-01	-1.23336063E+00	3.75190194E-01	-1.20395529E+00	3.58850050E-01	-1.15961275E+00	3.32953755E-01

ROW	CHORD 5 REAL	CHORD 5 IMAGINARY	CHORD 6 REAL	CHORD 6 IMAGINARY	CHORD 7 REAL	CHORD 7 IMAGINARY	CHORD 8 REAL	CHORD 8 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	-4.31035421E-01	5.18509973E-02
5	-5.72820386E-01	9.61220392E-02	-5.72820386E-01	9.61220392E-02	-5.54896716E-01	8.79472804E-02	-5.04367922E-01	6.90987110E-02
6	-6.86349473E-01	1.40572481E-01	-6.71238086E-01	1.31584297E-01	-6.29446382E-01	1.10566235E-01	-5.63710060E-01	8.49252329E-02
7	-7.79422470E-01	1.81520812E-01	-7.44311854E-01	1.58781501E-01	-6.90148366E-01	1.30552859E-01	-6.13078893E-01	9.94721747E-02
8	-8.49236846E-01	2.12298217E-01	-8.03921915E-01	1.82037984E-01	-7.40682396E-01	1.48208497E-01	-6.54641651E-01	1.12474107E-01
9	-9.06162895E-01	2.37781841E-01	-8.53471496E-01	2.01821332E-01	-7.83107124E-01	1.63319440E-01	-6.89725341E-01	1.23653293E-01
10	-9.53383296E-01	2.58611438E-01	-8.94951395E-01	2.17991041E-01	-8.18828469E-01	1.75651617E-01	-7.19452286E-01	1.32729028E-01
11	-9.92850345E-01	2.74750360E-01	-9.29824763E-01	2.30394837E-01	-8.49019485E-01	1.84954421E-01	-7.44581289E-01	1.39385784E-01
12	-1.02607161E+00	2.86153959E-01	-9.59320766E-01	2.38846379E-01	-8.74578487E-01	1.90965435E-01	-7.65941648E-01	1.43432165E-01
13	-1.05432670E+00	2.92734807E-01	-9.84439589E-01	2.43170154E-01	-8.96410824E-01	1.93533520E-01	-7.84144167E-01	1.44638073E-01
14	-1.07869460E+00	2.94424260E-01	-1.00615700E+00	2.43293291E-01	-9.15257652E-01	1.92505383E-01	-7.99889576E-01	1.42924930E-01
15	-1.10020318E+00	2.91265457E-01	-1.02530542E+00	2.39162874E-01	-9.31691651E-01	1.87866023E-01	-8.13728078E-01	1.36189207E-01

ROW	CHORD 9 REAL	CHORD 9 IMAGINARY	CHORD 10 REAL	CHORD 10 IMAGINARY

1	-6.63184769E-02	6.99825447E-04	-6.63184769E-02	6.39825447E-04
2	-1.97801526E-01	1.06032274E-02	-1.57587972E-01	6.60200028E-03
3	-2.99774444E-01	2.43866111E-02	-2.08751114E-01	1.19663897E-02
4	-3.67333527E-01	3.57494077E-02	-2.48534137E-01	1.76262658E-02
5	-4.21240420E-01	4.67293868E-02	-2.80706427E-01	2.34318670E-02
6	-4.65829557E-01	5.73101529E-02	-3.08069576E-01	2.92289546E-02
7	-5.03457442E-01	6.72248226E-02	-3.30969647E-01	3.46723412E-02
8	-5.35252699E-01	7.61605875E-02	-3.50743423E-01	3.96982473E-02
9	-5.62332130E-01	8.38794754E-02	-3.67349192E-01	4.39258200E-02
10	-5.85231425E-01	9.00364358E-02	-3.81661317E-01	4.73447109E-02
11	-6.04717836E-01	9.44563009E-02	-3.93638296E-01	4.96062124E-02

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

WING VELOCITY POTENTIALS

(MACH 1.200 RED. FREQ.= .50000)

MODE SHAPE 2

PAGE CONTINUED

ROW	CHORD 9		CHORD 10	
	REAL	IMAGINARY	REAL	IMAGINARY
12	-6.21205689E-01	9.66617335E-02	-4.03949368E-01	5.07496370E-02
13	-6.35321693E-01	9.71596348E-02	-4.12613392E-01	5.04889462E-02
14	-6.47451767E-01	9.51715199E-02	-4.20176166E-01	4.89172886E-02
15	-6.58142073E-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 REAL	CHORD 1 IMAGINARY	CHORD 2 REAL	CHORD 2 IMAGINARY	CHORD 3 REAL	CHORD 3 IMAGINARY	CHORD 4 REAL	CHORD 4 IMAGINARY
19	5.97433780E-02	2.31088510E-02	5.91216173E-02	2.30340104E-02	5.78806629E-02	2.28405074E-02	5.57754206E-02	2.27629510E-02
20	1.46470060E-01	2.78757365E-02	1.45057517E-01	2.77702938E-02	1.43125207E-01	2.94306397E-02	1.40045624E-01	3.31633513E-02
21	2.01552933E-01	-5.37579813E-03	2.01274448E-01	-2.32696529E-03	1.99487290E-01	2.62429453E-03	1.96206358E-01	1.04770871E-02
22	2.56580431E-01	-4.70753809E-02	2.55802374E-01	-4.30567946E-02	2.54021831E-01	-3.50384370E-02	2.49149730E-01	-2.41029791E-02
23	3.08448622E-01	-9.74763524E-02	3.06874214E-01	-9.26085968E-02	3.03440111E-01	-8.27094581E-02	2.97336699E-01	-6.75252971E-02
24	3.54545105E-01	-1.55595509E-01	3.52155682E-01	-1.49457387E-01	3.47013540E-01	-1.37057780E-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	3.72801832E-01	-1.71411789E-01
26	4.23409796E-01	-2.85771694E-01	4.19514982E-01	-2.75976075E-01	4.11430948E-01	-2.56308588E-01	3.94204272E-01	-2.28690102E-01
27	4.44295615E-01	-3.52018374E-01	4.39753714E-01	-3.40039537E-01	4.25972110E-01	-3.17893554E-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	4.54332338E-01	-4.72205637E-01	4.40185045E-01	-4.59374499E-01	4.19285624E-01	-4.35215577E-01	3.91594525E-01	-4.00829068E-01
30	4.37143590E-01	-5.22134107E-01	4.25676392E-01	-5.10412302E-01	4.03243902E-01	-4.86273229E-01	3.74763514E-01	-4.50826051E-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	3.74898221E-01	-5.97697388E-01	3.66908708E-01	-5.86474744E-01	3.52562795E-01	-5.64599112E-01	3.29866176E-01	-5.30358871E-01
33	3.36840700E-01	-6.22786219E-01	3.31036845E-01	-6.12047491E-01	3.19023216E-01	-5.90246446E-01	3.01479551E-01	-5.57447765E-01

ROW	CHORD 5 REAL	CHORD 5 IMAGINARY	CHORD 6 REAL	CHORD 6 IMAGINARY	CHORD 7 REAL	CHORD 7 IMAGINARY	CHORD 8 REAL	CHORD 8 IMAGINARY
19	5.27392407E-02	2.25273225E-02	4.85333313E-02	2.21201277E-02	4.29256515E-02	2.15236602E-02	3.55402098E-02	2.02575324E-02
20	1.34119663E-01	3.64134763E-02	1.24965086E-01	3.94574982E-02	1.12265350E-01	4.27409130E-02	9.44370744E-02	4.47694904E-02
21	1.90130695E-01	2.03593159E-02	1.78897556E-01	3.00571568E-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	4.28959115E-02
23	2.86599571E-01	-4.77504760E-02	2.70773363E-01	-2.26849413E-02	2.43073813E-01	2.25471358E-03	2.00846176E-01	2.11704993E-02
24	3.29026922E-01	-9.13774307E-02	3.01548347E-01	-6.24702939E-02	2.65034587E-01	-3.42866084E-02	2.19522889E-01	-8.53602719E-03
25	3.52230953E-01	-1.41518821E-01	3.20329212E-01	-1.09050184E-01	2.79463213E-01	-7.58958706E-02	2.29167812E-01	-4.44157259E-02
26	3.66273073E-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	3.32654184E-01	-2.10039671E-01	2.85300485E-01	-1.66149224E-01	2.31124392E-01	-1.21215542E-01
28	3.68089941E-01	-3.05952223E-01	3.27263856E-01	-2.60338892E-01	2.80059837E-01	-2.10677344E-01	2.25905927E-01	-1.58883438E-01
29	3.57083773E-01	-3.57939545E-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	3.21939231E-01	-4.47771530E-01	2.84285562E-01	-3.91774332E-01	2.41727585E-01	-3.28257770E-01	1.93798630E-01	-2.58994292E-01
32	2.99267426E-01	-4.83436870E-01	2.64207216E-01	-4.25935686E-01	2.24422900E-01	-3.59638340E-01	1.79326366E-01	-2.86202224E-01
33	2.75979176E-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 REAL	CHORD 9 IMAGINARY	CHORD 10 REAL	CHORD 10 IMAGINARY

19	2.60450000E-02	1.90552832E-02	1.70750952E-02	1.84678707E-02
20	7.15521769E-02	4.57770154E-02	4.11306969E-02	3.68189604E-02
21	1.04803119E-01	5.50223982E-02	5.71184533E-02	4.02919310E-02
22	1.30132826E-01	4.93833270E-02	7.50743746E-02	4.05107805E-02
23	1.50055431E-01	3.39158233E-02	8.70721134E-02	3.16234622E-02
24	1.63251455E-01	1.09206750E-02	9.34423749E-02	1.67325050E-02
25	1.69862355E-01	-1.69498835E-02	9.78418707E-02	2.79335586E-05
26	1.71529728E-0	-4.68245435E-02	9.94733429E-02	-1.80265241E-02
27	1.69835881E-	7 58091329E-02	9.75119620E-02	-3.68526302E-02
28	1.65623585E-0	-1.06053035E-01	9.42648411E-02	-5.49432734E-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	-1.84265674E-01	7.94168208E-02	-1.03457324E-01
32	1.29703770E-01	-2.05856326E-01	7.26917087E-02	-1.16966695E-01
33	1.17931938E-01	-2.24565247E-01	6.52551056E-02	-1.28768131E-01

ENTERING PROGRAM CHORDF 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 REAL	CHORD 1 IMAGINARY	CHORD 2 REAL	CHORD 2 IMAGINARY	CHORD 3 REAL	CHORD 3 IMAGINARY	CHORD 4 REAL	CHORD 4 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	-1.97083941E-01	1.27223759E-02	-1.97143415E-01	1.28083146E-02	-1.97483806E-01	1.30861618E-02
3	-3.24556639E-01	2.78274530E-02	-3.25551934E-01	2.98614676E-02	-3.27380349E-01	3.30453819E-02	-3.29647122E-01	3.56443884E-02
4	-4.48830232E-01	5.23722576E-02	-4.50871795E-01	5.63349495E-02	-4.54319883E-01	6.23455157E-02	-4.57878050E-01	6.67555593E-02
5	-5.69164048E-01	8.86699981E-02	-5.71621530E-01	9.34466986E-02	-5.75418680E-01	1.00118235E-01	-5.78361484E-01	1.04244901E-01
6	-6.84089498E-01	1.36746250E-01	-6.86020340E-01	1.40147900E-01	-6.88388640E-01	1.44566246E-01	-6.88319982E-01	1.45640959E-01
7	-7.91721792E-01	1.93055618E-01	-7.92141143E-01	1.93598327E-01	-7.91286080E-01	1.93003593E-01	-7.85936163E-01	1.88314745E-01
8	-8.90037439E-01	2.53205022E-01	-8.88123104E-01	2.49727622E-01	-8.82600909E-01	2.42173813E-01	-8.70275106E-01	2.29638882E-01
9	-9.77151748E-01	3.11676968E-01	-9.72384162E-01	3.03796575E-01	-9.61345816E-01	2.88568094E-01	-9.41206748E-01	2.67146750E-01
10	-1.05159632E+00	3.62752835E-01	-1.04383355E+00	3.50958405E-01	-1.02714544E+00	3.28743423E-01	-9.99328279E-01	2.98691636E-01
11	-1.11259656E+00	4.01236152E-01	-1.10208434E+00	3.86820042E-01	-1.08032556E+00	3.59640745E-01	-1.04588654E+00	3.22605876E-01
12	-1.16034915E+00	4.23175879E-01	-1.14766593E+00	4.08003403E-01	-1.12200226E+00	3.78903116E-01	-1.08270043E+00	3.37860004E-01
13	-1.19629959E+00	4.26589686E-01	-1.18223663E+00	4.12706675E-01	-1.15417112E+00	3.85193856E-01	-1.11208329E+00	3.44221897E-01
14	-1.22341965E+00	4.12187232E-01	-1.20879613E+00	4.01265594E-01	-1.17979639E+00	3.78514702E-01	-1.13676531E+00	3.42415923E-01
15	-1.24648491E+00	3.84093449E-01	-1.23189805E+00	3.76714729E-01	-1.20290018E+00	3.60523966E-01	-1.15981590E+00	3.34282086E-01

ROW	CHORD 5 REAL	CHORD 5 IMAGINARY	CHORD 6 REAL	CHORD 6 IMAGINARY	CHORD 7 REAL	CHORD 7 IMAGINARY	CHORD 8 REAL	CHORD 8 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	-1.98219650E-01	1.35987591E-02	-1.99178143E-01	1.42478878E-02	-1.99574854E-01	1.45287950E-02	-1.97359615E-01	1.38503823E-02
3	-3.31648431E-01	3.71963629E-02	-3.31782358E-01	3.75643401E-02	-3.27178757E-01	3.55904975E-02	-3.13350077E-01	3.07418372E-02
4	-4.59639168E-01	6.82277631E-02	-4.56277017E-01	6.65536327E-02	-4.42775937E-01	6.02536010E-02	-4.12652[illegible]8E-01	4.92501208E-02
5	-5.77356182E-01	1.03656555E-01	-5.67662406E-01	9.79173967E-02	-5.42778938E-01	8.57675835E-02	-4.95089647E-01	6.77768178E-02
6	-6.81620779E-01	1.40723786E-01	-6.63357301E-01	1.29082285E-01	-6.25897784E-01	1.10151881E-01	-5.61656328E-01	8.51798730E-02
7	-7.71638657E-01	1.76996946E-01	-7.42763071E-01	1.58127810E-01	-6.92670670E-01	1.32085035E-01	-6.14222480E-01	1.00704298E-01
8	-8.46727012E-01	2.10418947E-01	-8.06827782E-01	1.83714187E-01	-7.44994650E-01	1.50793840E-01	-6.55242880E-01	1.13912876E-01
9	-9.08041773E-01	2.39357113E-01	-8.57610296E-01	2.05010171E-01	-7.85656331E-01	1.65942494E-01	-6.87464203E-01	1.24616871E-01
10	-9.57304800E-01	2.62652162E-01	-8.97844381E-01	2.21620898E-01	-8.17862559E-01	1.77521739E-01	-7.13632526E-01	1.32806731E-01
11	-9.96731103E-01	2.79667193E-01	-9.30502807E-01	2.33515726E-01	-8.44771107E-01	1.85738019E-01	-7.36200823E-01	1.38582793E-01
12	-1.02875604E+00	2.90336672E-01	-9.58361449E-01	2.40956070E-01	-8.69021374E-01	1.90902619E-01	-7.57036476E-01	1.42085996E-01
13	-1.05576254E+00	2.95215416E-01	-9.83563396E-01	2.44423251E-01	-8.92265063E-01	1.93320817E-01	-7.77128768E-01	1.43428578E-01
14	-1.07980829E+00	2.95527578E-01	-1.00718305E+00	2.44546327E-01	-9.14696881E-01	1.93181031E-01	-7.96296387E-01	1.42624789E-01
15	-1.10235295E+00	2.93215636E-01	-1.02879021E+00	2.42029937E-01	-9.34585222E-01	1.90443967E-01	-8.12894929E-01	1.39521595E-01

ROW	CHORD 9 REAL	CHORD 9 IMAGINARY	CHORD 10 REAL	CHORD 10 IMAGINARY

1	-6.63184769E-02	6.99825447E-04	-6.63184769E-02	6.99825447E-04
2	-1.87618875E-01	1.15968923E-02	-1.45345422E-01	6.43275503E-03
3	-2.82695229E-01	2.30677982E-02	-2.03047262E-01	1.20415075E-02
4	-3.57193181E-01	3.47372587E-02	-2.45865286E-01	1.76860369E-02
5	-4.15793190E-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	-3.26854662E-01	3.44972878E-02
8	-5.30491357E-01	7.62516302E-02	-3.45993044E-01	3.95090314E-02
9	-5.56265453E-01	8.37623829E-02	-3.62964003E-01	4.38330190E-02
10	-5.78331092E-01	8.96277688E-02	-3.77955130E-01	4.72265012E-02
11	-5.97836599E-01	9.36891743E-02	-3.90882892E-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	CHORD 9		CHORD 10	
	REAL	IMAGINARY	REAL	IMAGINARY
12	-6.14418108E-01	9.58535425E-02	-4.01612967E-01	5.04614257E-02
13	-6.29292840E-01	9.61020248E-02	-4.10180572E-01	5.01163937E-02
14	-6.42352395E-01	9.44986311E-02	-4.11010800E-01	4.85269414E-02
15	-6.53756028E-01	9.11988815E-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	5.97433780E-02	2.31086510E-02	5.91216173E-02	2.30340104E-02	5.78806629E-02	2.28405074E-02	5.57754206E-02	2.27629510E-02
20	1.37159179E-01	1.70348754E-02	1.36236081E-01	1.76367243E-02	1.34671664E-01	1.96073137E-02	1.31929613E-01	2.32984408E-02
21	2.01505110E-01	-7.60417390E-03	2.01017365E-01	-5.85533567E-03	2.00069169E-01	-1.51753043E-03	1.97212464E-01	5.96041923E-03
22	2.57299536E-01	-4.73861893E-02	2.57146395E-01	-4.41446737E-02	2.56433384E-01	-3.71176455E-02	2.52884636E-01	-2.57274212E-02
23	3.06928959E-01	-9.88213134E-02	3.06545565E-01	-9.38770040E-02	3.04984542E-01	-8.36683105E-02	2.99613792E-01	-6.84432957E-02
24	3.51082130E-01	-1.58445810E-01	3.49737711E-01	-1.51735395E-01	3.46053979E-01	-1.38584073E-01	3.37625229E-01	-1.19090063E-01
25	3.89184169E-01	-2.22915939E-01	3.86217079E-01	-2.14519415E-01	3.79335215E-01	-1.98266359E-01	3.66852495E-01	-1.74817963E-01
26	4.19830679E-01	-2.89101822E-01	4.14818298E-01	-2.79230274E-01	4.04135036E-01	-2.60151082E-01	3.87088015E-01	-2.33047351E-01
27	4.41221865E-01	-3.54181322E-01	4.34085351E-01	-3.43154958E-01	4.19624570E-01	-3.21756255E-01	3.98133717E-01	-2.91491434E-01
28	4.51596648E-01	-4.15733910E-01	4.42640543E-01	-4.03950425E-01	4.25090365E-01	-3.80929597E-01	3.99951651E-01	-3.48179012E-01
29	4.49666781E-01	-4.71834540E-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	4.08708980E-01	-5.63020373E-01	3.99181280E-01	-5.51446654E-01	3.81214832E-01	-5.28281265E-01	3.55224362E-01	-4.93201984E-01
32	3.73375767E-01	-5.97577737E-01	3.65592568E-01	-5.86638113E-01	3.50547409E-01	-5.64464988E-01	3.28226101E-01	-5.30259066E-01
33	3.33995581E-01	-6.25815209E-01	3.28492378E-01	-6.15478988E-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	2.02575324E-02
20	1.26981680E-01	2.73878696E-02	1.19055130E-01	3.17193941E-02	1.07386336E-01	3.62411181E-02	9.05652650E-02	3.94935158E-02
21	1.90709405E-01	1.52077630E-02	1.79339232E-01	2.55213333E-02	1.61652194E-01	3.59942943E-02	1.35933077E-01	4.41803853E-02
22	2.44310136E-01	-1.10564228E-02	2.29147868E-01	5.65873036E-03	2.05466085E-01	2.23719269E-02	1.71718108E-01	3.59108618E-02
23	2.88048902E-01	-4.83032874E-02	2.68525844E-01	-2.54842551E-02	2.38994491E-01	-2.59391333E-03	1.98340599E-01	1.66888498E-02
24	3.22128451E-01	-9.39973708E-02	2.97771442E-01	-6.53737189E-02	2.62756306E-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	2.31182925E-01	-8.32997251E-02
27	3.68773935E-01	-2.54287549E-01	3.30894603E-01	-2.12070002E-01	2.84421943E-01	-1.67449677E-01	2.29874905E-01	-1.22759653E-01
28	3.67113213E-01	-3.07987563E-01	3.26654694E-01	-2.62168500E-01	2.78862268E-01	-2.12779839E-01	2.24293890E-01	-1.61675472E-01
29	3.57987495E-01	-3.58607447E-01	3.16566656E-01	-3.09654841E-01	2.68924344E-01	-2.55666791E-01	2.15495789E-01	-1.98390563E-01
30	3.42494641E-01	-4.05499490E-01	3.01843554E-01	-3.53230887E-01	2.55746150E-01	-2.94832997E-01	2.04402619E-01	-2.31775339E-01
31	3.22093156E-01	-4.47261290E-01	2.83749763E-01	-3.92160164E-01	2.40319615E-01	-3.29659833E-01	1.91733984E-01	-2.61361067E-01
32	2.98662610E-01	-4.83786521E-01	2.63572069E-01	-4.26369497E-01	2.23419408E-01	-3.60325373E-01	1.77938549E-01	-2.87474081E-01
33	2.74964054E-01	-5.15315699E-01	2.42590774E-01	-4.56550636E-01	2.05531727E-01	-3.87942162E-01	1.63125515E-01	-3.11369708E-01

ROW	CHORD 9		CHORD 10	
	REAL	IMAGINARY	REAL	IMAGINARY

19	2.60450060E-02	1.90552832E-02	1.70750952E-02	1.84678707E-02
20	6.80787278E-02	4.04667885E-02	3.94010563E-02	3.28520168E-02
21	1.02114556E-01	4.79949451E-02	5.84053396E-02	3.84330200E-02
22	1.28515746E-01	4.36974129E-02	7.37037680E-02	3.65091120E-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	1.68001598E-01	-1.80701021E-02	9.73609899E-02	-4.11863254E-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	1.57595767E-01	-1.37382106E-01	9.02951685E-02	-7.39627241E-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

PAGE CONTINUED

ROW	CHORD 9 REAL	CHORD 9 IMAGINARY	CHORD 10 REAL	CHORD 10 IMAGINARY
30	1.49025425E-01	-1.63454682E-01	8.49136605E-02	-9.01922222E-02
31	1.39286652E-01	-1.86678961E-01	7.89026902E-02	-1.04702691E-01
32	1.28702570E-01	-2.07353039E-01	7.24345246E-02	-1.17669622E-01
33	1.17367156E-01	-2.26460981E-01	6.54684096E-02	-1.29692531E-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 RED.FREQ. = .50000)

MODE SHAPE 2

ROW	CHORD 1 REAL	CHORD 1 IMAGINARY	CHORD 2 REAL	CHORD 2 IMAGINARY	CHORD 3 REAL	CHORD 3 IMAGINARY	CHORD 4 REAL	CHORD 4 IMAGINARY
1	-3.97215651E-01	1.35929082E-02	-3.97164069E-01	1.36028822E-02	-3.97253729E-01	1.37324396E-02	-3.97766888E-01	1.41211589E-02
2	-3.90580254E-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	-3.85585731E-01	3.31932352E-02	-3.91012651E-01	4.19421057E-02	-3.96123532E-01	4.79751000E-02
4	-3.73996770E-01	4.71420761E-02	-3.76597369E-01	5.07711631E-02	-3.80166406E-01	5.56841431E-02	-3.81626565E-01	5.76313589E-02
5	-3.63553687E-01	7.02821740E-02	-3.63844440E-01	6.91906240E-02	-3.62883753E-01	6.64105472E-02	-3.57828774E-01	6.10860666E-02
6	-3.49193047E-01	8.86567527E-02	-3.46460615E-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	-3.12086371E-01	6.80203355E-02	-2.93139145E-01	4.80380167E-02
8	-3.04866679E-01	8.98246685E-02	-2.96699338E-01	7.73177985E-02	-2.80592080E-01	5.58088150E-02	-2.57043102E-01	3.18162087E-02
9	-2.74726874E-01	6.74343668E-02	-2.65121991E-01	5.53727320E-02	-2.46765890E-01	3.43740157E-02	-2.21269653E-01	9.98026936E-03
10	-2.40460015E-01	2.98559180E-02	-2.30626214E-01	2.07792311E-02	-2.12243053E-01	4.43150997E-03	-1.87679892E-01	-1.63250497E-02
11	-2.04074674E-01	-2.01684892E-02	-1.95215208E-01	-2.42098643E-02	-1.78966105E-01	-3.24139725E-02	-1.57948840E-01	-4.55401180E-02
12	-1.68504655E-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	-1.28381291E-01	-1.25647318E-01	-1.16002669E-01	-1.15928051E-01	-1.04340022E-01
14	-1.16875946E-01	-1.86407452E-01	-1.14993963E-01	-1.75139513E-01	-1.11313294E-01	-1.55170897E-01	-1.06201211E-01	-1.28661360E-01
15	-1.07953723E-01	-2.09354434E-01	-1.07326390E-01	-1.97213450E-01	-1.05712941E-01	-1.74534127E-01	-1.02928376E-01	-1.40506033E-01

ROW	CHORD 5 REAL	CHORD 5 IMAGINARY	CHORD 6 REAL	CHORD 6 IMAGINARY	CHORD 7 REAL	CHORD 7 IMAGINARY	CHORD 8 REAL	CHORD 8 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	-4.01359833E-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	4.53579540E-02	-3.31054862E-01	3.13672096E-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	-2.66305195E-01	2.79055432E-02	-2.32102668E-01	8.08438473E-03	-1.92753785E-01	-7.99700767E-03	-1.51157465E-01	-1.81056155E-02
8	-2.26677329E-01	9.33878777E-03	-1.91510125E-01	-1.00049594E-02	-1.55260544E-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	-1.59969413E-01	-3.49607483E-02	-1.32051684E-01	-4.68106982E-02	-1.06871052E-01	-5.19433781E-02	-8.67538918E-02	-5.03064332E-02
11	-1.35683519E-01	-5.79371412E-02	-1.14584485E-01	-6.39014123E-02	-9.56986178E-02	-6.43046753E-02	-7.92921967E-02	-5.96310642E-02
12	-1.18026905E-01	-7.94357765E-02	-1.04087455E-01	-7.93924315E-02	-9.06900945E-02	-7.54706392E-02	-7.59097972E-02	-6.83983531E-02
13	-1.06485694E-01	-9.77506678E-02	-9.80436516E-02	-9.29438243E-02	-8.81905000E-02	-8.57916709E-02	-7.35294000E-02	-7.69006154E-02
14	-9.97904451E-02	-1.10995609E-01	-9.26366271E-02	-1.04326361E-01	-8.31181436E-02	-9.58067036E-02	-6.81819961E-02	-8.55196361E-02
15	-9.72982854E-02	-1.17206063E-01	-8.93510578E-02	-1.10466222E-01	-7.90100018E-02	-1.01711083E-01	-6.39986477E-02	-9.06459757E-02

ROW	CHORD 9 REAL	CHORD 9 IMAGINARY	CHORD 10 REAL	CHORD 10 IMAGINARY

1	-3.82894944E-01	1.19061519E-02	-3.19165317E-01	4.12092196E-03
2	-3.27339916E-01	1.49591002E-02	-2.06769675E-01	2.56368684E-03
3	-2.57949665E-01	6.61589198E-03	-1.52743548E-01	-3.33976552E-05
4	-2.04126452E-01	-7.93463293E-04	-1.15740509E-01	-7.48498569E-03
5	-1.63083742E-01	-7.70889395E-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	-6.54453288E-02	-1.69136809E-02
8	-9.28178767E-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	-2.92779668E-02
11	-6.37721413E-02	-5.03679530E-02	-4.06138234E-02	-3.42114571E-02

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

WING BOX LIFTS

(MACH 1.200 RED.FREQ. = .50000)

MODE SHAPE 2

PAGE CONTINUED

ROW	CHORD 9 REAL	CHORD 9 IMAGINARY	CHORD 10 REAL	CHORD 10 IMAGINARY
12	-5.74596892E-02	-5.78043018E-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	-2.43881753E-02	-4.80176656E-02
15	-4.35031367E-02	-7.53247223E-02	-2.30696938E-02	-5.01546867E-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 REAL	CHORD 1 IMAGINARY	CHORD 2 REAL	CHORD 2 IMAGINARY	CHORD 3 REAL	CHORD 3 IMAGINARY	CHORD 4 REAL	CHORD 4 IMAGINARY
19	2.94530888E-01	6.64932824E-02	2.92212425E-01	6.72256014E-02	2.87999504E-01	6.97805643E-02	2.80699690E-01	7.50177029E-02
20	2.12010364E-01	-3.75857093E-02	2.12152216E-01	-2.99300274E-02	2.12396507E-01	-2.32539572E-02	2.10894522E-01	-1.21378085E-02
21	1.81878820E-01	-7.69678982E-02	1.82861775E-01	-7.30372254E-02	1.83714453E-01	-6.55091766E-02	1.81750516E-01	-5.41880214E-02
22	1.63671051E-01	-1.11765058E-01	1.63504215E-01	-1.06981511E-01	1.61077640E-01	-9.85051336E-02	1.56948552E-01	-8.68793571E-02
23	1.51264711E-01	-1.36735782E-01	1.48974362E-01	-1.31544559E-01	1.43494961E-01	-1.22467941E-01	1.34595581E-01	-1.10788099E-01
24	1.39848976E-01	-1.51971474E-01	1.35282866E-01	-1.46901507E-01	1.25946265E-01	-1.37856149E-01	1.13275165E-01	-1.26603322E-01
25	1.25933932E-01	-1.58052932E-01	1.19564618E-01	-1.53584053E-01	1.07387125E-01	-1.45335640E-01	9.20497520E-02	-1.35111825E-01
26	1.07359962E-01	-1.55906944E-01	1.00067162E-01	-1.52443562E-01	8.67535949E-02	-1.45754519E-01	7.04629508E-02	-1.37183075E-01
27	8.33071320E-02	-1.46782870E-01	7.62591086E-02	-1.44614168E-01	5.37669719E-02	-1.40118005E-01	4.85418042E-02	-1.33754139E-01
28	5.43045808E-02	-1.32209235E-01	4.86385529E-02	-1.31475876E-01	3.89385581E-02	-1.29563324E-01	2.67990625E-02	-1.25814625E-01
29	2.22399040E-02	-1.13950313E-01	1.89414858E-02	-1.14617667E-01	1.35744127E-02	-1.15334602E-01	6.23545681E-03	-1.14391620E-01
30	-9.63145674E-03	-9.39627186E-02	-9.94979052E-03	-9.58005995E-02	-1.02198671E-02	-9.87577539E-02	-1.16580268E-02	-1.00534624E-01
31	-3.66768273E-02	-7.43519948E-02	-3.39782479E-02	-7.69209091E-02	-2.95344056E-02	-8.12153800E-02	-2.48978739E-02	-8.53004907E-02
32	-5.28769134E-02	-5.73292000E-02	-4.79037183E-02	-5.99731195E-02	-4.06502774E-02	-6.41216544E-02	-3.10057681E-02	-6.97363645E-02
33	-5.61542084E-02	-5.17396212E-02	-5.03133835E-02	-5.41092721E-02	-4.23210481E-02	-5.75739501E-02	-3.05372041E-02	-6.34422365E-02

ROW	CHORD 5 REAL	CHORD 5 IMAGINARY	CHORD 6 REAL	CHORD 6 IMAGINARY	CHORD 7 REAL	CHORD 7 IMAGINARY	CHORD 8 REAL	CHORD 8 IMAGINARY
19	2.68386751E-01	8.05239074E-02	2.50437100E-01	8.60194661E-02	2.24451482E-01	9.13762451E-02	1.88085403E-01	9.36321154E-02
20	2.05259062E-01	1.66351703E-03	1.94025376E-01	1.70330234E-02	1.75362866E-01	3.25539353E-02	1.47398590E-01	4.51215842E-02
21	1.75398526E-01	-3.88468145E-02	1.63418943E-01	-2.13540056E-02	1.44261356E-01	-4.74337293E-03	1.17924476E-01	8.19225354E-03
22	1.47847854E-01	-7.14361020E-02	1.33888003E-01	-5.39790289E-02	1.14360707E-01	-3.76273033E-02	9.04917936E-02	-2.42732383E-02
23	1.22153852E-01	-9.62724900E-02	1.06002320E-01	-8.02328620E-02	8.60276489E-02	-6.49522222E-02	6.58394130E-02	-5.11951439E-02
24	9.78944860E-02	-1.13683419E-01	8.02295785E-02	-9.97873870E-02	6.17836301E-02	-8.59526793E-02	4.44718777E-02	-7.18700745E-02
25	7.48839581E-02	-1.24149582E-01	5.69454839E-02	-1.12598442E-01	4.03186206E-02	-1.00250529E-01	2.66728087E-02	-8.59778501E-02
26	5.31377841E-02	-1.28298887E-01	3.64439884E-02	-1.18908713E-01	2.25048874E-02	-1.07862050E-01	1.25183069E-02	-9.35883565E-02
27	3.20778695E-02	-1.26900409E-01	1.89474208E-02	-1.19250620E-01	8.41077335E-03	-1.09205068E-01	1.89035383E-03	-9.51683930E-02
28	1.45375864E-02	-1.20858353E-01	4.61665732E-03	-1.14449211E-01	-2.08552511E-03	-1.05106077E-01	-5.30978532E-03	-9.15885276E-02
29	-1.23314891E-03	-1.11206012E-01	-6.43871601E-03	-1.05625048E-01	-9.28217953E-03	-9.68073552E-02	-1.01481604E-02	-8.41299473E-02
30	-1.35628041E-02	-9.90997224E-02	-1.41502400E-02	-9.41970986E-02	-1.36399516E-02	-8.59740934E-02	-1.26447343E-02	-7.44913147E-02
31	-2.13531442E-02	-8.58128246E-02	-1.84804193E-02	-8.18856275E-02	-1.57684149E-02	-7.47015105E-02	-1.37599806E-02	-6.47956139E-02
32	-2.32741850E-02	-7.27295204E-02	-1.94125597E-02	-7.07150828E-02	-1.64121772E-02	-6.55219762E-02	-1.43814821E-02	-5.75970089E-02
33	-2.11263100E-02	-6.76076427E-02	-1.76059184E-02	-6.67404818E-02	-1.51391694E-02	-6.27145807E-02	-1.35260067E-02	-5.57354758E-02

ROW	CHORD 9 REAL	CHORD 9 IMAGINARY	CHORD 10 REAL	CHORD 10 IMAGINARY

19	1.39991339E-01	9.23373999E-02	8.32942147E-02	7.90751510E-02
20	1.10632483E-01	5.04360547E-02	5.90224862E-02	3.40387006E-02
21	8.63127384E-02	1.50818051E-02	4.78699816E-02	1.13538123E-02
22	6.49424992E-02	-1.46314300E-02	3.67249047E-02	-7.66663250E-03
23	4.59843523E-02	-3.83719577E-02	2.62063325E-02	-2.28682566E-02
24	2.99723444E-02	-5.60410657E-02	1.67831047E-02	-3.42263744E-02
25	1.65201716E-02	-6.77771167E-02	8.77887623E-03	-4.18508766E-02
26	6.32936778E-03	-7.39596418E-02	2.37717083E-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	-4.55428279E-02
29	-9.56995025E-03	-6.66848728E-02	-7.37343156E-03	-4.21942758E-02

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

PAGE CONTINUED

ROW	CHORD 9 REAL	CHORD 9 IMAGINARY	CHORD 10 REAL	CHORD 10 IMAGINARY
30	-1.12565620E-02	-5.94152631E-02	-8.15558508E-03	-3.78511791E-02
31	-1.19699486E-02	-5.22505983E-02	-8.34271615E-03	-3.35334692E-02
32	-1.23095785E-02	-4.71033928E-02	-8.48597774E-03	-3.04298476E-02
33	-1.15314834E-02	-4.58758961E-02	-8.03437375E-03	-2.97035954E-02

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

ROW	CHORD 1 REAL	CHORD 1 IMAGINARY	CHORD 2 REAL	CHORD 2 IMAGINARY	CHORD 3 REAL	CHORD 3 IMAGINARY	CHORD 4 REAL	CHORD 4 IMAGINARY
1	-5.98825125E+00	2.04920801E-01	-5.98747362E+00	2.05071164E-01	-5.98882530E+00	2.07024316E-01	-5.99656146E+00	2.12884481E-01
2	-5.88821887E+00	3.19370186E-01	-5.91034918E+00	3.65649373E-01	-5.95253362E+00	4.37921402E-01	-6.00443990E+00	4.96476574E-01
3	-5.76260045E+00	4.11996497E-01	-5.81292361E+00	5.00406849E-01	-5.89473751E+00	6.32301034E-01	-5.97176693E+00	7.23251845E-01
4	-5.63821345E+00	7.10693537E-01	-5.67741895E+00	7.65404002E-01	-5.7312[illegible]1E+00	8.39470043E-01	-5.75323694E+00	8.68825426E-01
5	-5.46077804E+00	1.05954364E+00	-5.48516131E+00	1.04308790E+00	-5.47067841E+00	1.00117667E+00	-5.39447174E+00	9.20937256E-01
6	-5.26428326E+00	1.33655129E+00	-5.22309029E+00	1.24195800E+00	-5.12401908E+00	1.07324592E+00	-4.9371[illegible]835E+00	8.72996862E-01
7	-4.97167734E+00	1.45322477E+00	-4.88510511E+00	1.29625052E+00	-4.70487906E+00	1.02544514E+00	-4.41923889E+00	7.24200350E-01
8	-4.59603812E+00	1.35415783E+00	-4.47291082E+00	1.16560967E+00	-4.23008477E+00	8.41349542E-01	-3.87507056E+00	4.79647[illegible]93E-01
9	-4.14166346E+00	1.01661133E+00	-3.99686440E+00	8.34775343E-01	-3.72013576E+00	5.18207785E-01	-3.33576553E+00	1.50458222E-01
10	-3.62516156E+00	4.50094899E-01	-3.47682099E+00	3.13238695E-01	-3.19968441E+00	6.68075265E-02	-2.82938083E+00	-2.46109385E-01
11	-3.07654146E+00	-3.04051415E-01	-2.94297999E+00	-3.64977437E-01	-2.69801555E+00	-4.88659022E-01	-2.38116835E+00	-6.86543110E-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	-2.07652863E+00	-2.05323145E+00	-2.01150125E+00	-1.93542079E+00	-1.89420458E+00	-1.74880603E+00	-1.74768113E+00	-1.57298502E+00
14	-1.76197118E+00	-2.81019807E+00	-1.73359922E+00	-2.64032750E+00	-1.67811105E+00	-2.33928929E+00	-1.60104350E+00	-1.93964298E+00
15	-1.6274[illegible]361E+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 REAL	CHORD 5 IMAGINARY	CHORD 6 REAL	CHORD 6 IMAGINARY	CHORD 7 REAL	CHORD 7 IMAGINARY	CHORD 8 REAL	CHORD 8 IMAGINARY
1	-6.01328517E+00	2.24988964E-01	-6.03506913E+00	2.39741890E-01	-6.04408528E+00	2.46126143E-01	-5.99373893E+00	2.30707674E-01
2	-6.05072714E+00	5.30639394E-01	-6.05474952E+00	5.37557530E-01	-5.95054[illegible]1E+00	4.92099406E-01	-5.63523478E+00	3.85242180E-01
3	-5.99742878E+00	7.41589451E-01	-5.89978659E+00	6.88586205E-01	-5.58095202E+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	-5.20584973E+00	7.77239692E-01	-4.85398613E+00	5.65322453E-01	-4.29115966E+00	3.15781780E-01	-3.48863809E+00	7.02095513E-02
6	-4.62765987E+00	6.38943580E-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	1.21876686E-01	-2.90587265E+00	-1.20559427E-01	-2.27878453E+00	-2.72952424E-01
8	-3.41728932E+00	1.40787523E-01	-2.88712377E+00	-1.50830438E-01	-2.34064076E+00	-3.53633997E-01	-1.83631474E+00	-4.44348229E-01
9	-2.87397597E+00	-1.81805969E-01	-2.37762355E+00	-4.31380004E-01	-1.90625654E+00	-5.76969237E-01	-1.51490439E+00	-6.06985503E-01
10	-2.41162964E+00	-5.27055112E-01	-1.99075403E+00	-7.05697827E-01	-1.61114173E+00	-7.83075889E-01	-1.30786413E+00	-7.58434197E-01
11	-2.04550602E+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	-1.19753939E+00	-1.56917742E+00	-1.19688594E+00	-1.36720462E+00	-1.13776270E+00	-1.14438325E+00	-1.03114397E+00
13	-1.60533224E+00	-1.47364676E+00	-1.47806366E+00	-1.40118087E+00	-1.32952181E+00	-1.29335810E+00	-1.10849741E+00	-1.15932040E+00
14	-1.90439756E+00	-1.67332177E+00	-1.39654970E+00	-1.57277907E+00	-1.25305316E+00	-1.44434040E+00	-1.02788227E+00	-1.28925702E+00
15	-1.46679669E+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 REAL	CHORD 9 IMAGINARY	CHORD 10 REAL	CHORD 10 IMAGINARY

1	-5.77235647E+00	1.79491993E-01	-4.81159819E+00	6.21252350E-02
2	-4.93513642E+00	2.25516921E-01	-3.11717014E+00	3.86490333E-02
3	-3.88873752E+00	9.97383243E-02	-2.30269562E+00	-5.03486596E-02
4	-3.07732205E+00	-1.19619092E-02	-1.74485383E+00	-1.12840405E-01
5	-2.45857992E+00	-1.16215949E-01	-1.37626681E+00	-[illegible]61868633E-01
6	-1.99446176E+00	-2.17948301E-01	-1.13940374E+00	-2.07065357E-01
7	-1.65119206E+00	-3.20516436E-01	-9.86625455E-01	-2.5498333[illegible]-01
8	-1.39928214E+00	-4.25851426E-01	-8.80228941E-01	-3.09428429E-01
9	-1.21354324E+00	-5.34598571E-01	-7.92491816E-01	-3.71791781E-01
10	-1.07309954E+00	-6.46258032E-01	-7.05716843E-01	-4.41381956E-01
11	-9.61401203E-01	-7.59325461E-01	-6.12276425E-01	-5.15757121E-01

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

PAGE CONTINUED

ROW	CHORD 9 REAL	CHORD 9 IMAGINARY	CHORD 10 REAL	CHORD 10 IMAGINARY
12	-6.66237408E-01	-8.71432638E-01	-5.14657102E-01	-5.91057198E-01
13	-7.79749415E-01	-9.79480093E-01	-4.25504049E-01	-6.62336031E-01
14	-6.98443591E-01	-1.07981775E+00	-3.67665577E-01	-7.23893546E-01
15	-6.55834462E-01	-1.13856292E+00	-3.47780765E-01	-7.56110352E-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 REAL	CHORD 1 IMAGINARY	CHORD 2 REAL	CHORD 2 IMAGINARY	CHORD 3 REAL	CHORD 3 IMAGINARY	CHORD 4 REAL	CHORD 4 IMAGINARY
19	4.44022021E+00	1.00242395E+00	4.40526807E+00	1.01346407E+00	4.34175589E+00	1.05198159E+00	4.23170705E+00	1.13093442E+00
20	3.19617650E+00	-4.91248051E-01	3.19831500E+00	-4.51212140E-01	3.20199782E+00	-3.50566595E-01	3.17935455E+00	-1.82984348E-01
21	2.74192038E+00	-1.16033472E+00	2.75674499E+00	-1.10107760E+00	2.76959959E+00	-9.87587996E-01	2.73999212E+00	-8.16915159E-01
22	2.46743394E+00	-1.68522316E+00	2.46491879E+00	-1.61280696E+00	2.44039725E+00	-1.48502076E+00	2.36608845E+00	-1.30975559E+00
23	2.28040132E+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	-1.90861689E+00
25	1.89852545E+00	-2.38273760E+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	1.25590227E+00	-2.21283502E+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	7.33253774E-01	-1.98207341E+00	5.87020820E-01	-1.95324060E+00	4.04011068E-01	-1.89672684E+00
29	3.35279169E-01	-1.71786560E+00	2.85553642E-01	-1.72792635E+00	2.04641972E-01	-1.73873454E+00	9.40030483E-02	-1.72451856E+00
30	-1.45199674E-01	-1.41654128E+00	-1.49998736E-01	-1.44424837E+00	-1.54070293E-01	-1.48882916E+00	-1.75751366E-01	-1.51561648E+00
31	-5.52923975E-01	-1.12089850E+00	-5.12241361E-01	-1.15962634E+00	-4.45247917E-01	-1.22436792E+00	-3.75349571E-01	-1.28595328E+00
32	-7.97149463E-01	-8.64270209E-01	-7.22175725E-01	-9.04128795E-01	-6.12825989E-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 REAL	CHORD 5 IMAGINARY	CHORD 6 REAL	CHORD 6 IMAGINARY	CHORD 7 REAL	CHORD 7 IMAGINARY	CHORD 8 REAL	CHORD 8 IMAGINARY
19	4.05060517E+00	1.21394358E+00	3.77548133E+00	1.29679224E+00	3.38373340E+00	1.37754873E+00	2.83549414E+00	1.41155725E+00
20	3.09439679E+00	2.50784628E-02	2.92504260E+00	2.56782490E-01	2.64369497E+00	4.90769040E-01	2.22211736E+00	6.80233476E-01
21	2.64362925E+00	-5.85637763E-01	2.46363327E+00	-3.21923746E-01	2.17482177E+00	-7.15090375E-02	1.77777836E+00	1.23502869E-01
22	2.22889027E+00	-1.07693976E+00	2.01843759E+00	-8.13764480E-01	1.72405253E+00	-5.67252940E-01	1.36421512E+00	-3.65932836E-01
23	1.84153861E+00	-1.45135240E+00	1.59804519E+00	-1.20955591E+00	1.30596094E+00	-9.79191592E-01	9.92566498E-01	-7.71795834E-01
24	1.47581491E+00	-1.71384202E+00	1.20950640E+00	-1.50435146E+00	9.31423270E-01	-1.29578539E+00	6.70438782E-01	-1.08348213E+00
25	1.12891814E+00	-1.87162538E+00	8.58485470E-01	-1.69748539E+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	3.39273944E-01	-1.62608159E+00	1.88720577E-01	-1.41089756E+00
27	4.95652531E-01	-1.91309564E+00	2.85643117E-01	-1.79777074E+00	1.26797179E-01	-1.64632835E+00	2.84981562E-02	-1.43471751E+00
28	2.19162361E-01	-1.82200623E+00	6.95987278E-02	-1.72538677E+00	-3.14404740E-02	-1.58453372E+00	-8.30631390E-02	-1.38074900E+00
29	-1.85904343E-02	-1.67649371E+00	-9.70672961E-02	-1.59235751E+00	-1.39934122E-01	-1.45942579E+00	-1.52989275E-01	-1.26830668E+00
30	-2.04466964E-01	-1.49398453E+00	-2.13322894E-01	-1.42007469E+00	-2.05630007E-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	-1.09644028E+00	-2.92655348E-01	-1.06606999E+00	-2.47422880E-01	-9.87780957E-01	-2.16809001E-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 REAL	CHORD 9 IMAGINARY	CHORD 10 REAL	CHORD 10 IMAGINARY

19	2.11044865E+00	1.39203868E+00	1.25570753E+00	1.19210276E+00
20	1.66784743E+00	7.60352133E-01	8.89797459E-01	5.13152719E-01
21	1.30121349E+00	2.27366766E-01	7.21667127E-01	1.71165161E-01
22	9.73014786E-01	-2.20577107E-01	5.53648770E-01	-1.15578834E-01
23	6.87209967E-01	-5.78479028E-01	3.95075328E-01	-3.44751940E-01
24	4.45819867E-01	-8.44850854E-01	2.53014823E-01	-5.15982008E-01
25	2.49050958E-01	-1.02177848E+00	1.32346539E-01	-6.30925704E-01
26	9.54188094E-02	-1.11498355E+00	3.58371987E-02	-6.93342154E-01
27	-1.81284612E-02	-1.13388519E+00	-3.57828613E-02	-7.09166585E-01
28	-9.60867332E-02	-1.09166169E+00	-8.37435136E-02	-6.86583964E-01
29	-1.44272428E-01	-1.00531228E+00	-1.11158663E-01	-6.36102642E-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 REAL	CHORD 9 IMAGINARY	CHORD 10 REAL	CHORD 10 IMAGINARY
30	-1.69699057E-01	-8.95718793E-01	-1.22950071E-01	-5.70627996E-01
31	-1.80453766E-01	-7.87707407E-01	-1.25771178E-01	-5.05536069E-01
32	-1.85573878E-01	-7.10110365E-01	-1.27930928E-01	-4.58747213E-01
33	-1.73843653E-01	-6.91609156E-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	-3.98859565E+00	-1.11155716E-01	-3.83484957E+00	-1.49584478E-01
5	-3.62845563E+00	-2.25619564E-01	-3.36649421E+00	-3.23848097E-01	-3.03655545E+00	-4.00020976E-01	-2.61857201E+00	-4.43659370E-01
9	-2.08452383E+00	-4.39411750E-01	-1.33493772E+00	-3.35515650E-01				

TOTAL LIFT - [illegible]

-3.21255904E+01 -2.59287395E+00

SECTION LIFTS
TAIL

CHORD	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY	REAL	IMAGINARY
1	1.38101091E+00	-1.42783847E+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	-8.77485637E-01	6.25322874E-01	-7.03464991E-01
9	4.35271594E-01	-5.11882055E-01	2.32736505E-01	-2.94431576E-01				

TOTAL LIFT - TAIL

9.44466186E+00 -1.00676363E+01

TOTAL LIFT

-2.26009286E+01 -1.26605102E+01

GENERALIZED FORCES

(MACH 1.200   RED. FREQ. = .50000)

WT. FUNCT	VELOCITY POTENTIAL MODE 1 REAL	VELOCITY POTENTIAL MODE 1 IMAGINARY	VELOCITY POTENTIAL MODE 2 REAL	VELOCITY POTENTIAL MODE 2 IMAGINARY
1	-3.65294231E+00	-1.49248849E+01	-3.21255904E+01	-2.59267395E+00
2	-1.31502372E+00	-4.98711150E+00	-1.14917114E+01	-3.45896395E+00

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

WT.
FUNCT

VELOCITY POTENTIAL MODES

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

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

PAGE CONTINUED

WT. FUNCT	VELOCITY POTENTIAL MODES	
	1	2
1	2.98497697E+00	5.18574791E-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 SMOOTHING.

****

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                IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC          FILES   00003
      COMMON /ARRAYS/ KBXCDW,LBXCDW,LBCXC,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 ITPE1,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,B1?,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,NPLKRN,NSPATK,NROWEA            KERN    00002
      COMMON /KVAL  / IKVAL,XKVAL(20), XKS(20)                          KVAL    00002
      COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUTP,         FILES   00002
     1                IOUFSP,MODESC,IVPSC,IGEOSC,IWYFSC,IAICSC          FILES   00003
      COMMON /IOCONT/ OPLAIC,OSPAIC,WTGEOM,WTGNAF,WTSL,WTDL,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 /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,MTYPEW,MTYPET                            MODCOM  00002
      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
      COMMON /SAMPLW/ ISMPLW,ICHORD(10),IBOXF(10),IBOXL(10),ZLOC(10)    SAMPLW  00002
      COMMON /PLANXY/ NWLE,NWTE,NTLE,NTTE, XWLE(10),YWLE(10),           PLANXY  00002
     1                XWTE(10),YWTE(10), XTLE(10),YTLE(10),             PLANXY  00003
     2                XTTE(10),YTTE(10)                                 PLANXY  00004
      COMMON /CHECKPR/ DPPCPR,GEOCPR,MODCPR,AICCPR,NWSCPR,SMCPR,GAFCPR  CHECKPR 00002
      LOGICAL  DPPCPR, GEOCPR, MODCPR, AICCPR,NWSCPR,  SMCPR, GAFCPR    CHECKPR 00003
      COMMON /RWBUFF/ BFCODE,IBFCNT,  BUFF(3280)                        RWBUFF  00002
      DATA            BFCODE,IBFCNT  / 8HBUFFSIZE,3280  /               BCSCNA  00001
      DATA  TEV147 /6HAFMBOX /                                          FTNX1   00002
      DATA  PREVEX /10HNEVER EXEC /                                     FTNX1   00003
      DATA  EXEC   /10HAFMBOX EXC /                                     FTNX1   00004
      CALL RDINIT                                                       CONTROL 00030
    1 CONTINUE                                                          CONTROL 00031
      PROG = 6HDATAPP                                                   CONTROL 00032
```

```
      CALL DTIME(CPTIME,PPTIME)
      WRITE (NT6,6005) PROG,CPTIME,PPTIME
 6005 FORMAT(1H0,10X,*ENTERING PROGRAM *,A6,*  CURRENT ELAPSED TIME IS *
     1 *  CP =*,F8.3,*,   PP =*,F8.3 )
      CALL OVERLAY(TEV147,1,1,0)
      PARM(3) = XMACH
      IPARM(5)= NKVALS
      IF (OMACH .EQ. XMACH .AND. PRVGEOM) GO TO 100
C
      PROG = 6HGEOM
      CALL DTIME(CPTIME,PPTIME)
      WRITE (NT6,6005) PROG,CPTIME,PPTIME
C         COMPUTE GEOMETRY SECTION
      CALL OVERLAY(TEV147,1,2,0)
C
C         READ MODE SHAPES, PLACE IN INTERNAL STORAGE CONVENTION,
C         STORE ON SCRATCH FILE.  COMPUTE AND STORE OPTIONAL
C         THICKNESS SLOPE FUNCTIONS
C
  100 CONTINUE
      IF (NKVALS .LE. 0) GO TO 810
      PROG = 6HMODES
      CALL DTIME(CPTIME,PPTIME)
      WRITE (NT6,6005) PROG,CPTIME,PPTIME
      CALL OVERLAY(TEV147,1,3,0)
C
C         SPACE OUTPUT TAPE IF DESIRED
      IF (NOUTP .LE. 0) GO TO 200
      IF (PREVEX .NE. EXEC) REWIND NOUTP
C         FILE SPACING A FUNCTION OF INSTALLATION CAPABILITIES
  200 CONTINUE
C
C         LOOP ON NUMBER OF K1 VALUES THRU KERNELS, DOWNWASHES AND
C         AIR FORCES
      DO 800 IKVAL = 1,NKVALS
C
C         CALL KERNEL ROUTINES
      PROG = 6HAIC
      CALL DTIME(CPTIME,PPTIME)
      WRITE (NT6,6005) PROG,CPTIME,PPTIME
      CALL OVERLAY(TEV147,1,4,0)
C
      PROG = 6HVELPOT
      CALL DTIME(CPTIME,PPTIME)
      WRITE (NT6,6005) PROG,CPTIME,PPTIME
C         CALL DOWNWASH AND VELOCITY POTENTIAL ROUTINES.
      CALL OVERLAY(TEV147,1,5,0)
C
      IF(.NOT.SMOOTH) GO TO 600
      IF(CRDFIT) GO TO 600
C
      PROG = 6HSMOOTH
      CALL DTIME(CPTIME,PPTIME)
      WRITE (NT6,6005) PROG,CPTIME,PPTIME
      CALL OVERLAY(TEV147,1,6,0)
      GO TO 700
C
```

```
  600 CONTINUE
      IF(.NOT.CRDFIT) GO TO 700
C
      PROG = 6HCHORDF
      CALL DTIME(CPTIME,PPTIME)
      WRITE (NT6,6005) PROG,CPTIME,PPTIME
      CALL OVERLAY(TEV147,1,7,0)
C
  700 CONTINUE
      PROG = 6HFORCES
      CALL DTIME(CPTIME,PPTIME)
      WRITE (NT6,6005) PROG,CPTIME,PPTIME
      CALL OVERLAY(TEV147,1,8,0)
      IF(.NOT.(SMOOTH.OR.CRDFIT)) GO TO 800
      NIVPSC = IAICSC
      IAICSC = IVPSC
      IVPSC  = NIVPSC
      CALL DTIME(CPTIME,PPTIME)
      WRITE (NT6,6005) PROG,CPTIME,PPTIME
      WRITE (NT6,6010)
 6010 FORMAT(1H0,5X,90(1H*),///6X,*PROGRAM FORCES IS BEING RECALLED TO C
     1OMPUTE AIR FORCES WITHOUT SMOOTHING.*//6X,90(1H*) )
      CALL OVERLAY(TEV147,1,8,6HRECALL)
C
  800 CONTINUE
C          END OF LOOP ON REDUCED FREQUENCIES
C
  810 CONTINUE
      IF (NOUTP .GT. 0) REWIND NOUTP
      IF (NPLAIC .GT. 0) REWIND NPLAIC
      IF (NSPAIC .GT. 0) REWIND NSPAIC
      CALL DTIME(CPTIME,PPTIME)
      WRITE (NT6,6006) CPTIME,PPTIME
 6006 FORMAT(1H0,10X*PROGRAM COMPLETED *,6X,*  CURRENT ELAPSED TIME IS *
     1 *  CP =*,F8.3,*,  PP =*,F8.3 )
      READ(5,8005) LINK,L1,L2
 8005 FORMAT(A6,4X,2I10)
C
C          DETERMINE IF ANOTHER CYCLE IS WANTED.
C          IF L1 = -1,  RECYCLE
      IF(L1.EQ.-1) GO TO 1
C          IF L1 = -2,  RETURN TO CALLING PROGRAM
      IF(L1.EQ.-2) RETURN
C          IF L1 =  0,  CALL EXIT
      IF(L1.EQ.0) CALL EXIT
C          IF L1 = POS.  CALL OVERLAY
      IF(L1.GT.0) CALL OVERLAY(LINK,L1,L2,0)
      END
```

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

```
      SUBROUTINE RDINIT                                          RDINIT  00002
      COMMON /TAPEIO/ NFS,NMS,LS,NMR,ID(20),NID,ITYPE,LRS,LWS,M,N,  RDINIT  00003
     1                PARM(10), IRR                              RDINIT  00004
      DIMENSION IA(1)                                            RDINIT  00005
      EQUIVALENCE (IA,NFS)                                       RDINIT  00006
      DO 10 I= 1,41                                              RDINIT  00007
      IA(I) = 0                                                  RDINIT  00008
   10 CONTINUE                                                   RDINIT  00009
      NID = 20                                                   RDINIT  00010
      RETURN                                                     RDINIT  00011
      END                                                        RDINIT  00012
```

```
      SUBROUTINE DTIME(CPTIME,PPTIME)                                    DTIME   00002
C        ROUTINE TO INTERROGATE THE SYSTEM CLOCKS                        DTIME   00003
      PPTIME = 0                                                         DTIME   00004
      CALL SECOND(CPTIME)                                                DTIME   00005
      RETURN                                                             DTIME   00006
      END                                                                DTIME   00007
```

```
      SUBROUTINE READMX(INFILE,MXREAD,RANDIN, NFS, NMS, LS, NMR, K, NID,
     1                  ID, ITYPE, LRS, A, M, N, PARM, IRR )
C
C         ROUTINE TO READ A MATRIX ON TAPE OR DISK FILE.
C       THIS VERSION WILL WORK WITH SEQUENTIAL FILES ONLY.
C       SOME VARIABLES ARE PASSED FOR RANDOM OPERATION BUT
C       ARE NOT CURRENTLY USED.
C
C     INPUT -
C       INFILE - TAPE NUMBER OR LEFT ADJUSTED FILE NAME
C       MXREAD - .T. SNAPT FORMAT       (NOT USED)
C                .F. TEL001 FORMAT
C       RANDIN - .T. RANDOM FILE        (NOT USED)
C                .F. SEQUENTIAL FILE
C       NFS    - NUMBER OF FILES TO SPACE
C       NMS    - NUMBER OF MATRICES TO SPACE
C       LS     - LEVEL NUMBER TO SPACE  (NOT USED)
C       NMR    - IDENTIFIER  (NAME OR NUMBER)  (NOT USED)
C       K      - ROW DIMENSION OF ARRAY A
C                (IF K=0, MATRIX WILL BE LEFT IN /RWBUFF/. IT WILL
C                 BE STORED AS A ROW-WISE MATRIX, NOT AS A FORTRAN
C                 COLUMN-WISE MATRIX. M-ROWS AND N-COLUMNS )
C       NID    - NUMBER OF WORDS AVAILABLE IN ID ARRAY
C     IN/OUT
C       ID     - IDENTIFICATION ARRAY
C       ITYPE  - REAL,DIAGONAL,NULL,MIXED,COMPLEX
C     OUTPUT -
C       LRS    - LEVEL NUMBER OF MATRIX READ    (NOT USED)
C       A      - ARRAY CONTAINING MATRIX
C       M      - ROW DIMENSION OF MATRIX
C       N      - COLUMN DIMENSION OF MATRIX
C       PARM   - ARRAY OF NUMERICAL PARAMETERS STORED WITH THE MATRIX
C       IRR    -
C                0, NO ERROR
C                1, MATRIX SPACING IS NEGATIVE
C                2, FILE SPACING IS NEGATIVE
C                4, MATRIX DIMENSIONS ILLEGAL
C                5, M .GT. K
C                1500 + I, ENCOUNTERED ' ' ' AFTER MATRIX I WHILE
C                           SKIPPING MATRICES.
C
      DIMENSION ID(1), A(K,1), PARM(10), B(16)
C
      COMMON /RWBUFF/ BFCODE,IBFCNT,    BUFF(3280)
C
      DIMENSION IBUFF(2500),  IPARM(10),IB(16)
      EQUIVALENCE (BUFF,IBUFF),(B,IB)
C
      LOGICAL   MXREAD,RANDIN
      IRR = 0
C
C         DO FILE SPACING
C
      IF(NFS) 215,230,220
  215 CONTINUE
      IRR = 2
      GO TO 1000
```

```
  220 CONTINUE
      DO 225 I=1,NFS
  222 CONTINUE
      BUFFER IN (INFILE,1) (BUFF(1),BUFF(IBFCNT))
  221 CONTINUE
      IF(UNIT,INFILE) 221,222,225
  225 CONTINUE
  230 CONTINUE
C
C           DO MATRIX SPACING
C
      IF(NMS) 235,250,240
  235 CONTINUE
      IRR = 1
      GO TO 1000
  240 CONTINUE
      NM2 = NMS + NMS
      DO 245 I=1,NM2
      BUFFER IN (INFILE,1) (BUFF(1),BUFF(IBFCNT))
  241 CONTINUE
      IF(UNIT,INFILE) 241,242,243
  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 + NMS +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
```

```
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.7HCOMPLEX) 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 CBUFFR(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 CBUFFR(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 = M*N
      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
```

```
      SUBROUTINE WRTEMX(IOUTFL, MXWRIT, RANDOU, NFS, NMS, LS, NMR, LWS,
     1         K, ID, A, ITYPE, M, N, PARM, IRR )
C
C          ROUTINE TO WRITE A MATRIX ON TAPE OR DISK FILE.
C        THIS VERSION WILL ONLY WORK WITH SEQUENTIAL FILES.
C        SOME VARIABLES ARE PASSED FOR RANDOM OPERATION BUT
C        ARE NOT CURRENTLY USED.
C
C        IOUTFL - TAPE NUMBER OR LEFT-JUSTIFIED FILE NAME
C        MXWRIT - .T. SNARK FORMAT  (NOT USED)
C                 .F. TEL001 FORMAT (NOT USED).
C        RANDOU - .T. RANDOM FILE       (NOT USED)
C                 .F. SEQUENTIAL FILE (NOT USED)
C        NFS    - NUMBER OF FILES TO SPACE  (SEQ. ONLY)
C        NMS    - NUMBER OF MATRICES TO SPACE
C        LS     - LEVEL NUMBER TO SPACE       (NOT USED )
C        NMR    - IDENTIFIER (NAME OR NUMBER) (NOT USED )
C        LWS    - LEVEL NUMBER OF THIS MATRIX (NOT USED )
C        K      - ROW DIMENSION OF A
C               - (IF 0, MATRIX IS ALREADY IN  /RWBUFF/ )
C        ID     - ARRAY CONTAINING MATRIX NAME
C        A      - ARRAY CONTAINING MATRIX
C        ITYPE  - REAL,DIAGONAL,NULL,MIXED,COMPLEX
C        M      - ROW DIMENSION OF MATRIX
C        N      - COLUMN DIMENSION OF MATRIX
C        PARM   - 10 WORD PARAMETER ARRAY
C        IRR    - ERROR RETURN
C               = 0, NO ERROR
C                 1, MATRIX SPACING IS NEGATIVE
C                 2, FILE SPACING IS NEGATIVE
C                 4, M*N DIMENSIONS ARE .GT. IBFCNT
C                 1500 + I, ENCOUNTERED EOF AFTER MATRIX I/2 WHILE
C                           SKIPPING MATRICES.
C
C
      DIMENSION ID(1), A(K,1), PARM(10), B(16)
      DIMENSION  IB(16)
      EQUIVALENCE (B,IB)
      LOGICAL   MXWRIT,RANDOU
C
      COMMON /RWBUFF/ BFCODE,IBFCNT,   BUFF(3280)
      DATA             BFCODE,IBFCNT  /8HBUFFSIZE, 3280 /
C
C
C         TEST FOR PROPER SIZE
C
      ISIZ = M*N
      IF(ITYPE.EQ.7HCOMPLEX) ISIZ = ISIZ+ISIZ
      IF(ISIZ.LE.IBFCNT) GO TO 205
      IRR = 4
      GO TO 1000
C
  205 CONTINUE
      IF(K.LE.0) GO TO 300
      IF(K.GE.M) GO TO 210
      IRR = 5
      GO TO 1000
```

```
  210 CONTINUE
C
C        DO FILE SPACING
C
      IF(NFS) 215,230,220
  215 CONTINUE
      IRR = 2
      GO TO 1000
  220 CONTINUE
      DO 225 I=1,NFS
  222 CONTINUE
      BUFFER IN (IOUTFL,1) (BUFF(1),BUFF(IBFCNT))
  221 CONTINUE
      IF(UNIT,IOUTFL) 221,222,225
  225 CONTINUE
  230 CONTINUE
C
C        DO MATRIX SPACING
C
      IF(NMS) 235,250,240
  235 CONTINUE
      IRR = 1
      GO TO 1000
  240 CONTINUE
      NM2 = NMS + NMS
      DO 245 I=1,NM2
      BUFFER IN (IOUTFL,1) (BUFF(1),BUFF(IBFCNT))
  241 CONTINUE
      IF (UNIT,IOUTFL) 241,242,243
  242 CONTINUE
      GO TO 245
  243 CONTINUE
      IRR = 1500 +(I+1)/2
      GO TO 1000
  245 CONTINUE
  250 CONTINUE
C
C        CREATE B HEADER RECORD
C
  300 CONTINUE
      IB(1) = ID(2)
      IB(2) = M
      IB(3) = N
      B(4) = 0
      IB(5) = 0
      IB(6) = ISIZ
      DO 325 I=7,16
      B(I) = PARM(I-6)
  325 CONTINUE
C
      IF(K.LE.0) GO TO 400
C
C        PUT ARRAY A INTO BUFFER
C
      IF(ITYPE.EQ.7HCOMPLEX) GO TO 375
C
C        NOT COMPLEX PUT INTO BUFFER.
```

```
C
      IX = 0
      DO 350 I=1,M
      DO 350 J=1,N
      IX = IX + 1
      BUFF(IX) = A(I,J)
  350 CONTINUE
      GO TO 400
C
C          COMPLEX, CALL ROUTINE TO STORE INTO BUFFER.
C
  375 CONTINUE
      K2 = K+K
      CALL COMBUF(A,K2,M,N,BUFF)
      IX = 2*M*N
C
  400 CONTINUE
C
C          WRITE THE B HEADER RECORD AND THE BUFFER ARRAY RECORD
C
      BUFFER OUT (IOUTFL,1) (B(1),B(16))
  500 CONTINUE
      IF (UNIT,IOUTFL) 500,510,510
  510 CONTINUE
C
      BUFFER OUT (IOUTFL,1) (BUFF(1),BUFF(IX))
  520 CONTINUE
      IF (UNIT,IOUTFL) 520,530,530
  530 CONTINUE
C
 1000 CONTINUE
      RETURN
      END
```

```
      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 = M*N                                                         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,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                IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC          FILES  00003
      COMMON / KERN / ERR,MXSKRN,IPKERN,NPLKRN,NSPATK,NROWEA            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,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                            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 /CHECKPR/ DPPCPR,GEOCPR,MODCPR,AICCPR,NWSCPR,SMCPR,GAFCPR  CHECKPR 00002
      LOGICAL  DPPCPR, GEOCPR, MODCPR, AICCPR,NWSCPR,  SMCPR, GAFCPR    CHECKPR 00003
      EQUIVALENCE (CHECKPR,DPPCPR)                                      DATAPP 00016
      LOGICAL CHECKPR                                                   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   /10HAFMBOX EXC /                                     FTNX1  00007
      NAMELIST /CARDB / XMACH                                           DATAPP 00022
      NAMELIST /CARDC / DEFAULT,PRVGEOM,PRVMODE,SYM,MTYPEW,MTYPET,      DATAPP 00023
     1                  NSURF,DIHW,DIHT,ISMPLW,WTGNAF,WTBL,             DATAPP 00024
     2                  PRGNAF,PRBL,PRSL,PRPAIC,PRSAIC,PRCOEF,PRMODS,   DATAPP 00025
     X                  PRDCP,PRGNAC,PRLW,PRNW,PRUW,PRCM,               BCSFRB 00003
     3                  PRBOX,PRDW,PRSW,PRVP,SUBDV,EXAIC,SMOOTH,NDEG,   DATAPP 00027
     4                  DPPCPR,GEOCPR,MODCPR,AICCPR,NWSCPR,SMCPR,GAFCPR, DATAPP 00028
     5                  NROWEA,CRDFIT,PLYWOOD                           DATAPP 00029
      NAMELIST /CARDD / OAIC,NAIC,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                                                                 DATAPP 00039
C          SET CONTROL PARAMETERS TO DEFAULT OPTIONS               DATAPP 00040
      DEFAULT = .FALSE.                                           DATAPP 00041
  100 CONTINUE                                                    DATAPP 00042
      PRVGEOM = .FALSE.                                           DATAPP 00043
      PRVMODE = .FALSE.                                           DATAPP 00044
      SYM     = 1.0                                               DATAPP 00045
      MTYPEW  = 2                                                 DATAPP 00046
      MTYPET  = 2                                                 DATAPP 00047
      NSURF   = 1                                                 DATAPP 00048
      DIHW    = .TRUE.                                            DATAPP 00049
      DIHT    = .TRUE.                                            DATAPP 00050
      ISMPLW  = 0                                                 DATAPP 00051
      WTGNAF  = .TRUE.                                            DATAPP 00052
      WTBL    = .FALSE.                                           DATAPP 00053
      PRGNAF  = .TRUE.                                            DATAPP 00054
      PRBL    = .FALSE.                                           DATAPP 00055
      PRSL    = .FALSE.                                           DATAPP 00056
      PRPAIC  = .FALSE.                                           DATAPP 00057
      PRSAIC  = .FALSE.                                           DATAPP 00058
      PRCOEF  = .FALSE.                                           DATAPP 00059
      PRMODS  = .FALSE.                                           DATAPP 00060
      PRBOX   = .FALSE.                                           DATAPP 00061
      PRDW    = .FALSE.                                           DATAPP 00062
      PRSW    = .FALSE.                                           DATAPP 00063
      PRDCP  = .FALSE.                                            DATAPP 00064
      PRGNAC = .FALSE.                                            DATAPP 00065
      PRLW   = .FALSE.                                            DATAPP 00066
      PRNW   = .FALSE.                                            DATAPP 00067
      PRUW   = .FALSE.                                            DATAPP 00068
      PRVP    = .FALSE.                                           DATAPP 00069
      PRCM   = .FALSE.                                            BCSFRB 00004
      SUBDV   = .FALSE.                                           DATAPP 00070
      EXAIC   = .FALSE.                                           DATAPP 00071
      SMOOTH = .FALSE.                                            DATAPP 00072
      CRDFIT = .FALSE.                                            DATAPP 00073
      NDEG = 0                                                    DATAPP 00074
      NROWEA = 0                                                  DATAPP 00075
      PLYWOOD = .FALSE.                                           DATAPP 00076
      DPPCPR = .FALSE.                                            DATAPP 00077
      GEOCPR = .FALSE.                                            DATAPP 00078
      MODCPR = .FALSE.                                            DATAPP 00079
      AICCPR = .FALSE.                                            DATAPP 00080
      NWSCPR = .FALSE.                                            DATAPP 00081
      SMCPR  = .FALSE.                                            DATAPP 00082
      GAFCPR = .FALSE.                                            DATAPP 00083
      PREVEX = EXEC                                               DATAPP 00084
      IF(DEFAULT) 400,300                                         DATAPP 00085
  200 CONTINUE                                                    DATAPP 00086
      OMACH = XMACH                                               DATAPP 00087
  300 CONTINUE                                                    DATAPP 00088
      READ (NT5,9005) TITLE                                       DATAPP 00089
 9005 FORMAT(8A10)                                                DATAPP 00090
      READ(NT5,CARDB)                                             DATAPP 00091
      IF(XMACH.GT.1.0) GO TO 310                                  DATAPP 00092
      WRITE (NT6,8005) XMACH                                      DATAPP 00093
      CALL FLUSH(1)                                               DATAPP 0009*
```

```
  310 CONTINUE
      IF(XMACH.GE.1.2) GO TO 320
      WRITE (NT6,8010)
      GO TO 350
  320 CONTINUE
      IF(XMACH.LE.3.0) GO TO 350
C
C           MACH NO. GREATER THAN 3.0
      IF(XMACH.LT.5.0) GO TO 340
      WRITE (NT6,8015) XMACH
      CALL FLUSH(1)
  340 CONTINUE
      WRITE (NT6,8020)
  350 CONTINUE
 8005 FORMAT(52H0*** MACH NUMBER OF LESS THAN 1.0 CAN NOT BE USED.
     1        14HMACH NUMBER = E15.6, 6H   *** )
 8010 FORMAT(62H0*** WARNING -- MACH NUMBER LESS THAN 1.2 IS BEING USED.
     1  *** )
 8015 FORMAT(43H0*** MACH NUMBER GREATER THAN 5.0. XMACH = E15.6,
     1        25H  PROGRAM TERMINATED. *** )
 8020 FORMAT(64H0*** WARNING -- MACH NUMBER GREATER THAN 3.0 IS BEING US
     1ED. *** )
      READ (NT5,CARDC)
      IF(SUBDV)500,510
  500 NSUBDV =3
      GO TO 515
  510 NSUBDV =1
  515 CONTINUE
C
      IF(DEFAULT) 100,400
C
C           CARD D
  400 CONTINUE
      OAIC    = 0
      NAIC    = 0
      OBAIC = 0
      NBAIC = 0
      INTAPE = 0
      NOUTP  = 1
      INFSP  = 0
      IOUFSP = 0
      READ (NT5,CARDD)
C
      IF(OAIC.EQ.0) GO TO 520
      NPLAIC = OAIC
      OPLAIC = .TRUE.
      GO TO 530
  520 CONTINUE
      NPLAIC = 0
      OPLAIC = .FALSE.
  530 CONTINUE
      IF(NAIC.EQ.0) GO TO 540
      NPLAIC = NAIC
      OPLAIC = .FALSE.
  540 CONTINUE
C
C           DETERMINE OPTIONS OF SPATIAL KERNELS
```

```
      IF(OSAIC.EQ.0) GO TO 560                                          DATAPP 00152
      NBPAIC = OSAIC                                                    DATAPP 00153
      OBPAIC = .TRUE.                                                   DATAPP 00154
      GO TO 570                                                         DATAPP 00155
  560 CONTINUE                                                          DATAPP 00156
      NBPAIC = 0                                                        DATAPP 00157
      OBPAIC = .FALSE.                                                  DATAPP 00158
  570 CONTINUE                                                          DATAPP 00159
      IF(NSAIC.EQ.0) GO TO 580                                          DATAPP 00160
      NBPAIC = NSAIC                                                    DATAPP 00161
      OBPAIC = .FALSE.                                                  DATAPP 00162
  580 CONTINUE                                                          DATAPP 00163
C                                                                       DATAPP 00164
      IF (NOUTP.NE.0) GO TO 600                                         DATAPP 00165
      IF(WTGNAF) WRITE (NT6,9041)                                       DATAPP 00166
      IF(WTBL  ) WRITE (NT6,9042)                                       DATAPP 00167
      WTGNAF = .FALSE.                                                  DATAPP 00168
      WTBL   = .FALSE.                                                  DATAPP 00169
  600 CONTINUE                                                          DATAPP 00170
C          CARD E                                                       DATAPP 00171
      DO 610 I=1,20                                                     DATAPP 00172
      XKS(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(XKS(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 (DPPCPR.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. NROWEA .NE. 0) WRITE (NT6,9546) NROWEA            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(PRPAIC)      WRITE (NT6,9583)                                  DATAPP  00211
      IF(PRSAIC)      WRITE (NT6,9584)                                  DATAPP  00212
      IF(PRDW)        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(PRVMODE)     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(MTYPET.EQ.1)WRITE (NT6,9536)                                   DATAPP  00231
      IF(MTYPET.EQ.2)WRITE (NT6,9537)                                   DATAPP  00232
      IF(MTYPET.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
      LBOXC  =   8                                                      DATAPP  00251
      LJALPH =200                                                       DATAPP  00252
      LPNTRM = 100                                                      DATAPP  00253
      LMODES = 1000                                                     DATAPP  00254
      LPNTSD =90                                                        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,OSAIC,NSAIC,INTAPE,INFSP,NOUTP,IOUFSP  DATAPP  00262
C                                                                       DATAPP  00263
 6001 FORMAT(1H0/45X, 39HTHE FOLLOWING TAPE SETUP IS REQUESTED - /)     DATAPP  00264
```

```
6002 FORMAT(51X,*OLD AIC TAPE =*,I3,/51X,*NEW AIC TAPE =*,I3,/          DATAPP
    1 51X,*OLD SPATIAL AIC TAPE =*,I3,/51X,*NEW SPATIAL AIC TAPE =*,I3, DATAPP
    2 / 51X,*INPUT DATA TAPE =*,I3,*  SPACED*,I3,* FILES,*              DATAPP
    3 / 51X,*OUTPUT TAPE     =*,I3,*  SPACED*,I3,* FILES,* //)          DATAPP
C                                                                       DATAPP
C        PRINT THE XKVAL OR XKS ARRAY.                                  DATAPP
C                                                                       DATAPP
      IF(XK1(1).EQ.-1.0) GO TO 700                                      DATAPP
      WRITE(NT6,6003)                                                   DATAPP
      WRITE(NT6,6004) (XK1(I),I=1,NKVALS)                               DATAPP
      GO TO 900                                                         DATAPP
C                                                                       DATAPP
  700 CONTINUE                                                          DATAPP
      IF(XKS(1).EQ.-1.0) GO TO 800                                      DATAPP
      WRITE(NT6,6005)                                                   DATAPP
      WRITE(NT6,6004) (XKS(I),I=1,NKVALS)                               DATAPP
      GO TO 900                                                         DATAPP
C                                                                       DATAPP
  800 CONTINUE                                                          DATAPP
      WRITE(NT6,6006)                                                   DATAPP
C                                                                       DATAPP
  900 CONTINUE                                                          DATAPP
C                                                                       DATAPP
 6003 FORMAT(1H0,29X, *THE FOLLOWING IS THE REDUCED FREQUENCY ARRAY BASE DATAPP
    1D ON BOX LENGTH* /)                                                DATAPP
 6005 FORMAT(1H0,29X, *THE FOLLOWING IS THE REDUCED FREQUENCY ARRAY BASE DATAPP
    1D ON WING SEMI-SPAN* /)                                            DATAPP
 6004 FORMAT(1H / (31X,6F11.5)  )                                       DATAPP
 6006 FORMAT(49H0*** WARNING -- NO REDUCED FREQUENCIES SPECIFIED.       DATAPP
    1    51H  PROGRAM WILL TERMINATE AFTER GEOMETRY SECTION *** )       DATAPP
C                                                                       DATAPP
C                                                                       DATAPP
 1000 RETURN                                                            DATAPP
 9500 FORMAT(1H1,29X,58(1H*),/30X,1H*,56X,1H*,/30X,58H*      UNSTEADY AE DATAPP
    1RODYNAMICS OF WING-HORIZONTAL TAIL     *,/30X,1H*,12X,*CONFIGURATI DATAPP
    2ONS IN SUPERSONIC FLOW*11X,1H*,/30X,1H*,56X,1H*,/30X,58H*     PREP DATAPP
    3ARED UNDER CONTRACT NO. AF 33615-70-C-1126     *,/30X,1H*,20X,*PRO DATAPP
    4JECT NO. 1370*, 20X,1H*,/30X,1H*,56X,1H*  )                        DATAPP
 9501 FORMAT(30X,1H*,5X,*FOR  DEPARTMENT OF THE AIR FORCE*,19X,1H*, /   DATAPP
    1 30X,1H*,10X,*AERONAUTICAL SYSTEMS DIVISION*, 17X,1H*, /           DATAPP
    2 30X,1H*,10X,*AIR FORCE FLIGHT DYNAMICS LABORATORY*,10X,1H*, /     DATAPP
    3 30X,1H*,10X,*WRIGHT-PATTERSON AIR FORCE BASE*,15X,1H*, /          DATAPP
    4 30X,1H*,56X,1H*, /                                                DATAPP
    5 30X,1H*,5X,*BY   THE BOEING COMPANY*,28X,1H*, /                   DATAPP
    6 30X,1H*,10X,*COMMERCIAL AIRPLANE DIVISION*,18X,1H*, /             DATAPP
    7 30X,1H*,10X,*SEATTLE, WASHINGTON*, 27X,1H*, /                     DATAPP
    8 30X,1H*,56X,1H*,/ 30X,58(1H*),/ )                                 DATAPP
 9041 FORMAT(72H0*** WARNING -- NO OUTPUT TAPE WAS REQUESTED FOR GENERAL DATAPP
    1IZED FORCES. *** )                                                 DATAPP
 9042 FORMAT(63H0*** WARNING -- NO OUTPUT TAPE WAS REQUESTED FOR BOX LIF DATAPP
    1TS. *** )                                                          DATAPP
 9043 FORMAT(54H0*** WARNING -- ORDER FOR VELOCITY POTENTIAL SMOOTHING  DATAPP
    1    36H TOO LARGE.  IT HAS BEEN REDUCED TO ,I2, 4H ***)            DATAPP
 9551 FORMAT(1H0,5X,7HTITLE -,13X,8A10,13X,7H- TITLE /1H0/45X,          DATAPP
    1    37HTHE FOLLOWING OPTIONS ARE REQUESTED - / )                   DATAPP
 9531 FORMAT(51X,*GEOMETRY FROM PREVIOUS CYCLE* )                       DATAPP
 9532 FORMAT(51X,*MODE SHAPES FROM PREVIOUS CYCLE* )                    DATAPP
```

```
9533 FORMAT(51X,*MODAL INPUT FOR WING IS POLYNOMIAL COEFFICIENTS* )      DATAPP  00323
9534 FORMAT(51X,*MODAL INPUT FOR WING IS ARBITRARY LOCATIONS FOR SURFAC DATAPP  00324
    1E FITTING * )                                                      DATAPP  00325
9535 FORMAT(51X,*MODAL INPUT FOR WING IS BOX CENTER VALUES* )           DATAPP  00326
9536 FORMAT(51X,*MODAL INPUT FOR TAIL IS POLYNOMIAL COEFFICIENTS* )      DATAPP  00327
9537 FORMAT(51X,*MODAL INPUT FOR TAIL IS ARBITRARY LOCATIONS FOR SURFAC DATAPP  00328
    1E FITTING * )                                                      DATAPP  00329
9538 FORMAT(51X,*MODAL INPUT FOR TAIL IS BOX CENTER VALUES* )           DATAPP  00330
9539 FORMAT(51X,*DIHEDRAL WING INFLUENCE CALCULATED* )                  DATAPP  00331
9540 FORMAT(51X,*DIHEDRAL TAIL INFLUENCE CALCULATED* )                  DATAPP  00332
9542 FORMAT(51X,*PRINT PRESSURE DIFFERENCE COEFFICIENTS* )              DATAPP  00333
9543 FORMAT(51X,*PRINT GENERALIZED AERODYNAMIC COEFFICIENTS* )          DATAPP  00334
9544 FORMAT(51X,*PRINT LONGITUDINAL WASHES ALONG SAMPLING CHORDS*)      DATAPP  00335
9545 FORMAT(51X,*PRINT NORMAL WASHES* )                                 DATAPP  00336
9546 FORMAT(51X,*EFFECTIVE SUBDIVIDED AREA OF*,I3,* ROWS REQUESTED* )   DATAPP  00337
9552 FORMAT(51X,*SYMMETRIC ANALYSIS*)                                   DATAPP  00338
9553 FORMAT(51X,*ANTI-SYMMETRIC ANALYSIS*)                              DATAPP  00339
9554 FORMAT(51X,*PLYWOOD OPTION IS USED. (PLANFORM BOUNDARY DETERMINED  DATAPP  00340
    1BY BOX PATTERN.) *)                                                DATAPP  00341
9556 FORMAT(51X,*SINGLE PLANFORM ANALYSIS*)                             DATAPP  00342
9557 FORMAT(51X,*ANALYSIS FOR 2 PLANFORMS*)                             DATAPP  00343
9558 FORMAT(1H0,100(1H$)///*  THE SPRUCE GOOSE IS LOOSE * //1H0,        DATAPP  00344
    1       100(1H$) )                                                  DATAPP  00345
9561 FORMAT(51X,*WRITE GENERALIZED AIR FORCES ON TAPE*)                 DATAPP  00346
9562 FORMAT(51X,*WRITE BOX LIFTS ON TAPE*)                              DATAPP  00347
9564 FORMAT(51X,*PRINT GENERALIZED AIR FORCES*)                         DATAPP  00348
9565 FORMAT(51X,*PRINT THE BOX LIFTS*)                                  DATAPP  00349
9566 FORMAT(51X,*PRINT THE SECTION LIFTS*)                              DATAPP  00350
9567 FORMAT(51X,*SECTION MOMENTS WILL BE COMPUTED WITH MODE SHAPE ONE*/ BCSFRB  00006
    1 51X,* ASSUMED FOR THE PITCH MODE.*   )                            BCSFRB  00007
9568 FORMAT(51X,*PRINT MODE SHAPES USED*)                               DATAPP  00351
9569 FORMAT(51X,*PRINT THE BOX PATTERN*)                                DATAPP  00352
9570 FORMAT(51X,*PRINT THE UPWASHES ALONG SAMPLING CHORDS*)             DATAPP  00353
9571 FORMAT(51X,*PRINT THE VELOCITY POTENTIALS*)                        DATAPP  00354
9572 FORMAT(51X,*BASIC (UNSUBDIVIDED) ANALYSIS WILL BE USED* )          DATAPP  00355
9573 FORMAT(51X,*SUBDIVISION WILL BE APPLIED*)                          DATAPP  00356
9575 FORMAT(51X,*ALL PARAMETERS SET TO DEFAULT VALUES*)                 DATAPP  00357
9576 FORMAT(51X,*APPROXIMATE KERNELS WILL BE USED*)                     DATAPP  00358
9577 FORMAT(51X,*EXACT KERNELS WILL BE USED*)                           DATAPP  00359
9578 FORMAT(51X,*PRINT THE SIDEWASHES ALONG SAMPLING CHORDS*)           DATAPP  00360
9580 FORMAT(51X,*MACH NUMBER = *, F8.6)                                 DATAPP  00361
9581 FORMAT(51X,*VELOCITY POTENTIALS WILL BE SMOOTHED BY A LEAST-SQUA*, DATAPP  00362
    1       *RES* / 61X,*POLYNOMIAL SURFACE FIT, OF ORDER*,I2,1H./      DATAPP  00363
    2     61X,*(0 = PROGRAM DETERMINED.) * )                            DATAPP  00364
9582 FORMAT(51X,*PRINT MODE SHAPE POLYNOMIAL COEFFICIENTS, IF AVAILAB*, DATAPP  00365
    1       *LE * )                                                     DATAPP  00366
9583 FORMAT(51X,*PRINT THE PLANAR AIC ARRAYS USED * )                   DATAPP  00367
9584 FORMAT(51X,*PRINT THE SPATIAL AIC ARRAYS USED * )                  DATAPP  00368
9585 FORMAT(51X,*VELOCITY POTENTIALS WILL BE SMOOTHED BY A LEAST SQUA*, DATAPP  00369
    1       *RES* / 61X,*POLYNOMIAL CHORDWISE FIT, OF ORDER*,I2,1H./    DATAPP  00370
    2     61X,*(0 = PROGRAM DETERMINED.) * )                            DATAPP  00371
     END                                                                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 (KPTWW, 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                MXBW,MYBBW,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,NPLKRN,NSPATK,NROWEA          KERN    00002
      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,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 /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,MTYPEW,MTYPET                          MODCOM  00002
      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
      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   0000?
```

```
      COMMON /PLANXY/ NWLE,NWTE,NTLE,NTTE, XWLE(10),YWLE(10),           PLANXY  00002
     1                XWTE(10),YWTE(10), XTLE(10),YTLE(10),            PLANXY  00003
     2                XTTE(10),YTTE(10)                                PLANXY  00004
      LOGICAL MXWRIT,RANDOU, MXREAD,RANDIN                              GEOMBX  00036
      COMMON /CHECKPR/ DPPCPR,GEOCPR,MODCPR,AICCPR,NWSCPR,SMCPR,GAFCPR  CHECKPR 00002
      LOGICAL  DPPCPR, GEOCPR, MODCPR, AICCPR,NWSCPR,  SMCPR, GAFCPR    CHECKPR 00003
      EQUIVALENCE (CHECKPR,GEOCPR)                                      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 KPTWW(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. /                       FTNX1   00008
      DATA  NBWRD  /20/                                                 FTNX1   00009
      DATA  INIT,XINIT / 377670OB.  37654321777777777777B/              GEOMBX  00058
      DATA  EPS  / 1.0E-4 /                                             GEOMBX  00059
C                                                                       FTNX1   00010
C         NAMELIST PARAMETERS  FOR CARDS TO BE READ IN THIS SECTION     FTNX1   00011
      NAMELIST /CARDF / WLAX,WLAZ,PSIW,  TLAX,TLAZ, PSIT,     CHECKPR   FTNX1   00012
     1         /CARDG / NCHRDS, XCENTR, XEDGE, ICHORD,IBOXF,IBOXL,ZLOC  FTNX1   00013
C              CARDH   NWLE, NWTE, NTLE, NTTE    (4I5)                  FTNX1   00014
C              CARDS I TO L    (6E10.0)                                 FTNX1   00015
C                                                                       FTNX1   00016
 5001 FORMAT(4I5)                                                       FTNX1   00017
 5002 FORMAT(6E10.0)                                                    FTNX1   00018
C                                                                       FTNX1   00019
C                                                                       GEOMBX  00060
      LSCHDS = LBOXC * 20                                               GEOMBX  00061
      XSUBDV = NSUBDV                                                   GEOMBX  00062
      NSUBD2 = NSUBDV/2                                                 GEOMBX  00063
      NSUBCN = NSUBD2 + 1                                               GEOMBX  00064
      HALFBX = XSUBDV/2.0                                               GEOMBX  00065
C                                                                       GEOMBX  00066
C         IS PREVIOUS GEOMETRY TO BE USED -                             GEOMBX  00067
      IF (.NOT. PRVGEOM) GO TO 15                                       GEOMBX  00068
C         YES.  HAS THE MACH NUMBER CHANGED -                           GEOMBX  00069
      IF ( XMACH .EQ. OMACH) GO TO 2000                                 GEOMBX  00070
C         YES.  SKIP THE GEOMETRY READS, BUT REDO THE REST FOR THE      GEOMBX  00071
C               NEW BOX PATTERN                                         GEOMBX  00072
      BIO = 31                                                          GEOMBX  00073
      XCNTRO = XCENTR                                                   GEOMBX  00074
      GO TO 272                                                         GEOMBX  00075
C                                                                       GEOMBX  00076
C         READ CARDS F AND G                                            GEOMBX  00077
   15 CONTINUE                                                          GEOMBX  00078
      WLAX = 0.                                                         GEOMBX  00079
      WLAZ = 0.                                                         GEOMBX  00080
      PSIW = 0.                                                         GEOMBX  00081
      TLAX = 0.                                                         GEOMBX  00082
      TLAZ = 0.                                                         GEOMBX  00083
      PSIT = 0.                                                         GEOMBX  00084
      READ (NT5,CARDF)                                                  GEOMBX  00085
      WRITE (NT6,6010) WLAX,WLAZ,PSIW                                   GEOMBX  00086
```

```
      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
      NCHRDS = INIT                                                       GEOMBX 00108
      XCENTR = XINIT                                                      GEOMBX 00109
      XEDGE  = XINIT                                                      GEOMBX 00110
      DO 50 I=1,10                                                        GEOMBX 00111
      ICHORD(I) = INIT                                                    GEOMBX 00112
      IBOXF(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) NCHRDS,XCENTR, XEDGE                               GEOMBX 00119
C                                                                         GEOMBX 00120
      IVAL = 6HNCHRDS                                                     GEOMBX 00121
      IF (NCHRDS .EQ. INIT) GO TO 8010                                    GEOMBX 00122
      IF (NCHRDS .LE. 0 .OR. NCHRDS .GE. LSCHDS/NSUBDV) GO TO 8015        GEOMBX 00123
      NYBW = NCHRDS                                                       GEOMBX 00124
      IVAL = 6HXCENTR                                                     GEOMBX 00125
      IF (XCENTR .EQ. XINIT) GO TO 120                                    GEOMBX 00126
C        USE XCENTR DIRECTLY, IGNORE XEDGE                                GEOMBX 00127
      IF (XEDGE .NE. XINIT)  WRITE (NT6,9010)                             GEOMBX 00128
      GO TO 125                                                           GEOMBX 00129
C        GET XCENTR 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. NYBW   .OR. ICHORD(I) .LE. 0) GO TO 130          GEOMBX 00137
      IF (IBOXF(I) .EQ. INIT .OR. IBOXL(I) .EQ. INIT) GO TO 140           GEOMBX 00138
      IF (IBOXF(I) .LT. 1 .OR. IBOXL(I) .GT. LBXCDW/NSUBDV) GO TO 130     GEOMBX 00139
      IF (IBOXF(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
      IF (ZLOC(I) .EQ. XINIT) ZLOC(I) = 0.
  150 CONTINUE
  160 CONTINUE
      WRITE (NT6,6017) ISMPLW, (ICHORD(I), IBOXF(I),IBOXL(I), ZLOC(I),
     1    I = 1,ISMPLW )
      GO TO 200
C         SAMPLING OF SOWNWASHES ILLEGAL IF TAIL DEFINED
  170 CONTINUE
      WRITE (NT6,9030) ISMPLW
      ISMPLW = 0
  200 CONTINUE
C
C         OBTAIN THE LEADING AND TRAILING EDGE VALUES
C         CARD INPUT OF PLANFORMS IS REQUIRED
      READ (NT5,5001) NWLE,NWTE,NTLE,NTTE
  210 WRITE(NT6,6021) NWLE,NWTE
      GO TO (214,212),NSURF
  212 WRITE(NT6,6022) NTLE,NTTE
  214 IVAL = 4HNWLE
      IF (NWLE .LT. 2 .OR. NWLE .GT. 10) GO TO 8030
      IVAL = 4HNWTE
      IF (NWTE .LT. 2 .OR. NWTE .GT. 10) GO TO 8030
      IF (NSURF .EQ. 1) GO TO 220
      IVAL = 4HNTLE
      IF (NTLE .LT. 2 .OR. NTLE .GT. 10) GO TO 8030
      IVAL = 4HNTTE
      IF (NTTE .LT. 2 .OR. NTTE .GT. 10) GO TO 8030
  220 CONTINUE
C
C         CARDS I AND J - WING DEFINITION POINTS
      WRITE (NT6,6029)
      IVAL = 9HWING L.E.
      READ (NT5,5002) (XWLE(I),YWLE(I),I=1,NWLE)
      WRITE (NT6,6030) IVAL,(XWLE(I),YWLE(I),I=1,NWLE)
      CALL EDGCHK(XWLE,YWLE,NWLE,1,IRR)
      IF (IRR .NE. 0) GO TO 8050
      IVAL = 9HWING T.E.
      READ (NT5,5002) (XWTE(I),YWTE(I),I=1,NWTE)
      WRITE (NT6,6030) IVAL,(XWTE(I),YWTE(I),I=1,NWTE)
      CALL EDGCHK(XWTE,YWTE,NWTE,2,IRR)
      IF (IRR .NE. 0) GO TO 8050
      IF (NSURF .EQ. 1) GO TO 270
C
C         CARDS K AND L - TAIL DEFINITION POINTS
      IVAL = 9HTAIL L.E.
      READ (NT5,5002) (XTLE(I),YTLE(I),I=1,NTLE)
      WRITE (NT6,6030) IVAL,(XTLE(I),YTLE(I),I=1,NTLE)
      CALL  EDGCHK(XTLE,YTLE,NTLE,1,IRR)
      IF (IRR .NE. 0) GO TO 8050
      IVAL = 9HTAIL T.E.
      READ (NT5,5002) (XTTE(I),YTTE(I),I=1,NTTE)
      WRITE (NT6,6030) IVAL, (XTTE(I),YTTE(I),I=1,NTTE)
      CALL EDGCHK(XTTE,YTTE,NTTE,2,IRR)
      IF (IRR .NE. 0) GO TO 8050
      GO TO 270
C
```

```
C        PLANFORM DEFINITIONS TO BE READ FROM TAPE
  230 CONTINUE
C
  270 CONTINUE
      B1BETA = YWLE(NWLE)/MYBW
  272 CONTINUE
      B1  =  B1BETA * SQRT(XMACH*XMACH-1.0)
      B1BTA2 = B1BETA* 0.5
      B12  =  B1 * 0.5
      IF (NSUBDV .NE. 1) GO TO 275
      B1S  = B1
      B1BTAS = B1BETA
      GO TO 280
  275 B1S  = B1/XSUBDV
      B1BTAS = B1BETA/XSUBDV
  280 CONTINUE
      WRITE (NT6,6040) B1,B1BETA
C
C        SET THE XKVAL ARRAY IF XKS WAS INPUT
C
      IF(XKS(1).EQ.-1.0) GO TO 295
      DO 290 I=1,NKVALS
      XKVAL(I) = XKS(I) * (B1/YWLE(NWLE))
  290 CONTINUE
  295 CONTINUE
C
C        DETERMINE THE GLOBAL COORDINATE LOCATION OF THE FIRST UN-
C        SUBDIVIDED PLANFORM BOX CENTER, XCENTR
      IVAL = 6HXCENTR
      IF (XCENTR .EQ. XINIT) XCENTR = XEDGE + B12
      XEDGEW = XWLE(1) + (XWLE(2)-XWLE(1)) * B1BTA2 /  YWLE(2)
      IF (PRVGEOM) XEDGEW = B1O* XWLE(1) - B1O + XCNTRO  +
     1     B1O *(XWLE(2) - XWLE(1)) * .5 /(YWLE(2) - .5)
      IF (XCENTR-XEDGEW) 310,330,320
  310 DO 315 I = 1,51
      XCENTR = XCENTR + B1
      IF (XCENTR .GE. XEDGEW) GO TO 330
  315 CONTINUE
      GO TO 8060
  320 DO 325 I = 1,51
      IF (XCENTR-B1 .LT. XEDGEW) GO TO 330
      XCENTR = XCENTR - B1
  325 CONTINUE
      GO TO 8060
  330 CONTINUE
C        IS PREVIOUS GEOMETRY BEGIN USED -
      IF (.NOT. PRVGEOM) GO TO 355
C
C        YES.  CONVERT X-COORDINATE VALUES TO NEW BOX LENGTH
      PSIDIF = PSIT - PSIW
      SLIDE =  -B1O + XCNTRO - XCENTR
      DO 335 I = 1,NWLE
  335 XWLE(I) = (B1O*XWLE(I) + SLIDE)/B1 + 1.0
      DO 340 I = 1,NWTE
  340 XWTE(I) = (B1O*XWTE(I) + SLIDE)/B1 + 1.0
      IF (NSURF .EQ. 1) GO TO 390
      DO 345 I = 1,NTLE
```

```
  345 XTLE(I) = (B1O*XTLE(I) + SLIDE)/B1 + 1.0
      DO 350 I = 1,NTTE
  350 XTTE(I) = (B1O*XTTE(I) + SLIDE)/B1 + 1.0
      GO TO 390
C
C         CONVERT GEOMETRIC INFORMATION TO THE NON-DIMENSIONAL
C         NC, MC, LC COORDINATE SYSTEM
  355 CONTINUE
      DO 360 I = 1,NWLE
      XWLE(I) = (XWLE(I)-XCENTR)/B1 + 1.0
  360 YWLE(I) =  YWLE(I)/B1BETA    + 0.5
      DO 365 I = 1,NWTE
      XWTE(I) = (XWTE(I)-XCENTR)/B1 + 1.0
  365 YWTE(I) =  YWTE(I)/B1BETA    + 0.5
      GO TO (370,375),NSURF
  370 CAPL = 0.
      PSIT = 0.
      PSIDIF = -PSIW
      NWBT  = 0
      NWBBT = 0
      NWBST = 0
      NWBBST = 0
      IF (ISMPLW .EQ. 0) GO TO 390
C         TRANSFORM ZLOC FOR THE SAMPLE WASH CHORDS TO A NON-DIMENSIONAL
C         UNROTATED LC COORDINATE HAVING ITS ZERO ON THE WING CENTER
C         LINE
      DO 372 I = 1,ISMPLW
      ZLOC(I)  = (ZLOC(I) - WLAZ) /B1BETA
  372 CONTINUE
      GO TO 390
  375 XDIFF = WLAX + XCENTR - TLAX
      DO 380 I = 1,NTLE
      XTLE(I) = (XTLE(I)-XDIFF)/B1 + 1.0
  380 YTLE(I) =  YTLE(I)/B1BETA  + 0.5
      DO 385 I = 1,NTTE
      XTTE(I) = (XTTE(I)-XDIFF)/B1 + 1.0
  385 YTTE(I) =  YTTE(I)/B1BETA  + 0.5
      CAPL  =  (TLAZ-WLAZ)/B1BETA
      PSIDIF = PSIT - PSIW
C
C         CHECK FOR TAIL CROSSING WING
      IF (PSIDIF) 386,389,387
  386 IF (CAPL .LE. 0) GO TO 389
      GO TO 388
  387 IF (CAPL .GE. 0) GO TO 389
  388 YCROSS = CAPL/(SIN(PSIW)-SIN(PSIT)) + .5
      IF (YWLE(NWLE)*COS(PSIW) .LT. YCROSS) GO TO 390
      IF (YTLE(NTLE)*COS(PSIT) .GE. YCROSS) GO TO 8080
      GO TO 390
  389 YCROSS = .5
C
C         ZERO OUT THE BOX CODE ARRAYS
  390 CONTINUE
      DO 430 J = 1,LBOXC
      DO 420 I = 1,LBXCDW
  420 IBOXW(I,J) = 0
      DO 430 I = 1,LBXCDT
```

```
 430 IBOXT(I,J) = 0
C
C        GET THE (SUBDIVIDED) BOX CODE ARRAY FOR THE ON-PLANFORM WING
C        BOXES
      IXBW = 0
      IXBT = 0
      IXBST = 0
      MXBT = 0
C
      CALL BXCDPF( XWLE,YWLE,NWLE,XWTE,YWTE,NWTE, LBXCDW,IBOXW)
C        RETURNS-  IBOXW ,  ONES FOR ON PLANFORM BOXES
C                  IXBW   = LOCATION OF FIRST UNSUBDIVIDED BOX CENTER
C                  MXBSW  = NUMBER OF SUBDIVIDED ROWS ON THE WING
C                  MYBSW  = NUMBER OF SUBDIVIDED CHORDS ON THE WING
C                  MXBW   = NUMBER OF UNSUBDIVIDED ROWS
C                  MYBW   = NUMBER OF UNSUBDIVIDED CHORDS
C                  FEXLOC = ARRAY OF LEADING EDGE LOCATIONS
C                  TEXLOC = ARRAY OF TRAILING EDGE LOCATIONS
C
      MYBBSW = MYBSW
      MXBBW = MXBW
      MXBBSW = MXBSW
      IF (.NOT. CHECKPR) GO TO 440
      CALL PRNTBC(IBOXW,LBXCDW, 1, MXBSW, MYBSW, .T. )
      WRITE (NT6,7040) ( FEXLOC(I), I = 1,MYBSW)
      WRITE (NT6,7045) (TEXLOC(I), I = 1,MYBSW)
  440 CONTINUE
C
C        SEARCH THE WING FOR THE FORWARD MOST DIAGONAL INTERSECTING
C        AN ON-PLANFORM BOX.  THIS DEFINES THE LIMIT FOR ANY TIP
C        DIAPHRAGM.
C              JDIAG = THE J-LOCATION (SUBDIVIDED) OF THE DIAGONAL AT
C                      THE FIRST ROW OF THE PATTERN.
      JDIAG = 1
      PREV  = 0.
      DO 530 J = 2,MYBSW
      PREV = PREV + 1.0
      IF (FEXLOC(J) .GT. PREV) GO TO 530
      PREV = FLOAT(IFIX(FEXLOC(J)))
      JDIAG = J - PREV
  530 CONTINUE
C
C        INITIALIZE THE IWAKE ARRAY
      DO 540 J = 1,MYBSW
      IWAKE(J) = TEXLOC(J)
  540 CONTINUE
      IF (MYBSW .EQ. LSCHDS) GO TO 548
      MYBSW1 = MYBSW + 1
      DO 544 J = MYBSW1,LSCHDS
  544 IWAKE(J) = 0
  548 CONTINUE
      IF (NSURF .NE. 2) GO TO 705
C        THERE ARE 2 SURFACES.  DETERMINE THE FIRST PLANFORM BOX OF THE
C        SECOND SURFACE
      YMIN = .5*(1.0 + 1.0/XSUBDV)
      DELE = (XTLE(2)-XTLE(1)) / (YTLE(2)-YTLE(1))
      XMINS = XTLE(1)+ (YMIN-YTLE(1)) * DELE
```

```
      IXBST = XSUBDV*(XMINS-1.0) + IXBW + 1                              GEOMBX 00372
      IF (AINT(XMINS) .EQ. XMINS) 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 BOUNDSOF 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. CHECKPR) 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                            DIAPHRAM, FOR TAIL                          GEOMBX 00412
C                   MYBBT  = NUMBER OF UNSUBDIVIDED CHORDS               GEOMBX 00413
C                                                                        GEOMBX 00414
      IF (.NOT. (PRBOX .OR. CHECKPR) ) 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
```

```
      CAPLL = CAPL
C       (DO 680 JT = NSUBCN,MYBBST,NSUBDV )
      JT = NSUBCN
  525 CONTINUE
      YCT = JT - .5
C          Y-OFFSET OF RECEIVING CHORD FROM CENTER-LINE, TAIL PLANE
C           GET ICT, THE I-LOCATION OF AFTMOST RECEIVING BOX ON TAIL CHRD
C          IS THE TAIL CHORD ON-PLANFORM OR DIAPHRAGM -
      IF (JT .GT. MYBST) GO TO 550
      JJ = JT + MYBSW
      ICT =  TEXLOC(JJ) + EPS  - AMOD( TEXLOC(JJ)-IXBW, XSUBDV)
      GO TO 555
  550 CONTINUE
      ICT =  IXBT - NSUBDV
  555 CONTINUE
      ICTP1 = ICT + NSUBDV
      IF (ICTP1 .GT. MXBST) GO TO 570
C          CHECK FOR WAKE DIAPHRAGM AFT OF TAIL CHORD
      CALL DCODER( IBOXT(ISUBT,1),LBXCDT, ICTP1,JT, MXBST,JT,
     1    .T., ICODE)
      II = 1
      DO 560 I = ICTP1,MXBST,NSUBDV
      IF (ICODE(II).EQ. 0) GO TO 570
      ICT = I
      II = II + 1
  560 CONTINUE
  570 CONTINUE
C          ICT     = X-LOCATION OF AFT-MOST TAIL BOX ON THE CHORD
      EL  = COS(PSIW)*CAPL*XSUBDV + SIN(PSIDIF)*YCT
C          EL      = PERPENDICULAR DISTANCE FROM RECEIVING CHORD TO RIGHT
C                     WING PLANE, POSITIVE DOWNWARD.
C
C          ENTRY INTO THE LOOP FOR WASH SAMPLING CHORDS, FROM 705*
  580 CONTINUE
C     START OF LOOP ON WING CHORDS, ENDING AT 650
      JW = NSUBCN
  600 CONTINUE
      YJW = JW - .5
C              YJW    = Y-OFFSET OF SENDING CHORD FROM CENTER LINE,
C                         WING PLANE
      YMUBAR = -YJW + COS(PSIDIF)*YCT + SIN(PSIW)*CAPLL*XSUBDV
C              YMUBAR = Y-DISTANCE BETWEEN CHORD CENTERS, SENDING (WING)
C                         PLANE
      IF (ABS(YMUBAR) .LE. HALFBX) GO TO 630
      IF (YMUBAR .LT. -HALFBX) YMUBAR = YMUBAR + HALFBX
      IF (YMUBAR .GT.  HALFBX) YMUBAR = YMUBAR - HALFBX
C              YMUBAR = Y-DISTANCE TO NEAREST BOX EDGE, WING PLANE
      XNUBAR = SQRT(YMUBAR**2  +   (EL*XSUBDV)**2 )
C              XNUBAR = DISTANCE FORWARD FROM RECEIVING CENTER TO NEAR-
C                         EST PORTION OF SENDING CHORD
      GO TO 635
  630 CONTINUE
      XNUBAR = ABS(EL)*XSUBDV
  635 CONTINUE
      XNUBAR = XNUBAR + HALFBX
      INTRST = ICT - IFIX(XNUBAR+EPS - AMOD(XNUBAR,XSUBDV) )
      IF (JW .GT. MYBBSW) GO TO 640
```

```
      IWAKE(JW) = MAX0(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
  660 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 = MAX0(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,IBOXW)         GEOMBX  00520
      MXBBSW = MXBST                                                    GEOMBX  00521
      IF (.NOT. CHECKPR) GO TO 720                                      GEOMBX  00522
      CALL PRNTBC(IBOXW,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 = (ICHORD(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                                                   FTNX1  00028
      IF (JCHRD .LE. ISMPLW) GO TO 704                                    FTNX1  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,LBXCDW,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,LBXCDW, 1, MXBBSW,  MYBBSW, .T.)                  GEOMBX 00564
      IF (NSUBDV .NE. 1)  CALL PRNTBC(IBOXW,LBXCDW,                       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
      NPLKRN = MAX0(MXBBW, MXBT-IXBT/NSUBDV + 1)                          GEOMBX 00570
      IF (COPLAN) NPLKRN = 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 = 5HMIXED                                                     GEOMBX 00576
      IVAL = 5HIBOXW                                                      GEOMBX 00577
      PARM(1)  = 0.                                                       GEOMBX 00578
      PARM(2)  = XMACH                                                    GEOMBX 00579
      M = MXBBSW                                                          GEOMBX 00580
      N = (MYBBSW-1)/NBWRD + 1                                            GEOMBX 00581
      K = LBXCDW                                                          GEOMBX 00582
      CALL WRTEMX(IGEOSC,MXWRIT,RANDOU,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 = 6HFEXLOC                                                  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 = 6HTEXLOC                                                  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 GMAREA(IBOXW,LBXCDW, .T., ALPHA,IJALPH, NALPHW)             GEOMBX 00618
      NALPH = NALPHW                                                   GEOMBX 00619
      IF (NSURF .EQ. 1 .OR. COPLAN) GO TO 740                          GEOMBX 00620
      CALL GMAREA(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  = 5HALPHA                                                  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  = 6HIJALPH                                                 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 MUAIC 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 MUAICS TRANSFERRED TO IWTFSC           GEOMBX 00645
C             NSCR  = TOTAL NUMBER OF MUAICS PRESENTLY ON IVPSC        GEOMBX 00646
C                                                                      GEOMBX 00647
      ISCR = 0                                                         GEOMBX 00648
```

```
      NSCR = 0
      NROWBR = 0
      NROWBLL = 0
      IPARM(3) = 0
      IPARM(6) = 0
      REWIND IWTFSC
      REWIND IVPSC
      M = 2
      K = 2
      NSPATK = 0
      NWAK = 0
      IF (SYM .EQ. 0) GO TO 904
      IF (PSIW .EQ. 0 .OR. .NOT. DIHW) GO TO 800
C
C        START OF LOOP FOR WING-WING PARAMETERS ON RECEIVING CHORDS,
C        UNSUBDIVIDED  (DETERMINES SPATIAL MUAIC VALUES)
      DO 790 JCOL =1,MYBBW
C
      CALL PWMAIC (.T., IBOXW,LBXCDW, IWAKE, JCOL)
C        COMPUTES MUAIC ARRAY FOR THE LEFT SURFACE CONTRIBUTION TO
C        CHORD JCOL OF THE RIGHT SURFACE
C             SURF = .T. INDICATES SOME RIGHT SURFACE CONTRIBUTION WAS
C                   FOUND
      IF (.NOT. SURF) GO TO 800
      NWAK = NWAK + 1
      NSPATK = NWAK
      KPTWW(NWAK) = NSPATK
      IVAL = 10H WING-WING
      IF (CHECKPR) WRITE(NT6,7020) IVAL,JCOL,YBAR,EL, NROWS,(MUAIC(1,
     1    NROWS-I+1),MUAIC(2,NROWS-I+1), I=1,NROWS)
C
C        WRITE MUAIC ARRAY ON THE SCRATCH FILE
      N = NROWS
      PARM(4) = YBAR
      PARM(5) = EL
      CALL WRTEMX(IVPSC, MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS, K, ID,
     1            MUAIC, ITYPE, M,N, PARM, IRR)
      IF (IRR .NE. 0) GO TO 8075
      NSCR = NSCR + 1
C
  790 CONTINUE
C        END OF LOOP ON RECEIVING CHORDS FOR WING-WING PARAMETERS,
C
  800 CONTINUE
      NTTK = 0
      IF (NSURF .NE. 2) GO TO 900
      IF (SYMT .EQ. 0) GO TO 900
      IF (PSIT .EQ. 0 .OR. .NOT. DIHT) GO TO 900
      IF (PSIDIF .EQ. 0.) REWIND IVPSC
C
C        START OF LOOP FOR TAIL-TAIL PARAMETERS ON RECEIVING CHORDS,
C        UNSUBDIVIDED
      DO 895  JCOL = 1,MYBBT
C
      CALL PWMAIC (.F., IBOXT(ISUBT,1),LBXCDT, IWAKE, JCOL)
C        COMPUTES MUAIC ARRAY FOR THE CONTRIBUTION OF THE LEFT TAIL ON
C        CHORD JCOL OF THE RIGHT TAIL SURFACE
```

```
C
      IF (.NOT. SURF) GO TO 900
      NTTK = NTTK + 1
C
      IVAL = 10H TAIL-TAIL
C         IF THE WING AND TAIL HAVE THE SAME DIHEDRAL, MODIFY THE WING
C         MUAIC ARRAY WHERE NEEDED BY THE TAIL, AND USE IT
      IF (PSIDIF .EQ. 0.) GO TO 840
C         OTHERWISE, WRITE THE MUAICS FOUND ONTO IVPSC
      NSPATK = NSPATK + 1
      KPTTT(NTTK) = NSPATK
      IVAL = 10H TAIL-TAIL
      IF (CHECKPR) WRITE(NT6,7020) IVAL,JCOL,YBAR,EL, NROWS,(MUAIC(1,
     1    NROWS-I+1),MUAIC(2,NROWS-I+1), I=1,NROWS)
C
C         WRITE MUAIC ARRAY ON SCRATCH FILE
      N = NROWS
      PARM(4) = YBAR
      PARM(5) = EL
      CALL WRTEMX(IVPSC, MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS, K, ID,
     1            MUAIC, ITYPE, M,N, PARM, IRR)
      IF (IRR .NE. 0) GO TO 8075
      NSCR = NSCR + 1
C
      GO TO 895
  840 CONTINUE
      IF (ISCR .GE. NSCR) GO TO 850
      CALL RDINIT
      CALL READMX(IVPSC, MXWRIT,RANDOU,NFS,NMS,LS,NMR, K, NID,ID,
     1            ITYPE, LRS,  MUAICL, M,N, PARM, IRR)
      IF (IRR .NE. 0) GO TO 8090
      ISCR = ISCR + 1
C         MERGE THE TWO MUAIC ARRAYS
      DO 845 I = 1,N
      IF (I .GT. NROWS) GO TO 842
      IF (MUAICL(1,I) .EQ. 0) GO TO 845
      IF (MUAIC(1,I)  .EQ. 0) GO TO 842
      MUAIC(1,I) = MIN0( MUAIC(1,I),MUAICL(1,I) )
      MUAIC(2,I) = MAX0( MUAIC(2,I),MUAICL(2,I) )
      GO TO 845
  842 MUAIC(1,I) = MUAICL(1,I)
      MUAIC(2,I) = MUAICL(2,I)
  845 CONTINUE
      NROWS = MAX0(NROWS,N)
      KPTTT(NTTK) = ISCR
      IF (CHECKPR) WRITE(NT6,7020) IVAL,JCOL,YBAR,EL, NROWS,
     1    (MUAIC(1,NROWS-I+1),MUAIC(2,NROWS-I+1), I = 1,NROWS)
      GO TO 855
C         THERE WERE NO MATRICES TO BE MERGED
  850 NSPATK = NSPATK + 1
      KPTTT(NTTK) = NSPATK
C         WRITE MERGED AICS ONTO 2ND SCRATCH FILE
  855 CONTINUE
      N = NROWS
      CALL WRTEMX(IWTFSC, MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS, K, ID,
     1            MUAIC, ITYPE, M,N, PARM, IRR)
      IF (IRR .NE. 0) GO TO 8110
```

```
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 (COPLAN) GO TO 1015                                            GEOMBX  00797
      IF (NSURF .NE. 2) GO TO 1120                                      GEOMBX  00798
      CAPLL = CAPL                                                      GEOMBX  00799
      YMUVSP = 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 JCOL = 1,MYBBT )                                      FTNX1   00030
      JCOL = 1                                                          FTNX1   00031
  909 CONTINUE                                                          FTNX1   00032
      IF (JCOL .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 DCODER(IBOXT(ISUBT,1),LBXCDT, IROW,JCOL, MXBT,JCOL,.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
  917 CONTINUE
  918 CONTINUE
      III = I
      DO 920 I = III, MXBT
      IF (ICODE(II) .EQ. 0) GO TO 925
      II = II + 1
  920 CONTINUE
  925 IROW = IROW + II - 2
C           ENTRY INTO THE LOOP FROM SAMPLE WASH LOOP, FROM 1120*
  930 CONTINUE
      CALL PWTAIC( IBOXW,LBXCDW, IROW,JCOL,  CAPLL, YMUVSP)
C           GETS THE MUAIC AND MUAICL ARRAYS FOR RIGHT AND LEFT CONTRI-
C           BUTIONS TO THE TAIL
      IF (SURF) GO TO 935
      IF ( .NOT. SURFL) GO TO 1015
      GO TO 985
  935 NRWTK = NRWTK + 1
C           DETERMINE WHETHER WING AND TAIL ARE PARALLEL
      IF (PSIDIF .EQ. 0 .AND. NSURF .EQ. 2) GO TO 940
      NSPATK = NSPATK + 1
      KPTRWT(NRWTK) = NSPATK
C           WRITE NEW MUAIC ARRAY ON SCRATCH
      IVAL = 10HR WNG-TAIL
      N = NROWS
      PARM(4) = YBAR
      PARM(5) = EL
      CALL WRTEMX(IVPSC, MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS, K, ID,
     1             MUAIC, ITYPE, M,N, PARM, IRR)
      IF (IRR .NE. 0) GO TO 8075
      NSCR2= NSCR2 + 1
      IF (CHECKPR) WRITE(NT6,7020) IVAL,JCOL,YBAR,EL, NROWS,(MUAIC(1,
     1     NROWS-I+1),MUAIC(2,NROWS-I+1), I=1,NROWS)
C           REINITIALIZE THE RIGHT MUAIC ARRAY
      DO 937 I = 1,NROWS
      MUAIC(1,I) = I+I
      MUAIC(2,I) = 0
  937 CONTINUE
C
      GO TO 982
C           OLD MUAIC ARRAY HAS BEEN FOUND WHICH MATCHES
  940 CONTINUE
C           ALL RIGHT HAND MUAICS ARE THE SAME, PARALLEL SURFACES
      IF (JCOL .GT. 1) GO TO 945
      NSPATK = NSPATK + 1
      IPARAL = NSPATK
  945 CONTINUE
      KPTRWT(NRWTK) = IPARAL
      NROWSR = MAX0(NROWSR,NROWS)
C
  980 CONTINUE
      IVAL = 10HR WNG-TAIL
      IF (CHECKPR) WRITE(NT6,7020) IVAL,JCOL,YBAR,EL, NROWS,(MUAIC(1,
     1     NROWS-I+1),MUAIC(2,NROWS-I+1), I=1,NROWS)
C
  982 CONTINUE
      IF (.NOT. SURFL) GO TO 1000
```

```
  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 TO990 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
      NROWSLL = MAX0(NROWSLL,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 (ISMPLW .NE. 0) GO TO 1150                                     GEOMBX  00909
 1010 CONTINUE                                                          GEOMBX  00910
      JCOL = JCOL + 1                                                   FTNX1   00033
      IF (JCOL .LE. MYBBT) GO TO 909                                    FTNX1   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
      NROWSX = MIN0(NROWSR,NROWSLL)                                     GEOMBX  00916
      DO 1012 I = 1,NROWSX                                              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 IWTFSC                                   GEOMBX  00922
      IVAL = 10HPARAL TAIL                                              GEOMBX  00923
      PARM(4) = YBAR                                                    GEOMBX  00924
      PARM(5) = EL                                                      GEOMBX  00925
      IPARM(6) = 2                                                      GEOMBX  00926
      N = MAX0( NROWSR,NROWSLL)                                         GEOMBX  00927
      CALL WRTEMX(IWTFSC, MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS, K, ID,      GEOMBX  00928
     1            MUAIC, ITYPE, M,N, PARM, IRR)                         GEOMBX  00929
```

```
      IF (IRR .NE. 0) GO TO 8110
      NSCR = NSCR + 1
C         PLACE THE KPT-- ARRAYS ON GEOMETRY SCRATCH AS ONE MATRIX.
 1015 CONTINUE
      IF (NSPATK .EQ. 0) ENDFILE IGEOSC
      IF (NSPATK .EQ. 0) GO TO 2000
      IPARM(3) = NWWK
      IPARM(4) = NTTK
      IPARM(5) = NRWTK
      IPARM(6) = NLWTK
      M = 0
      N = 0
      IF (NWWK .EQ. 0) GO TO 1030
      M = 1
      N = NWWK
      DO 1020 I = 1,NWWK
 1020 KPT(1,I) = KPTWW(I)
 1030 CONTINUE
      IF (NTTK .EQ. 0) GO TO 1050
      M = M + 1
      N = MAX0(N,NTTK)
      DO 1040 I = 1,NTTK
 1040 KPT(M,I) = KPTTT(I)
 1050 CONTINUE
      IF (NRWTK .EQ. 0) GO TO 1070
      M = M + 1
      N = MAX0(N,NRWTK)
      DO 1060 I = 1,NRWTK
 1060 KPT(M,I) = KPTRWT(I)
 1070 CONTINUE
      IF (NLWTK .EQ. 0) GO TO 1090
      M = M + 1
      N = MAX0(N,NLWTK)
      DO 1080 I = 1,NLWTK
 1080 KPT(M,I) = KPTLWT(I)
 1090 CONTINUE
      K = 4
      IVAL = 3HKPT
      CALL WRTEHX(IGEOSC, MXWRIT, RANDOU, NFS,NMS, LS,NMR,LWS, K, ID,
     1            KPT,   ITYPE, M,N, PARM, IRR)
      IF (IRR .NE. 0) GO TO 8070
      END FILE IGEOSC
C
C         MOVE THE MUAIC ARRAYS TO THE GEOMETRY SCRATCH TAPE
      REWIND IWTFSC
      REWIND IVPSC
      IVAL = 6HMUAIC
      K = 2
      IF (NSCR .LE. 0) GO TO 1096
      DO 1094 I = 1,NSCR
      CALL RDINIT
      CALL READMX(IWTFSC, MXWRIT,RANDOU,NFS,NMS,LS,NMR, K, NID,ID,
     1            ITYPE, LRS,  MUAIC,  M,N, PARM, IRR)
      IF (IRR .NE. 0) GO TO 8100
      CALL WRTEHX(IGEOSC, MXWRIT, RANDOU, NFS,NMS, LS,NMR,LWS, K, ID,
     1            MUAIC, ITYPE, M,N, PARM, IRR)
      IF (IRR .NE. 0) GO TO 8070
```

```
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 WRTEMX(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 )                                    FTNX1   00035
     JCHRD = 1                                                         FTNX1   00036
1130 CONTINUE                                                          FTNX1   00037
     JCOL = ICHORD(JCHRD)                                              GEOMBX  01009
     CAPLL = ZLOC(JCHRD)                                               GEOMBX  01010
     YMUVSP = CAPLL*SIN(PSIW)                                          GEOMBX  01011
     IROW = IBOXL(JCHRD)                                               GEOMBX  01012
     GO TO 930                                                         GEOMBX  01013
C        THE LOGIC FOR A 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                                                 FTNX1   00038
     IF (JCHRD .LE. ISMPLW) GO TO 1130                                 FTNX1   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 IWTFSC                                                     GEOMBX  01025
     REWIND IVPSC                                                      GEOMBX  01026
     RETURN                                                            GEOMBX  01027
C                                                                      GEOMBX  01028
C        OUTPUT FORMATS                                                GEOMBX  01029
C        CARD F                                                        GEOMBX  01030
8010 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   ?HDIHEDRAL ANGLE (PSI) / 27X,5HWING , F10.3,4X, F10.3, 8X,     GEOMBX  01034
    3   .2, 8H DEGREES )                                               GEOMBX  01035
8012 FORMAT(27X,5HTAIL , F10.3,4X, F10.3, 8X, F7.2, 8H DEGREES )       GEOMBX  01036
C        CARD G                                                        GEOMBX  01037
8015 FORMAT(1H0,30HCARDG -BOX PATTERN DEFINITION-,5X,6HNCHRDS,10X,     GEOMBX  01038
    1   6HXCENTR,5X,2HCR,4X,5HXEDGE /37X, I3, 8X, F10.4, 6X,F10.4 )    GEOMBX  01039
```

```
 6017 FORMAT(7X,13H-SAMPLE WASH,I3, 8H CHORDS-,5X,6HICHORD,6X,5HIBOXF,
     1   5X,5HIBOXL, 6X, 4HZLOC /(37X, I3,8X, I3, 7X, I3, 6X, F7.2 ) )
C        CARD H
 6021 FORMAT(1H0,40HCARDH -PLANFORM DEFINITION POINT COUNTS-,5X,
     1  12HLEADING EDGE,4X,13HTRAILING EDGE /42X,4HWING, I7, 9X, I7 )
 6022 FORMAT(42X,4HTAIL, I7,9X,I7)
C        CARDS I TO L
 6029 FORMAT(1H0,37HCARDI TO CARDL -PLANFORM DEFINITIONS-,9X,1HX,9X,1HY,
     1   4X,12H(LOCAL AXES) )
 6030 FORMAT( 32X,A10, F9.3, F10.3/ (41X,2F10.3))
 6040 FORMAT(1H0,7X,31H-BOX DIMENSIONS-  B1 (LENGTH) = , E18.8,5X,
     1       17HB1/BETA (WIDTH) =, E18.8)
C
C        CHECK PRINT FORMATS, USED ONLY WHEN CHECPR = .T.
 7010 FORMAT(15H0IWAKE ARRAY - , 40I3 / (15X,40I3) )
 7020 FORMAT(17H0MUAIC ARRAY FOR ,A10, 8H, CHORD I2, 9H, YBAR = F6.3,
     1    7H, EL = F7.2 / 10X, 3HROW,I3,2I4 / (16X,2I4)  )
 7030 FORMAT(51H0  IJALPH (= J*1000 + I OCTAL) AND ALPHA ARRAYS, AS
     1   7H STORED / 6(  5X,13HIJALPH  ALPHA ) / (6(5X,O6, F7.4) )  )
 7040 FORMAT(21H0 CHECK PRINT, FEXLOC/(10F12.7))
 7045 FORMAT( 1H0 14X, 6HTEXLOC / (10F12.7) )
C
C        DIAGNOSTIC FORMATS
 9010 FORMAT(52H0*** WARNING - XEDGE AND XCENTR WERE BOTH SPECIFIED.
     1  20H XEDGE WILL BE IGNORED *** )
 9020 FORMAT(45H0*** WARNING - SAMPLE WASH SPECIFICATION SET I2,6H IS IN
     1  55H ERROR.  ONLY THE PRECEDING ONES WILL BE CALCULATED *** )
 9030 FORMAT(53H0*** WARNING - SAMPLING OF UPWASHES CANNOT BE DONE IF
     1     35H A TAIL HAS BEEN DEFINED.  ISMPLW =,I3,16H WILL BE IGNORED
     2     4H *** )
 9110 FORMAT(23H0*** ERROR - PARAMETER ,A6,23H WAS NOT SPECIFIED.  IT
     1     25H MUST ALWAYS BE GIVEN *** )
 9120 FORMAT(53H0*** ERROR - EITHER XEDGE OR XCENTR MUST BE SPECIFIED
     1     4H *** )
 9130 FORMAT(13H0*** ERROR - ,A6,29H IS OUTSIDE ALLOWED RANGE *** )
 9150 FORMAT(13H0*** ERROR - ,A10,23HDEFINITION POINTS ERROR, I3,
     1     25H, A COMBINATION OF-   *** / 13X,24H1, NON-MONOTONIC Y-VALUE
     2      1HS, 10X,25H2, NON-MONOTONIC X-VALUES  / 13X,11H4, FIRST Y-
     3     14HVALUE NON-ZERO, 10X,34H8, TIP T.E. Y-VALUE DISAGREES WITH
     4     15H TIP L.E. VALUE )
 9160 FORMAT(51H0*** ERROR - XCENTR NOT WITHIN 50 BOX LENGTHS (B1 = ,
     1     E15.8,20H) OF THE WING L.E. (,E15.8, 5H) *** )
 9170 FORMAT(52H0*** ERROR - WHILE WRITING ON GEOMETRY SCRATCH FILE A10,
     1  15H, ERROR CODE = I4, 4H *** )
 9172 FORMAT(14X, 6HARRAY ,A6,15H, DIMENSIONED (I4,1H,I4,11H) WAS BEING
     1   8H WRITTEN )
 9175 FORMAT(14X,20HTHE MUAIC ARRAY FOR A10,15H, DIMENSIONED (I4,1H,I4,
     1  19H) WAS BEING WRITTEN )
 9180 FORMAT(52H0*** ERROR - THE TAIL AND WING, OR THEIR DIAPHRAGMS,
     1  20H CROSS - ABOVE TO BELOW *** )
 9190 FORMAT(54H0*** ERROR - WHILE READING FROM GEOMETRY SCRATCH FILE
     1     A10,15H, ERROR CODE = I4, 4H ***)
 9192 FORMAT(14X, 6HARRAY ,A6,15H, DIMENSIONED (I4,1H,I4,11H) WAS BEING
     1    5H READ )
C
C        ERRORS - ALL ERRORS CALL FLUSH
 8010 CONTINUE
```

```
      WRITE (NT6,9110) IVAL                                    GEOMBX 01097
      GO TO 8900                                               GEOMBX 01098
 8015 CONTINUE                                                 GEOMBX 01099
      WRITE (NT6,9130) IVAL                                    GEOMBX 01100
      GO TO 8900                                               GEOMBX 01101
 8020 CONTINUE                                                 GEOMBX 01102
      WRITE (NT6,9120)                                         GEOMBX 01103
      GO TO 8900                                               GEOMBX 01104
 8030 CONTINUE                                                 GEOMBX 01105
      WRITE (NT6,9130) IVAL                                    GEOMBX 01106
      GO TO 8900                                               GEOMBX 01107
 8040 CONTINUE                                                 GEOMBX 01108
      GO TO 8900                                               GEOMBX 01109
 8050 CONTINUE                                                 GEOMBX 01110
      WRITE (NT6,9150) IVAL,IRR                                GEOMBX 01111
      GO TO 8900                                               GEOMBX 01112
 8060 CONTINUE                                                 GEOMBX 01113
      WRITE (NT6,9160) B1, XEDGEW                              GEOMBX 01114
      GO TO 8900                                               GEOMBX 01115
 8070 CONTINUE                                                 GEOMBX 01116
      WRITE (NT6,9170) IGEOSC, IRR                             GEOMBX 01117
      WRITE (NT6,9172) IVAL, M,N                               GEOMBX 01118
      GO TO 8900                                               GEOMBX 01119
 8075 CONTINUE                                                 GEOMBX 01120
      WRITE (NT6,9170) IVPSC,IRR                               GEOMBX 01121
      WRITE (NT6,9175) IVAL, M,N                               GEOMBX 01122
      GO TO 8900                                               GEOMBX 01123
 8080 CONTINUE                                                 GEOMBX 01124
      WRITE (NT6,9180)                                         GEOMBX 01125
      GO TO 8900                                               GEOMBX 01126
 8090 CONTINUE                                                 GEOMBX 01127
      WRITE (NT6,9190) IVPSC,IRR                               GEOMBX 01128
      WRITE (NT6,9192) IVAL, M,N                               GEOMBX 01129
      GO TO 8900                                               GEOMBX 01130
 8100 WRITE (NT6,9190) IWTFSC,IRR                              GEOMBX 01131
      WRITE (NT6,9192) IVAL, M,N                               GEOMBX 01132
      GO TO 8900                                               GEOMBX 01133
 8110 WRITE (NT6,9170) IWTFSC,IRR                              GEOMBX 01134
      WRITE (NT6,9175) IVAL, M,N                               GEOMBX 01135
 8900 CALL FLUSH(1)                                            GEOMBX 01136
      END                                                      GEOMBX 01137
```

```
      SUBROUTINE EDGCHK(XEDGE,YEDGE,NEDGE,IEDGE,IRR)
C
C        CHECKS FOR MONOTONIC EDGE VALUES OF X FOR LEADING EDGES, AND
C        Y FOR EITHER LEADING OR TRAILING.  CHECKS Y FIRST AND LAST
C        VALUES TO ENSURE DEFINITION FROM CENTERLINE TO TIP
C              XEDGE  = X-VALUES FOR ONE PLANFORM EDGE
C              YEDGE  = Y-VALUES
C              NEDGE  = NUMBER OF (XEDGE,YEDGE) SETS DEFINING THE PLAN-
C                         FORM EDGE
C              IEDGE  = 1 FOR LEADING EDGE, 2 FOR TRAILING EDGE
C              IRR    = ERROR RETURN, 0 SUCCESSFUL
C                         1, NON-MONOTONIC Y-VALUES
C                         2, NON-MONOTONIC X-VALUES, LEADING EDGE ONLY
C                         4, FIRST Y-VALUE NON-ZERO
C                         8, TIP Y-VALUE OF A T.E. DISAGREES WITH PREVIOUS
C                            L.E. TIP VALUE
C
      DIMENSION XEDGE(1),YEDGE(1)
C
      IRR = 0
      IF (YEDGE(1) .NE. 0. ) IRR = 4
      DO 100 I = 2,NEDGE
      IF (YEDGE(I) .LT. YEDGE(I-1) ) GO TO 150
      GO TO (50,100), IEDGE
   50 IF (XEDGE(I) .LT. XEDGE(I-1) ) GO TO 200
  100 CONTINUE
      GO TO 250
  150 IRR = IRR + 1
      GO TO 250
  200 IRR = IRR + 2
  250 CONTINUE
      GO TO (300,350),IEDGE
  300 YTIP = YEDGE(NEDGE)
      GO TO 500
  350 IF (YEDGE(NEDGE) .NE. YTIP) IRR = IRR + 8
C
  500 RETURN
C
      END
```

```
      SUBROUTINE EDGCHK(XEDGE,YEDGE,NEDGE,IEDGE,IRR)
C
C         CHECKS FOR MONOTONIC EDGE VALUES OF X FOR LEADING EDGES, AND
C         Y FOR EITHER LEADING OR TRAILING.  CHECKS Y FIRST AND LAST
C         VALUES TO ENSURE DEFINITION FROM CENTERLINE TO TIP
C              XEDGE  = X-VALUES FOR ONE PLANFORM EDGE
C              YEDGE  = Y-VALUES
C              NEDGE  = NUMBER OF (XEDGE,YEDGE) SETS DEFINING THE PLAN-
C                       FORM EDGE
C              IEDGE  = 1 FOR LEADING EDGE, 2 FOR TRAILING EDGE
C              IRR    = ERROR RETURN, 0 SUCCESSFUL
C                       1, NON-MONOTONIC Y-VALUES
C                       2, NON-MONOTONIC X-VALUES, LEADING EDGE ONLY
C                       4, FIRST Y-VALUE NON-ZERO
C                       8, TIP Y-VALUE OF A T.E. DISAGREES WITH PREVIOUS
C                          L.E. TIP VALUE
C
      DIMENSION XEDGE(1),YEDGE(1)
C
      IRR = 0
      IF (YEDGE(1) .NE. 0. ) IRR = 4
      DO 100 I = 2,NEDGE
      IF (YEDGE(I) .LT. YEDGE(I-1) ) GO TO 150
      GO TO (50,100), IEDGE
   50 IF (XEDGE(I) .LT. XEDGE(I-1) ) GO TO 200
  100 CONTINUE
      GO TO 250
  150 IRR = IRR + 1
      GO TO 250
  200 IRR = IRR + 2
  250 CONTINUE
      GO TO (300,350),IEDGE
  300 YTIP = YEDGE(NEDGE)
      GO TO 500
  350 IF (YEDGE(NEDGE) .NE. YTIP) IRR = IRR + 8
C
  500 RETURN
C
      END
```

```
      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
      DO 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
```

```
      SUBROUTINE NCODER(IBOX,LBOX, IA,JA, IL, ICODE )
      DIMENSION IBOX(LBOX,1)
C
C        ENCODES ONE INTEGER BOX CODE INTO THE PACKED CODE ARRAY ALONG A
C        PORTION OF A CHORD, REPLACING PREVIOUS VALUES
C
C         IBOX   - ARRAY OF BOX CODES,PACKED 20 PER WORD
C         LBOX   - ROW DIMENSION OF BOX CODES ARRAY
C         IA     - I-TH INDEX OF FIRST BOX CODE TO SET
C         JA     - J-TH INDEX OF FIRST BOX CODE TO SET
C         IL     - I-TH INDEX OF THE LAST BOX CODE TO SET
C         ICODE  - CODE VALUE, 1,2, OR 3, FOR THE
C
      INTEGER SHIFT
      DATA MASK / 77777777777777777770B /
      DATA NBWRD /20/
      JSB = (JA-1)/NBWRD + 1
      JB = (NBWRD - MOD(JA,NBWRD) ) * 3
      IF (JB .EQ. 60) JB = 0
C         JB  =  A LEFT SHIFT COUNT
      ICOD  = SHIFT(ICODE,JB)
      IJMASK = SHIFT(MASK,JB)
      DO 100 II = IA,IL
      IJCODE = IJMASK .AND. IBOX(II,JSB)
  100 IBOX(II,JSB) = IJCODE .OR. ICOD
      RETURN
      END
```

```
      SUBROUTINE PRNTBC(IBOX,LBXCD, IFRST,ILAST, JLAST, SUBD)           PRNTBC  00002
      DIMENSION IBOX(LBXCD,1),ICODE(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/ 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 /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUTP,         FILES   00002
     1                IOUFSP,MODESC,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 /                                                  FTNX1   00040
 6001 FORMAT(1H1, 5X,8A10/1H0,20X,16HBOX CODE PATTERN )                 PRNTBC  00023
 6002 FORMAT(19X,20HFOR SUBDIVIDED BOXES,25X, 6HIXBW =,I2,11H  (SUBDIVID PRNTBC  00024
     1  30HED ROW OF UNSUBDIVIDED CENTER)  )                            PRNTBC  00025
 6005 FORMAT(22X,4HMACH,F11.7,56X,*CODE - 1 = PLANFORM BOX* /           PRNTBC  00026
     1 19X,20(1H-),61X,*2 = DIAPHRAGM BOX* /100X,*3 = WAKE BOX * )      PRNTBC  00027
 6010 FORMAT(1H0,4X,31I4 / ( 9X,30I4) )                                 PRNTBC  00028
 6012 FORMAT(1H )                                                       PRNTBC  00029
 6020 FORMAT(1X,I3,2X,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, ICODE )         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) ICODE(J) = BCD(1)                               PRNTBC  00047
      IF(ICODE(J).EQ.1) ICODE(J) = BCD(2)                               PRNTBC  00048
      IF(ICODE(J).EQ.2) ICODE(J) = BCD(3)                               PRNTBC  00049
```

```
      IF(ICODE(J).EQ.3) ICODE(J) =BCD(4)                    PRNTBC  00050
  200 CONTINUE                                              PRNTBC  00051
      WRITE (NT6,6020) IROW, (ICODE(J), J = 1,JLAST )       PRNTBC  00052
  250 CONTINUE                                              PRNTBC  00053
      RETURN                                                PRNTBC  00054
      END                                                   PRNTBC  00055
```

```
      SUBROUTINE BXCDPF(XLE,YLE,NLE,XTE,YTE,NTE, LSROWS,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              LSROWS = 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
      COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUTP,         FILES   00002
     1                IOUFSP,MODESC,IVPSC,IGEOSC,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,PSIT,MXBT,MYBT,MYBBT,MXBST,MYBST,       GEOM2   00002
     1                MYBBST,IXBT,IXBST,CAPL                            GEOM2   00003
      COMMON /EDGES / FEXLOC(250), TEXLOC(250),JDIAG                    EDGES   00002
C                                                                       BXCDPF  00024
      LOGICAL WING                                                      BXCDPF  00025
      DIMENSION XLE(1),YLE(1),XTE(1),YTE(1)                             BXCDPF  00026
      DIMENSION IBOX(LSROWS,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              MXBSW  = NUMBER OF SUBDIVIDED ROWS TO AFT END OF (WING)  BXCDPF  00042
C              MXBST                                            (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              FEXLOC = ARRAY OF (LEADING)  EDGE X-LOCATIONS, SUBDIVIDED BXCDPF 00050
C              TEXLOC             (TRAILING)                            BXCDPF  00051
C                                                                       BXCDPF  00052
```

```
      MXBS = 0
      DEL = 1.0/XSUBDV
C        LOCATION OF FIRST CHORD
      YMIN = .5*(1.0 + DEL)
C        SLOPES OF FIRST LEADING AND TRAILING EDGE SEGMENTS
      XREFLE = XLE(1)
      YREFLE = YLE(1)
      XREFTE = XTE(1)
      YREFTE = YTE(1)
      DELE   = (XLE(2)-XREFLE) / (YLE(2)-YREFLE)
      DELTE  = (XTE(2)-XREFTE) / (YTE(2)-YREFTE)
      ILE = 2
      ITE = 2
      IERR = 0
C
C        WAS THIS CALL FOR WING OR TAIL -
      IF (IXBW .EQ. 0) GO TO 120
C        SET UP COUNTERS FOR TAIL
      SURF = 4HTAIL
      WING = .F.
      MYBT    = IFIX(YLE(NLE))
      NSCHRD  = MYBT * NSUBDV
      JEXLOC  = MYBW * NSUBDV + 1
      IXB = IXBST
      LSRR = LSROWS
      IF (.NOT. COPLAN) LSRR = LSRR + IXBST - 1
      GO TO 130
C
C        SET UP COUNTERS FOR WING
  120 CONTINUE
      SURF = 4HWING
      WING = .T.
      NSCHRD = MYBW * NSUBDV
      JEXLOC = 1
      XMIN = XREFLE + (YMIN-YREFLE)*DELE
      IXBW = (1.-XMIN)*XSUBDV + 1
      IXB  = 1
      LSRR = LSROWS
C
C        START LOOP ON SUBDIVIDED CHORDS
  130 CONTINUE
      YCHORD = YMIN
      DO 350 JCHRD = 1,NSCHRD
C        FIND LEADING EDGE OF THIS CHORD
  140 CONTINUE
C        IS THE CURRENT L.E. SEGMENT STILL GOOD -
      IF (YCHORD - YLE(ILE)) 180,170,150
C        NO, ANOTHER SEGMENT IS NEEDED
  150 CONTINUE
      YREFLE = YLE(ILE)
      ILE = ILE + 1
C        CHECK FOR EXCEEDING LIMIT
      IF (ILE .GT. NLE) GO TO 710
C        CHECK FOR EDGE SEGMENT PARALLEL TO (SKIP THE SEGMENT) OR
C        CUTTING BACK TOWARD CENTER-LINE (ERROR)
      IF (YREFLE - YLE(ILE)) 160,150,730
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
C         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
C         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
C         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
C         SET BOX CODES TO 1 FOR PLANFORM BOXES OF THIS CHORD        BXCDPF  00151
  300 CONTINUE                                                       BXCDPF  00152
      CALL NCODER(IBOX,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 (WING) GO TO 360                                            BXCDPF  00161
      MXBST = MXBS                                                   BXCDPF  00162
      MXBT = (MXBS-IXBW)/NSUBDV + 1                                  BXCDPF  00163
      MYBST = NSCHRD                                                 BXCDPF  00164
      GO TO 370                                                      BXCDPF  00165
  360 MXBSW = MXBS                                                   BXCDPF  00166
```

```
      MXBW = (MXBS-IXBW)/NSUBDV + 1
      MYBSW = NSCHRD
  370 CONTINUE
      IF (IERR .NE. 0) GO TO 750
C
      RETURN
C
C         ERROR DIAGOSTICS
C
C         LIMIT EXCEEDED
  710 IERR = 1
      EDGE = 8H LEADING
      ISEC = ILE - 1
      GO TO 750
  720 IERR = 1
      EDGE = 8HTRAILING
      ISEC = ITE - 1
      GO TO 750
C         BAD EDGE DEFINITION
  730 IERR = 2
      EDGE = 8H LEADING
      ISEC = ILE - 1
      GO TO 750
  740 IERR = 2
      EDGE = 8HTRAILING
      ISEC = ITE - 1
C
  750 CONTINUE
      WRITE(NT6,7500) SURF, EDGE
 7500 FORMAT(22H0*** ERROR PROCESSING ,A5,10HGEOMETRY, ,A8, 9H EDGE ***)
      GO TO (755,760,800) IERR
  755 WRITE(NT6,7550) ISEC
 7550 FORMAT(5X, 8HSECTION ,I2,24H IS BEYOND THOSE DEFINED )
      GO TO 800
  760 WRITE(NT6,7600) ISEC
 7600 FORMAT(5X, 8HSECTION ,I2,36H OF THE EDGE DOUBLES BACK TOWARD THE
     1  12H CENTER LINE )
      GO TO 800
C         PLANFORM EXCEEDS BOX PATTERN LIMIT
  770 CONTINUE
      IERR = 3
      EDGE = 8HTRAILING
      ISEC = ITE - 1
      WRITE (NT6,7700) ISEC,SURF,JCHRD,IEND
 7700 FORMAT(20H *** ERROR - SECTION,I3,29H OF THE TRAILING EDGE OF THE
     1    A4,14H CAUSES CHORD I3,14H TO GO TO ROW I3,15H, WHICH EXCEEDS
     2    14H THE LIMIT *** )
      IEND = LSROWS
C         GO BACK TO FINISH THE SURFACE, THEN PRINT PLANFORM AND FLUSH
      GO TO 300
C
  800 CALL PRNTBC(IBOX,LSROWS, IXB,MXBS, NSCHRD, .T.)
C
 8000 CALL FLUSH(1)
C
      END
```

```
      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,NSPAIC,NOUTP,         FILES  00002
     1                IOUFSP,MODESC,IVPSC,IGEOSC,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,PSIT,MXBT,MYBT,MYBBT,MXBST,MYBST,       GEOM2  00002
     1                MYBBST,IXBT,IXBST,CAPL                            GEOM2  00003
      COMMON /SAMPLW/ ISMPLW,ICHORD(10),IBOXF(10),IBOXL(10),ZLOC(10)    SAMPLW 00002
C        COMMON INPUT VALUES-                                           BXCDI  00015
C            MXBBSW, MXBBST,IXBST, MYBSW,MYBST,MYBBSW,MYBBST, NSUBDV    BXCDI  00016
C            ISMPLW                                                     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*NSUBDV + 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,LSROWB, 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., ICOD )                BXCDI  00049
      ICODE(1) = ICODE(2)                                               BXCDI  00050
      ICODE(2) = ICODE(3)                                               BXCDI  00051
```

```
      CALL DCODER(IBOX,LSROWS, I+1,J-1, I+1,J-1, .T., ICODE(3))         BXCDI
      IF (ICOD .NE. 0) GO TO 120                                         BXCDI
      IF (ICODE(1) .EQ. 0 .OR. ICODE(3) .EQ. 0) GO TO 120                BXCDI
      CALL NCODER(IBOX,LSROWS, I,J, I, 2 )                               BXCDI
  120 CONTINUE                                                           BXCDI
  130 CONTINUE                                                           BXCDI
C         END OF DOUBLE LOOP TO DETERMINE LEADING EDGE DIAPHRAGM AREAS    BXCDI
C                                                                        BXCDI
C         DETERMINE TRAILING EDGE (WAKE) DIAPHRAGM                       BXCDI
      IWK = 0                                                            BXCDI
      DO 180 J = 1,MYBS                                                  BXCDI
      IF (WING) IWK = IWAKE(J)                                           BXCDI
      CALL DCODER(IBOX,LSROWS, IXBS,J, IXBS,J, .T., ICOD)                BXCDI
      DO 170 I = IXBS1,MXBBS                                             BXCDI
      ICODM1 = ICOD                                                      BXCDI
      CALL DCODER(IBOX,LSROWS, I,J, I,J, .T., ICOD)                      BXCDI
      IF (ICOD .EQ. 1) GO TO 170                                         BXCDI
      IF (ICODM1 .NE. 1 .AND. ICODM1 .NE. 3) GO TO 170                   BXCDI
      IF (I .LE. IWK) GO TO 160                                          BXCDI
      IF (I .EQ. MXBBS) GO TO 180                                        BXCDI
C         THE BOX IS A CANDIDATE.  SEARCH DIAGONALLY FOR POSSIBLE        BXCDI
C         RECEIVING BOXES DOWNSTREAM.                                    BXCDI
      JP = J                                                             BXCDI
      JM = J                                                             BXCDI
      IS = I+1                                                           BXCDI
      DO 150 II = IS,MXBBS                                               BXCDI
      IF (JM .GT. 1) JM = JM - 1                                         BXCDI
      CALL DCODER(IBOX,LSROWS, II,JM, II,JM, .T., ICOD )                 BXCDI
      IF (ICOD .NE. 0) GO TO 160                                         BXCDI
      IF (JP .GE. MYBBS) GO TO 150                                       BXCDI
      JP = JP + 1                                                        BXCDI
      CALL DCODER(IBOX,LSROWS, II,JP, II,JP, .T., ICOD )                 BXCDI
      IF (ICOD .NE. 0) GO TO 160                                         BXCDI
      IF (.NOT. WING)  GO TO 150                                         BXCDI
      IF ( II .LE. IWAKE(JM) ) GO TO 160                                 BXCDI
      IF ( II .LE. IWAKE(JP) ) GO TO 160                                 BXCDI
  150 CONTINUE                                                           BXCDI
C         END OF LOOP ON DIAGONAL SEARCH                                 BXCDI
      GO TO 170                                                          BXCDI
C                                                                        BXCDI
C         CONDITIONS HAVE BEEN FOUND FOR A VALID WAKE BOX                BXCDI
  160 CONTINUE                                                           BXCDI
      CALL NCODER(IBOX,LSROWS, I,J, I, 3 )                               BXCDI
      ICOD = 3                                                           BXCDI
  170 CONTINUE                                                           BXCDI
C         END OF LOOP ON ROWS, AND                                       BXCDI
  180 CONTINUE                                                           BXCDI
C         END OF LOOP ON CHORDS, FOR WAKE DIAPHRAGM, FROM 130+           BXCDI
C                                                                        BXCDI
C         DETERMINE THE TIP DIAPHRAGM REGION                             BXCDI
      LBB = 2                                                            BXCDI
      DO 300 I = IXBS1,MXBBS1                                            BXCDI
C         SEARCH FOR LAST NON-ZERO BOX CODE ON THE ROW, FROM LBB OUTWARD BXCDI
      CALL DCODER(IBOX,LSROWS, I,LBB, I,MYBBS,  .T., ICODE(LBB) )        BXCDI
  200 LBBP1 = LBB+1                                                      BXCDI
      DO 2?0 J = LBBP1,MYBBS                                             BXCDI
      IF (ICODE(J) .EQ. 0) GO TO 220                                     BXCDI
```

```
  210 CONTINUE
      LBB = MYBBS
      GO TO 225
  220 LBB = J - 1
C           LBB  = THE SUBSCRIPT OF THE LAST NON-ZERO BOX ON THE ROW
C           TEST BOX DIRECTLY AHEAD FOR NON-ZERO VALUE
  225 CONTINUE
      CALL DCODER(IBOX,LSROWS, I-1,LBB, I-1,LBB, .T., ICOD)
      IF (ICOD .EQ. 0) GO TO 280
C           SEARCH FOR A NON-ZERO CODE ON THE INBOARD DIAGONAL AFT
      J = LBB
      IP1 = I + 1
      IF (IP1 .GT. MXBBS) GO TO 310
      DO 230 II = IP1,MXBBS
      CALL DCODER(IBOX,LSROWS, II,J, II,J, .T.,  ICOD)
      IF (ICOD .NE. 0) GO TO 255
      J = J - 1
  230 CONTINUE
C
C           NO DIAGONAL BOX WAS FOUND
      IF (NSURF .EQ. 1 .AND. ISMPLW .EQ. 0) GO TO 310
      IF (.NOT. WING) GO TO 310
C           SEARCH BACK ALONG THE DIAGONAL FOR   A  DIAPHRAGM REGION
C           CAUSED BY A TAIL SURFACE
      II  = MXBBS + 1
      DO 240 III = IP1,MXBBS
      II  =  II - 1
      J = J + 1
      IF (IWAKE(J)  .GE. II ) GO TO 250
  240 CONTINUE
C           NO DIAPHRAGM FOUND
      GO TO 310
C
C           CONDITION FOUND REQUIRING DIAPHRAGM BOXES ON THE DIAGONAL.
  250 JJ = J
      GO TO 260
  255 JJ = J + 1
      II = II - 1
  260 CONTINUE
C           TEST FOR EXCEEDING BOX CODE ARRAY
      LBB = LBB + 1
      IF (LBB .GT. LSCHDS) GO TO 8500
C           SET DIAGONAL ELEMENTS
      DO 270 J = JJ,LBB
      CALL NCODER(IBOX,LSROWS, II,J, II, 2 )
      II = II - 1
  270 CONTINUE
      ICODE(LBB) = 2
      MYBBS = MAX0(MYBBS,LBB)
      GO TO 200
C
C           DETERMINE LAST NON-ZERO BOX ON NEXT ROW
  280 CONTINUE
      MYBBS = MAX0(MYBBS,LBB)
      II = I + 1
      DO 290 K = 1,LBB
      J = LBB - K + 1
```

```
      CALL DCODER(IBOX,LSROWS, II,J, II,J, .T., ICOD)                  BXCDI
      IF (ICOD .NE. 0) GO TO 295                                       BXCDI
  290 CONTINUE                                                         BXCDI
      GO TO 300                                                        BXCDI
  295 LBB = J                                                          BXCDI
  300 CONTINUE                                                         BXCDI
C         END OF LOOP ON ROWS DETERMINING TIP DIAPHRAGM CODES, FROM 180* BXCDI
C                                                                      BXCDI
  310 CONTINUE                                                         BXCDI
      IF (WING) GO TO 350                                              BXCDI
      MYBBST = MYBBS                                                   BXCDI
      MYBBT  = (MYBBS+NSUBD2)/NSUBDV                                   BXCDI
      GO TO 500                                                        BXCDI
  350 MYBBSW = MYBBS                                                   BXCDI
      MYBBW  = (MYBBS+NSUBD2)/NSUBDV                                   BXCDI
      IF (.NOT. COPLAN) GO TO 500                                      BXCDI
      MYBBST = MYBBSW                                                  BXCDI
      MYBBT = MYBBW                                                    BXCDI
C                                                                      BXCDI
  500 RETURN                                                           BXCDI
C                                                                      BXCDI
 8500 WRITE (NT6,9500)                                                 BXCDI
 9500 FORMAT(51H0*** ERROR - TOO MANY CHORDS FOR BOX CODE ARRAY *** )  BXCDI
      CALL PRNTBC(IBOX,LSROWS,IXBS,MXBBS,MYBBS, .T. )                  BXCDI
      CALL FLUSH(1)                                                    BXCDI
C                                                                      BXCDI
      END                                                              BXCDI
```

```
      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/ 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,CARL                              GEOM2   00003
      COMMON /PLANXY/ NWLE,NWTE,NTLE,NTTE, XWLE(10),YWLE(10),             PLANXY  00002
     1                XWTE(10),YWTE(10), XTLE(10),YTLE(10),               PLANXY  00003
     2                XTTE(10),YTTE(10)                                   PLANXY  00004
      COMMON /EDGES / FEXLOC(250), TEXLOC(250),JDIAG                      EDGES   00002
C                                                                         GMAREA  00016
C              COMMON PARAMETERS USED                                     GMAREA  00017
C       MXB    = LENGTH OF BOX PATERN (X-DIRECTION)                       GMAREA  00018
C       MYB    = MAXIMUM ON-PLANFORM SPAN (Y-DIRECTION)                   GMAREA  00019
C       COPLAN = .T., SECOND SURFACE EXISTS FOR PLANFORM                  GMAREA  00020
C              = .F., SINGLE SURFACE                                      GMAREA  00021
C       NWLE   = NUMBER OF POINTS DEFINING LEADING EDGE OF THE WING       GMAREA  00022
C       NWTE   = 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       XWLE   = X COORDINATE OF THE LEADING EDGE DEFINITION POINT        GMAREA  00026
C                FOR THE FIRST PLANFORM                                   GMAREA  00027
C       YWLE   = 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 IBX(50)                                                   GMAREA  00044
      DIMENSION ALPHA(1), IJALPH(1)                                       GMAREA  00045
      LOGICAL WING                                                        GMAREA  00046
      REAL LIN1,LIN2,LIN3,LIN4,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
      CEN2 = 0.
      GO TO 8
    5 MYB = MYBW
      IF (COPLAN) MYB = MAX0(MYB,MYBT)
      IXB = 1
      MXB = MXBW
      IF (COPLAN) MXB = MXBT
    8 CONTINUE
C
C         LOOP ON CHORDS
      DO 90 J=1,MYB
      YJ = J
      RIGHT = YJ + 0.5
      LEFT  = YJ - 0.5
      IF (.NOT. WING) GO TO 110
C
C         CALL NTRCEP TO DETERMINE LEFT, RIGHT AND CENTER LINE
C         INTERCEPTS, AND THE BREAK POINTS OF EDGES OVER THIS
C         CHORD.
C
      IF (YJ .GT. YWLE(NWLE) ) GO TO 10
      CALL NTRCEP(J, YWLE,XWLE, LIN1,CEN1,RIN1,NBK1,KINK1, 1)
      CALL NTRCEP(J, YWTE,XWTE, LIN2,CEN2,RIN2,NBK2,KINK2, 2)
   10 IF (NSURF .EQ. 1 .OR. .NOT. COPLAN) GO TO 20
C     COMPUTE SLOPE AND INTERCEPTS FOR SECOND PLANFORM.
  110 IF (YJ .GT. YTLE(NTLE) ) GO TO 20
      CALL NTRCEP(J, YTLE,XTLE, LIN3,CEN3,RIN3,NBK3,KINK3, 1)
      CALL NTRCEP(J, YTTE,XTTE, LIN4,CEN4,RIN4,NBK4,KINK4, 2)
   20 CONTINUE
C     SLOPE AND INTERVALS COMPLETED.
C
C     LOOP DOWN THE CHORD
      CALL DCODER(IBOX,LBXCD, IXB,J, MXB,J, .F., IBX)
      II = 1
      DO 85 I=IXB,MXB
      XI = I
      IF(IBX(II) .NE. 1) GO TO 80
      BOXLE = XI - 0.5
      BOXTE = XI + 0.5
      IF(.NOT.WING) GO TO 40
      IF (YJ .GT. YWLE(NWLE) ) GO TO 40
      IF (XI .GT.  CEN2)      GO TO 40
C     BOX IS ON PLANFORM 1
      ICODE =1
C          ICODE  = 1, 1ST L.E. BOX ON CHORD
C                 = 2, LAST T.E. BOX ON CHORD
C                 = 3, INTERNAL CUT BOX
      IF(II .EQ. 1) GO TO 24
      IF(IBX(II-1) .NE. 1) GO TO 24
      ICODE =2
      IF (I.EQ.MXB) GO TO 24
      IF    :(II+1) .NE. 1) GO TO 24
      IF(    ,1. .GT.CEN2) GO TO 24
      ICOC  =3
      IU=0
      IB=0
```

```
      IF(RIN1.GT.BOXLE) IU=1
      IF(RIN2.LT.BOXTE) IB=1
      IF(LIN2.LT.BOXTE) IB=1
      IF (IU+IB.EQ.0) GO TO 80
C     BOX IS NOT ENTIRELY ON PLANFORM. COMPUTE AREA.
   24 CONTINUE
      CALL ALPHAC(XI,   XWLE,YWLE,XWTE,YWTE,
     1LIN1,CEN1,RIN1,NBK1,KINK1,LIN2,CEN2,RIN2,NBK2,KINK2,ALPHA(NALPH))
      GO TO 75
C
   40 CONTINUE
      IF (NSURF .EQ. 1) GO TO 80
C     BOX IS ON PLANFORM 2.
      ICODE =2
      IF(I.EQ.MXB) GO TO 44
      ICODE = 1
      IF(XI-1. .LT.CEN3) GO TO 44
      IF(IBX(II-1) .NE. 1) GO TO 44
      ICODE =2
      IF(IBX(II+1) .NE. 1) GO TO 44
      ICODE =3
      IU=0
      IB=0
      IF(RIN3.GT.BOXLE) IU=1
      IF(RIN4.LT.BOXTE) IB=1
      IF(LIN4.LT.BOXTE) IB=1
      IF (IU+IB.EQ.0) GO TO 80
C     BOX IS NOT ENTIRELY ON PLANFORM. COMPUTE AREA.
   44 CONTINUE
      CALL ALPHAC(XI,   XTLE,YTLE,XTTE,YTTE,
     1LIN3,CEN3,RIN3,NBK3,KINK3,LIN4,CEN4,RIN4,NBK4,KINK4,ALPHA(NALPH))
   75 IJALPH(NALPH) = J*512 + I
      NALPH = NALPH + 1
   80 CONTINUE
      II = II + 1
   85 CONTINUE
   90 CONTINUE
      NALPH = NALPH -1
      RETURN
      END
```

```
      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  ENTER 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
      IF(XC(NX+1).GT.BOXTE) GO TO 250
      B = BOXTE - XC(NX+1)
  200 AREA = AREA + 0.5*(A+B)*DY
      GO TO 300
C
C        EDGE CROSSES BOXTE.  COMPUTE INTERSECTION FOR DY
  250 CONTINUE
      B = 0.0
      DX = XC(NX+1) - XC(NX)
      S = DY/DX
      DY = S*A
      GO TO 200
  300 CONTINUE
      IF (BOXTE-BOXLE.LT.EPS) GO TO 2000
      IF(R2.LT.BOXTE) ISLICE =2
      IF(L2.LT.BOXTE) ISLICE =1
      IF(ISLICE.NE.0) GO TO 5000
      GO TO 4000
C
C        TRAILING EDGE BOX
 2000 CONTINUE
      NTRAPS =NBK2 + 1
      NTM1 = NTRAPS- 1
      NXC  = NTRAPS+ 1
      XC(1)= L2
      YC(1)= LEFT
      XC(NXC)=R2
      YC(NXC)=RIGHT
      IF(NTRAPS.EQ.1) GO TO 2110
      DO 2100 NA = 2,NTRAPS
      KIDX = K2 + NA -2
      XC(NA) = XTED(KIDX)
      YC(NA) = YTED(KIDX)
 2100 CONTINUE
 2110 CONTINUE
      DO 2300 NX=1,NTRAPS
      IF(XC(NX).LT.BOXLE.AND.XC(NX+1).LT.BOXLE) GO TO 2300
      DY = YC(NX+1) - YC(NX)
      IF(DY.LT.EPS) GO TO 2300
      IF(XC(NX).LT.BOXLE.OR.XC(NX+1).LT.BOXLE)  GO TO 2250
C
C        DOES NOT INTERSECT BOXLE
      A = XC(NX) - BOXLE
      B = XC(NX+1) - BOXLE
 2225 AREA = AREA + 0.5*(A+B)*DY
      GO TO 2300
C
C        INTERSECTS BOXLE
 2250 CONTINUE
      DX = XC(NX+1) -XC(NX)
      S = DY/DX
      A = BOXLE - XC(NX)
      YINT = YC(NX) + S*A
      IF(S.LT.0.0) GO TO 2275
C
C        SLOPE POSITIVE
```

```
      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(1) = L2
      YC(1) = LEFT
      GO TO 5110
 5102 ITAG = 2
      I = 1
      XC(1) = R2
      YC(1)=  RIGHT
 5110 I= I+1
      IF(NBK2.EQ.0) GO TO 5150
      KIDX = K2 + I - 2
      IF(I   G.EQ.2) KIDX = K2    +NBK2-1 -I +2
      IF(XTED(KIDX).GT.BOXTE) GO TO 5125
      XC(I) = XTED(KIDX)
      YC(I) = YTED(KIDX)
      GO TO 5110
 5125 CONTINUE
```

```
      IF(ITAG.EQ.2) KIDX=KIDX+1
      XC(I) = BOXTE
      DX = XTED(KIDX) - XTED(KIDX-1)
      DY = YTED(KIDX) - YTED(KIDX-1)
      IF(ITAG.EQ.2) GO TO 5130
C
C          LOWER LEFT CORNER
      DIST = BOXTE -XC(I-1)
      YC(I) = YTED(KIDX-1) + DIST*(DY/DX)
      GO TO 5160
C
C          LOWER RIGHT CORNER
 5130 DIST = BOXTE - XC(I-1)
      YC(I) = YC(I-1)+ DIST* (DY/DX)
      GO TO 5160
 5150 CONTINUE
      XC(I) = BOXTE
      DX = R2 - L2
      DY = RIGHT - LEFT
      IF(ITAG.EQ.2) GO TO 5155
      DIST = BOXTE -L2
      YC(I) = LEFT + DIST*(DY/DX)
      GO TO 5160
 5155 DIST = BOXTE -R2
      YC(I) = RIGHT+ DIST*(DY/DX)
 5160 CONTINUE
      NTRAPS = I-1
      DO 5175 NX=1,NTRAPS
      A= BOXTE - XC(NX)
      B= BOXTE - XC(NX+1)
      IF(A.GT.1.) A = 1.0
      IF(B.GT.1.) B = 1.0
      H= YC(NX+1) - YC(NX)
      IF(ITAG.EQ.2) H = -H
      TA = TA  + 0.5*(A+B)*H
 5175 CONTINUE
      IF(ITAG.EQ.2) GO TO 5020
      GO TO 5010
C
C          COMPUTE FOR UPPER RIGHT HAND CORNER
 5300 CONTINUE
      I = 1
      XC(I) = R1
      YC(I) = RIGHT
 5310 I = I+1
      IF(NBK1.EQ.0) GO TO 5350
      KIDX =   K1  + NBK1-1 -I +2
      IF(XLED(KIDX).LT.BOXLE) GO TO 5325
      XC(I) = XLED(KIDX)
      YC(I) = YLED(KIDX)
      GO TO 5310
 5325 CONTINUE
      XC(I) = BOXLE
      DX = XLED(KIDX+1)-XLED(KIDX)
      DY = YLED(KIDX+1)-YLED(KIDX)
      DIST = BOXLE - XLED(KIDX)
      YC(I)= YLED(KIDX) + DIST*(DY/DX)
```

```
      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
C           A.GT.B1 AND B.LT.B1
      DX = XC(NX) - XC(NX+1)
      DY = YC(NX) - YC(NX+1)
      IF(DY.LT.EPS) GO TO 5375
      DIST = BOXTE - 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
```

```
      SUBROUTINE NTRCEP(J,  YEDG,XEDG,           L1,C1,R1,NBK1,K1,IDEX)  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 CEN1 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(IDEX.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 CEN1                                  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                                                           NTRCEP  00059
    6 DX = XEDG(K) - XEDG(K-1)                                          NTRCEP  00060
      DY = YEDG(K) - YEDG(K-1)                                          NTRCEP  00061
      DIST = RIGHT - YEDG(K-1)                                          NTRCEP  00062
      R1 = XEDG(K-1) + (DX/DY) * DIST                                   NTRCEP  00063
      RETURN                                                            NTRCEP  00064
      END                                                               NTRCEP  00065
```

```
      SUBROUTINE PWMAIC(WING,IBOX,LBOX,IWAKE, JCOL)                     PWMAIC  00002
C                                                                       PWMAIC  00003
C        COMPUTES THE POINTER ARRAY (MUAIC) FOR THE SPATIAL AIC ARRAY   PWMAIC  00004
C        OF THE LEFT WING (TAIL) ON ONE CHORD OF THE RIGHT WING (TAIL)  PWMAIC  00005
C                                                                       PWMAIC  00006
C              WING   = WING/TAIL INDICATOR                             PWMAIC  00007
C              IBOX   = BOX CODE ARRAY TO USE                           PWMAIC  00008
C              IWAKE  = ARRAY OF WAKE EDGE LOCATIONS FOR WING           PWMAIC  00009
C              JCOL   = THE (UNSUBDIVIDED) CHORD NUMBER OF INTEREST     PWMAIC  00010
C              SURF   = INDICATOR OF WHETHER ANY LEFT SURFACE IS INTER- PWMAIC  00011
C                       CEPTED BY THE MACH CONE FOR THIS CORD           PWMAIC  00012
C                                                                       PWMAIC  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                                                                       PWMAIC  00018
      LOGICAL WING                                                      PWMAIC  00019
      DIMENSION IBOX(LBOX,8), ICODE(50), IWAKE(1)                       PWMAIC  00020
      DATA  EPS / 1.0E-4 /                                              PWMAIC  0002'
C                                                                       PWMAIC  00022
      IF (WING) GO TO 100                                               PWMAIC  00023
C                                                                       PWMAIC  00024
C         THE CALL IS FOR A TAIL CHORD                                  PWMAIC  00025
      PSI2 = PSIT + PSIT                                                PWMAIC  00026
      IXB = (IXBT-IXBW)/NSUBDV + 1                                      PWMAIC  00027
      MXB = MXBT                                                        PWMAIC  00028
      IF (JCOL .LE. MYBT) GO TO 120                                     PWMAIC  00029
C         THE CHORD IS ON THE TIP DIAPHRAGM                             PWMAIC  00030
      IFRST = IXB                                                       PWMAIC  00031
      NJ = MXBT-IFRST+1                                                 PWMAIC  00032
      GO TO 130                                                         PWMAIC  00033
C                                                                       PWMAIC  00034
C         THE CALL IS FOR A WING CHORD                                  PWMAIC  00035
  100 CONTINUE                                                          PWMAIC  00036
      PSI2 = PSIW + PSIW                                                PWMAIC  00037
      IXB = 1                                                           PWMAIC  00038
      IF (COPLAN) GO TO 110                                             PWMAIC  00039
      MXB = MXBBW                                                       PWMAIC  00040
      IF (JCOL .GT. MYBW) GO TO 115                                     PWMAIC  00041
      ISUB = JCOL*NSUBDV - NSUBD2                                       PWMAIC  00042
      GO TO 125                                                         PWMAIC  00043
C                                                                       PWMAIC  00044
C         THE CALL IS FOR A COPLANAR WING-TAIL                          PWMAIC  00045
  110 CONTINUE                                                          PWMAIC  00046
      MXB = MXBT                                                        PWMAIC  00047
      IF (JCOL .LE. MYBT) GO TO 120                                     PWMAIC  00048
C         THE CHORD IS ON THE TIP DIAPHRAGM                             PWMAIC  00049
  115 CONTINUE                                                          PWMAIC  00050
      IFRST = 1                                                         PWMAIC  00051
```

```
      NJ = MXB
      GO TO 130
C
C         THE CHORD IS ON PLANFORM
  120 CONTINUE
      ISUB = MYBSW + JCOL*NSUBDV - NSUBD2
  125 CONTINUE
      IFRST = (TEXLOC(ISUB)-IXBW) / NSUBDV + 1
      NJ = MXB - IFRST + 1
C
  130 CONTINUE
      CALL DCODER(IBOX,LBOX, IFRST,JCOL, MXB,JCOL, .F., ICODE)
      IROW = IFRST
      DO 135 I = 1,NJ
      IA = I
      IF (ICODE(I) .NE. 0) GO TO 140
      IROW = IROW + 1
  135 CONTINUE
  140 CONTINUE
      IF (IA .GE. NJ) GO TO 155
      DO 145 I = IA,NJ
      IF (ICODE(I) .EQ. 0) GO TO 150
      IROW = IROW + 1
  145 CONTINUE
  150 CONTINUE
      IROW = IROW - 1
  155 CONTINUE
      NROWS = IROW - IXB + 1
C
C         COMPUTE HORIZONTAL AND VERTICAL OFFSETS
      YMUSND = (JCOL-.5)*COS(PSI2)
C              = Y-OFFSET ON THE SENDING SURFACE OF THE PROJECTION OF THE
C                RECEIVING CHORD
      JBAR   = IFIX(YMUSND) + 1
C              = CHORD CONTAINING YMUSND
      YBAR   = YMUSND - JBAR + .5
C              = DISTANCE FROM NEAREST SENDING CHORD CENTER TO PROJEC-
C                TION OF THE RECEIVING CHORD, POSITIVE RIGHT.
      EL     = (JCOL - .5) * SIN(PSI2)
C              = VERTICAL SEPARATION BETWEEN THE SENDING PLANE AND THE
C                RECEIVING CHORD
      IF (YBAR) 160,165,170
  160 JMIN = JBAR - 1
      NBOXES = 2
      GO TO 180
  165 JMIN = JBAR
      NBOXES = 1
      GO TO 180
  170 JMIN = JBAR
      NBOXES = 2
C
  180 CONTINUE
      JL = 1
      SURF = .F.
C
C         START OF LOOP ON ROWS, FORWARD FROM RECEIVING BOX CENTER, TO
C         DEFINE THE MUAIC ARRAY
```

```
      DO 280 I = 1,NROWS
      XI = I - .5 - ABS(EL)
      IF (XI .LE. EPS) GO TO 260
      IF (JMIN .GT. 0) GO TO 260
C         CENTER LINE HAS BEEN CROSSED, THEREFORE THERE MAY BE CONTRIBU-
C         TION FROM THE LEFT WING FOR THIS ROW
      JM = -JMIN + 1
      JMM = JM-JL+1
      CALL DCODER(IBOX,LBOX, IROW,1, IROW,JMM, .F., ICODE)
      DO 240 J = JL,JM
      IF (ICODE(JMM) .NE. 0) GO TO 250
      JMM = JMM-1
  240 CONTINUE
      NROWS = I - 1
      GO TO 290
C         CONTRIBUTING BOXES HAVE BEEN FOUND FOR THIS ROW
  250 CONTINUE
      SURF = .T.
      JL = J
      IF (YBAR .GE. 0) GO TO 255
      MUAIC(1,I) = NBOXES - JM + 1
      MUAIC(2,I) = NBOXES - JL + 1
      GO TO 270
  255 CONTINUE
      MUAIC(1,I) = JL
      MUAIC(2,I) = JM
      GO TO 270
C
C         CENTER LINE HAS NOT BEEN CROSSED
  260 MUAIC(1,I) = 0
      MUAIC(2,I) = 0
C
  270 CONTINUE
      NBOXES = NBOXES + 2
      JMIN = JMIN - 1
      IROW = IROW - 1
  280 CONTINUE
C         END OF LOOP FORWARD ON ROWS, FROM 180*
C
  290 CONTINUE
      RETURN
      END
```

```
      OVERLAY(AFMBOX,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), PRVGEOM,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,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 /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 /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
      COMMON / MODES/ SYM,SYMT,MTYPEW,MTYPET                           MODCOM   00002
      COMMON /CHECKPR/ DPPCPR,GEOCPR,MODCPR,AICCPR,NWSCPR,SMCPR,GAFCPR CHECKPR  00002
      LOGICAL  DPPCPR, GEOCPR, MODCPR, AICCPR,NWSCPR,  SMCPR, GAFCPR   CHECKPR  00003
      EQUIVALENCE (CHECKPR,MODCPR)                                     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/ IS(100),NOC(100),JS(100),JOC(100)                 MODES    00019
      DIMENSION  IPNTRM(2,100)                                         MODES    00020
      DIMENSION XP(6),YP(6),X1(100),Y1(100)                            MODES    00021
      DIMENSION DOS(50)                                                MODES    00022
      DIMENSION  FEXLOC(250),           TEXLOC(250)                    MODES    00023
C             FEXLOC(MYBSW+MYBST),    TEXLOC SAME )                    MODES    00024
C                                                                      MODES    00026
      LOGICAL MXREAD,RANDIN,MXWRIT,RANDOU                              MODES    00027
      NAMELIST /CARDM / NMODES, NTSLOP                                 FTNX1    00043
      MXREAD = .FALSE.                                                 MODES    00028
      MXWRIT = .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.PRVGEOM) GO TO 50
      WRITE (NT6,7005)
      RETURN
C
   50 CONTINUE
      IF(MTYPEW.EQ.3) GO TO 75
      IF(NSURF.EQ.2.AND.MTYPET.EQ.3) GO TO 75
      WRITE (NT6,7010)
      GO TO 125
C
   75 CONTINUE
      WRITE (NT6,7015)
      CALL FLUSH(1)
C
  100 CONTINUE
C
C
      NMODES = 0
      NTSLOP = 0
      READ(NT5,CARDM)
  125 CONTINUE
      REWIND IGEOSC
C
C        READ FEXLOC AND TEXLOC ARRAY FROM GEOMETRY SCRATCH FILE
      MNAME =6HFEXLOC
      CALL RDINIT
      ITYPE = 5HMIXED
      NMS = 1
      IF(.NOT.COPLAN.AND.NSURF.EQ.2) NMS = 2
      CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,
     1            LRS,FEXLOC, M,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 6010
C
      MNAME =6HTEXLOC
      CALL RDINIT
      ITYPE = 5HMIXED
      CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,
     1            LRS,TEXLOC,M,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 6010
C
C
C        ZERO OUT THE ROW AND COL POINTERS
      DO 150 I = 1,400
      IS(I) = 0
  150 CONTINUE
C
C
C        DETERMINE STARTING BOXES AND NUMBER OF BOXES PER CHORD.
      IYB1 = (NSUBDV+1)/2
      DO 300 NS=1,NSURF
      IF(NS.EQ.2) GO TO 200
      NC =1
      NCH = NYBW
      NCF= 0
      ICN = IXBW - IYB1
      IXB = IXBW
      GO TO 225
```

```
  200 CONTINUE
      NC = MYBW + 1
      NCH = MYBW + MYBT
      NCF = MYBW * NSUBDV
      IFBT = (IXBT-IXBW)/NSUBDV + 1
  225 CONTINUE
      IYB = IYB1 + NCF
      DO 250 J=NC,NCH
      IS(J) = FEXLOC(IYB) + 1.0
      ITEI  = TEXLOC(IYB)
      IF(NSUBDV.EQ.1) GO TO 240
      IS(J) = (IS(J)-ICN)/NSUBDV + 1
      ITEI  = (ITEI-IXB)/NSUBDV  + 1
  240 CONTINUE
      NOC(J) = ITEI-IS(J) + 1
      IYB =  IYB + NSUBDV
  250 CONTINUE
  300 CONTINUE
      CALL ROPER
C
C
C         FIND OVERLAP OF 2 PLANFORMS IF THEY ARE NON-COPLANAR
      IOVLAP = 0
      NPNTRS = MXBW+1
      IF(NSURF.EQ.1) GO TO 325
      IF(COPLAN) GO TO 324
      IF(IFBT.GT.MXBW) GO TO 324
      IOVLAP = MXBW - IFBT + 1
      NPNTRS = MXBT + IOVLAP + 1
      GO TO 325
  324 CONTINUE
      NPNTRS = MXBT + 1
  325 CONTINUE
C         COMPUTE POINTER ARRAY AND STORE ON MODESC
      REWIND MODESC
C
      IPNTRM(1,1) = 1
      IPNTRM(2,1) = JS(1)
      DO 320 I=2,NPNTRS
      IPNTRM(1,I) = IPNTRM(1,I-1) + JOC(I-1)
      IPNTRM(2,I) = JS(I)
  320 CONTINUE
      CALL RDINIT
      IPARM(3) = IOVLAP
      ITYPE = 5HMIXED
      CALL WRTEMX(MODESC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
     1             IPNTRM,ITYPE,2,NPNTRS,PARM,IRR)
      IF(IRR.NE.0) GO TO 6030
C         FIRST LOOP DETERMINES MODE SHAPES.
C         SECOND LOOP DETERMINES THICKNESS SLOPES.
C
      DO 3  0 IPASS=1,2
      IF(IPASS.EQ.2) GO TO 2100
C         LOOP ON NUMBER OF SURFACES
      DO 2000 NS=1,NSURF
      IF(NS.EQ.NSURF) GO TO 330
C
```

```
      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                                               BCSMDA  00002
      DEFSL(1,I) = 0.0                                                  MODES   00163
      DEFSL(2,I) = 0.0                                                  MODES   00164
  350 CONTINUE                                                          MODES   00165
C                                                                       BCSMDA  00003
C         ZERO OUT THE COEFFICIENT ARRAY                                BCSMDA  00004
      DO 355 I = 1,21                                                   BCSMDA  00005
      A(I) = 0.0                                                        BCSMDA  00006
  355 CONTINUE                                                          BCSMDA  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.PRVMODE) GO TO 450                                        MODES   00173
      CALL RDINIT                                                       MODES   00174
      IF(NS.EQ.1.AND.NM.EQ.1) NFS = 2                                   MODES   00175
      MNAME = 6H COEF.                                                  MODES   00176
      CALL READMX(IGEOSC,MXREAD,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
C          OF DEGREE IDEG TO THE POINTS USING METHOD OF LEAST SQUARES.
  502 CONTINUE
      READ (NT5,8010) IDEG,NPTS
      IF (IDEG .LT. 0 .OR. IDEG .GT. 5) GO TO 6005
      IF (NPTS .GT. 100 .OR. NPTS .LT. 1) GO TO 6005
      READ (NT5,8020) (XX(I),YY(I),ZZ(I),I=1,NPTS)
 8020 FORMAT(6E10.0)
      IDIM = 1
      CN = 1.0
C
C          CN IS A SCALE FACTOR TO REDUCE THE MAGNITUDE OF THE NUMBERS
C          IDIM IS A DIMENSION VARIABLE SET TO 1 TO INDICATE FIT IS
C          BEING MADE ON REAL VALUES . IDIM = 2 FOR COMPLEX Z VALUES.
      CALL FITTER(IDEG,NPTS,XX,YY,ZZ,A,CN,IDIM)
C
      IFLAG = 2
      MDEG = IDEG + 1
      DEG = IDEG + 1
      DEG2 = DEG/2.
      NC = DEG*DEG2 + DEG2 + EPS
C
C
  550 CONTINUE
C
C          STORE THE COEFFICIENTS ON THE THIRD FILE OF THE IGEOSC FILE.
C          IF THE COEFFICIENTS ARE TO BE PRINTED THE ONES FOR THE FIRST
C          SURFACE MUST BE STORED ON A SCRATCH FILE TEMPORARILY.
C
      CALL RDINIT
      IF(NS.EQ.1.AND.NM.EQ.1) NFS = 2
      IPARM(3) = IDEG
      IPARM(4) = IFLAG
      ITYPE = 5HMIXED
      CALL WRTEMX(IGEOSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,1,ID,
     1            A,ITYPE,1,NC,PARM,IRR)
      IF(IRR.NE.0) GO TO 6050
      NFS = 0
C
      IF(.NOT.PRCOEF) GO TO 3550
      IF(NS.EQ.2.OR.NSURF.EQ.1) GO TO 3550
      IF(NM.EQ.1) REWIND IAICSC
C
      CALL WRTEMX(IAICSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,1,ID,
     1            A,ITYPE,1,NC,PARM,IRR)
      IF(IRR.NE.0) GO TO 6060
C
 3550 CONTINUE
      IF(NS.EQ.2.AND.NM.EQ.1) REWIND IAICSC
C
C          EVALUATE THE POLYNOMIAL EQUATION FOR DEFLECTIONS.
C            :E PARTIAL DERIVATIVE WITH RESPECT TO X TO GET SLOPES.
C
C
  551 CONTI UE
      IF(NM.NE.1) GO TO 560
      IF(NS.EQ.2) GO TO 556
```

```
C
C         CALCULATE X,Y COORDINATES FOR EVALUATION OF POLYNOMIAL
      X1(1) = XCENTR
      Y1(1) = 0.5*B1BETA
      IF(NSURF.EQ.2) GO TO 552
      MMAX = MAX0(MXBW,MYBW)
      GO TO 554
  552 CONTINUE
      MMAX = MAX0(MXBT,MYBT,MYBW)
  554 CONTINUE
      DO 555 I= 2,MMAX
      X1(I) = X1(I-1)+B1
      Y1(I) = Y1(I-1)+B1BETA
  555 CONTINUE
      GO TO 560
C
  556 CONTINUE
      XADJ = TLAX - WLAX
      DO 557 I=1,MMAX
      X1(I) = X1(I) - XADJ
  557 CONTINUE
      GO TO 580
C
C
  560 CONTINUE
      IF(NS.EQ.2) GO TO 580
      IC = 0
      ILIM = MXBW
      IBEG = 1
      NCH = 0
      GO TO 564
  580 CONTINUE
      IBEG = IFBT
      ILIM = MXBT
      NCH  = MYBW
      IC = 0
      IUP = MXBW
      IF(COPLAN) IUP = IFBT-1
      DO 563 I=1,IUP
      IC = IC + JOC(I)
  563 CONTINUE
  564 CONTINUE
      DO 575 IX=IBEG,ILIM
      I = IX
      IF(NS.EQ.2) I = IX + ICWLAP
      XP(1) =1.
      DO 561 IP=2,MDEG
  561 XP(IP) = XP(IP-1) * X1(IX)
      JI= JS(I)
      JT= JOC(I)+ JI -1
      DO 570 J=JI,JT
      IC = IC +1
      IB = IS(J+NCH)
      IT = IB +  NOC(J+NCH) -1
      IF(IX.LT.IB) GO TO 570
      IF(IX.GT.IT) GO TO 570
      YP(1) =1.
```

```
      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                                        BCSMDA  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                                                          BCSMDA  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
      JSUM = JSUM - JOC(ITROW) + 1                                      MODES   00364
      READ(NT5,9015) (DOS(I),I=IST,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) =DOS(I)                                             MODES   00370
```

```
      JSUM = JSUM + JOC(IX)                                             MODES
  750 CONTINUE                                                          MODES
C                                                                       MODES
C         READ AND STORE SLOPES                                         MODES
      DO 775 J=NC,NCH                                                   MODES
      IST = IS(J)                                                       MODES
      NK = NOC(J) +IST -1                                               MODES
      JSUM = 0                                                          MODES
      ITROW = IST                                                       MODES
      IF(.NOT.COPLAN.AND.NS.EQ.2) ITROW = IST + IOVLAP                  MODES
      DO 770 I=1,ITROW                                                  MODES
  770 JSUM = JSUM + JOC(I)                                              MODES
      JSUM = JSUM - JOC(ITROW) + 1                                      MODES
      READ(NT5,9015) (DOS(I),I=IST,NK)                                  MODES
      DO 775 I=IST,NK                                                   MODES
      IX =I                                                             MODES
      IF(.NOT.COPLAN.AND.NS.EQ.2) IX = I + IOVLAP                       MODES
      ISUB = JSUM + J - JS(IX) - NC + 1                                 MODES
      DEFSL(2,ISUB) = DOS(I)                                            MODES
      JSUM = JSUM + JOC(IX)                                             MODES
  775 CONTINUE                                                          MODES
  900 CONTINUE                                                          MODES
C                                                                       MODES
C         WRITE THE DEFSL ARRAY ONTO MODESC FILE                        MODES
      IF(NS.EQ.NSURF) GO TO 925                                         MODES
      WRITE (IFILE) DEFSL                                               MODES
      GO TO 950                                                         MODES
  925 CONTINUE                                                          MODES
      CALL RDINIT                                                       MODES
      ITYPE = 5HMIXED                                                   MODES
      N = IPNTRM(1,NPNTRS)-1                                            MODES
      CALL WRTEMX(IFILE,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,          MODES
     1            DEFSL,ITYPE,2,N,PARM,IRR)                             MODES
      IF(IRR.NE.0) GO TO 6020                                           MODES
  950 CONTINUE                                                          MODES
      IF(NS.NE.NSURF) GO TO 1500                                        MODES
      IF(.NOT.PRMODS.AND..NOT. PRCOEF) GO TO 1500                       MODES
C                                                                       MODES
C         PRINT MODES, COEFFICIENTS OR BOTH                             MODES
C                                                                       MODES
      WRITE (NT6,9500) TITLE,NM,XMACH                                   MODES
      IF(.NOT.PRCOEF) GO TO 975                                         MODES
      IF(MTYPEW.EQ.3) GO TO 960                                         MODES
C                                                                       MODES
C         PRINT COEFFICIENTS                                            MODES
C                                                                       MODES
      IF(NSURF.EQ.1) GO TO 960                                          MODES
      CALL RDINIT                                                       MODES
      ITYPE = 5HMIXED                                                   MODES
      CALL READMX(IAICSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,   MODES
     1            LRS,XX,M,N,PARM,IRR)                                  MODES
      IF(IRR.NE.0) GO TO 6070                                           MODES
C                                                                       MODES
      IFLG = IPARM(4)                                                   MODES
      IDG1 = IPARM(3)                                                   MODES
      CALL PRECOF(IDG1,XX,IFLG)                                         MODES
C                                                                       MODES
```

```
  960 CONTINUE
      IF(NS.EQ.1.AND.MTYPEW.EQ.3) GO TO 975
      IF(NS.EQ.2.AND.MTYPET.EQ.3) GO TO 975
      CALL PRECOF(IDEG,A,IFLAG)
C
  975 CONTINUE
      NROWS = MXBW
      IF(NSURF.EQ.2) NROWS = MXBT + IOVLAP
C
C        CALL ROUTINE TO PRINT THE MODE SHAPES
      CALL MODOUT(DEFSL,JS,JOC,NROWS,NM,IOVLAP)
C
 1500 CONTINUE
      END FILE IFILE
      M = 1
      N = 400
      CALL WRTEMX(IFILE,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,1,ID,
     1            IS,ITYPE,M,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 6040
C
      END FILE IFILE
      REWIND IFILE
 2000 CONTINUE
      GO TO 3000
 2100 CONTINUE
C
C        DETERMINE THICKNESS SLOPES
C
      NBV = IPNTRM(1,NPNTRS) - 1
      REWIND ITSLSC
      IF(NTSLOP.NE.0) GO TO 2225
C
C        WRITE ARRAY OF ONES
C
      DO 2200 I=1,NBV
      DEFSL(1,I) = 1.0
 2200 CONTINUE
C
      CALL RDINIT
      ITYPE = 5HMIXED
      M = 1
      N = NBV
      CALL WRTEMX(ITSLSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
     1            DEFSL, ITYPE, M, N, PARM, IRR)
      IF(IRR.NE.0) GO TO 6080
C
      END FILE ITSLSC
      REWIND ITSLSC
      GO TO 3000
C
 2225 CONTINUE
      DO 2600 NS=1,NSURF
      IF(NS.EQ.NSURF) GO TO 2230
      IFILE = IVPSC
      REWIND IVPSC
      GO TO 2240
 2230 CONTINUE
```

```
      IFILE = ITSLSC
 2240 CONTINUE
C
      DO 2700 NSL=1,NTSLOP
C
C          ZERO OUT THE ARRAY
      DO 2250 I=1,500
      DEFSL(1,I) = 0.0
 2250 CONTINUE
C
      IF(NS.EQ.2) GO TO 2325
      NCH = MYBW
      NC  = 1
      GO TO 2350
 2325 CONTINUE
      NC  = MYBW + 1
      NCH = MYBW + MYBT
      READ (IVPSC) DEFSL
 2350 CONTINUE
C
      DO 2500 J=NC,NCH
      IST    = IS(J)
      NK     = NOC(J) + IST - 1
      JSUM   = 0
      ITROW = IST
      IF(.NOT.COPLAN.AND.NS.EQ.2) ITROW = IST + IOVLAP
      DO 2400 I=1,ITROW
 2400 JSUM = JSUM + JOC(I)
      JSUM = JSUM - JOC(ITROW) + 1
      READ (NT5,9015) (DOS(I),I=IST,NK)
      DO 2500 I=IST,NK
      IX = I
      IF(.NOT.COPLAN.AND.NS.EQ.2) IX = I + IOVLAP
      ISUB = JSUM + J - JS(IX) - NC + 1
      DEFSL(1,ISUB) = 1. + GAMC * DOS(I)
      JSUM = JSUM + JOC(IX)
 2500 CONTINUE
C
      IF(NS.EQ.NSURF) GO TO 2550
      WRITE (IVPSC) DEFSL
      GO TO 2600
 2550 CONTINUE
      CALL RDINIT
      ITYPE = 5HMIXED
      M = 1
      N = NBV
      CALL WRTEMX(IFILE,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
     1             DEFSL, ITYPE,M,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 8080
C
 2600 CONTINUE
C
 2700 CONTINUE
      END FILE IFILE
      REWIND IFILE
 2800 CONTINUE
 3000 CONTINUE
```

```
C
 9015 FORMAT(7E10.0)
      RETURN
C
C         INPUT DATA ERRORS
C
 6000 WRITE (NT6,9000) IDEG
      GO TO 6199
 6005 WRITE (NT6,9000) IDEG, NPTS
      GO TO 6199
C
C         AN ERROR FROM READING OR WRITING A MATRIX FROM TAPE OR
C         DISK FILE OCCURRED. PRINT MESSAGES AND FLUSH
C
 6010 CONTINUE
      WRITE (NT6,9010) IGEOSC,IRR
      WRITE (NT6,9011) MNAME
      GO TO 6100
 6020 CONTINUE
      WRITE (NT6,9020) MODESC,IRR
      WRITE (NT6,9021) NM
      GO TO 6100
C
 6030 CONTINUE
      WRITE (NT6,9020) MODESC,IRR
      WRITE (NT6,9022)
C
      GO TO 6100
 6040 CONTINUE
      WRITE (NT6,9020) MODESC,IRR
      WRITE (NT6,9023)
      GO TO 6100
C
 6050 CONTINUE
      WRITE (NT6,9050) IGEOSC,IRR
      WRITE (NT6,9051) NM
      GO TO 6100
C
 6060 WRITE (NT6,9050) IAICSC,IRR
      WRITE (NT6,9051) NM
      GO TO 6100
C
 6070 CONTINUE
      WRITE (NT6,9070) IAICSC,IRR
      WRITE (NT6,9071) NM
      GO TO 6100
C
 6080 CONTINUE
      WRITE (NT6,9080) ITSLSC,IRR
      WRITE (NT6,9081) NSL
C
C
 6100 CONTINUE
      WRITE(NT6,9101) ID(1),ID(2)
      WRITE(NT6,9102) PARM,IPARM
      WRITE(NT6,9103) NFS,NMS
      WRITE(NT6,9104) ITYPE,M,N
```

```
 6199 CONTINUE
      WRITE(NT6,9900)
C
      CALL FLUSH(1)
C
 7005 FORMAT(*0 PREVIOUS MODES AND GEOMETRY HAVE BEEN SPECIFIED. * )
 7010 FORMAT(86H0*** WARNING - PREVIOUS MODE SHAPES HAVE BEEN SPECIFIED,
     1 BUT GEOMETRY HAS CHANGED. *** )
 7015 FORMAT(80H0*** ERROR - PREVIOUS MODE SHAPES HAVE BEEN SPECIFIED, B
     1UT THE GEOM HAS CHANGED. / 13X,41HPREVIOUS MODE SHAPES WERE AT BOX
     2 CENTERS., 26X,4H *** )
 9000 FORMAT(43H0*** ERROR - SPECIFIED POLYNOMIAL DEGREE OF  I5,
     1   22H IS OUTSIDE LIMITS, OR, I6,24H IS TOO MANY POINTS *** )
 9010 FORMAT(53H0*** ERROR - WHILE READING THE GEOMETRY SCRATCH FILE A10
     1, 15H, ERROR CODE = I4,4H *** )
 9011 FORMAT(5X,32HAN ATTEMPT WAS MADE TO READ THE A6, 8H MATRIX.//)
 9020 FORMAT(52H0*** ERROR - WHILE WRITING ON THE MODE SCRATCH FILE A10,
     1  15H, ERROR CODE = I4,4H *** )
 9021 FORMAT(5X,40HAN ATTEMPT WAS MADE TO WRITE MODE SHAPE I3,//)
 9022 FORMAT(5X,47HAN ATTEMPT WAS MADE TO WRITE THE POINTER ARRAY. // )
 9023 FORMAT(5X,41HAN ATTEMPT WAS MADE TO WRITE INDEX ARRAY.  //)
 9050 FORMAT(57H0*** ERROR - WHILE WRITING THE COEFFICIENT ARRAY ON FILE
     1       A10,15H, ERROR CODE = I4,4H *** )
 9051 FORMAT(5X,44HAN ATTEMPT WAS MADE TO WRITE FOR MODE SHAPE I4 )
 9070 FORMAT(59H0*** ERROR - WHILE READING THE COEFFICIENT ARRAY FROM FI
     1LE     A10,15H, ERROR CODE = I4,4H *** )
 9071 FORMAT(5X,43HAN ATTEMPT WAS MADE TO READ FOR MODE SHAPE I4  )
 9080 FORMAT(56H0*** ERROR - WHILE WRITING ON THE THICKNESS SLOPE FILE
     1 2X,A10,15H, ERROR CODE = ,I4,4H *** )
 9081 FORMAT(5X,45HAN ATTEMPT WAS MADE TO WRITE THICKNESS SLOPE I4,//)
 9500 FORMAT(1H1,8A10,// 46X,*  MODE SHAPE NUMBER *,I3,
     1                / 46X,*MACH NUMBER =*,F11.6,/46X,24(1H-),/ )
 9101 FORMAT(5X,*MATRIX ID = *, A10, I10)
 9102 FORMAT(5X,*PARAMETERS *,10E11.3, /10X,*(INTEGER)*, I7, 9I11 )
 9103 FORMAT(5X,*FILE SPACING = *,I3,* MATRIX SPACING = *,I3)
 9104 FORMAT(5X,*MATRIX TYPE -*,A10,*, DIMENSIONED (*I4,2H X,I4,1H) )
 9900 FORMAT(*0    ERROR OCCURRED IN MODES SECTION (MAIN PROGRAM).* )
      END
```

```
      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,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
      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 /INDEX/ IS(100),NOC(100),JS(100),JOC(100)                  TAPMOD 00020
C                                                                       TAPMOD 00021
      LOGICAL MXREAD,RANDIN                                             TAPMOD 00022
      MXREAD = .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 = 5HMIXED                                                   TAPMOD 00032
      CALL READMX(INTAPE,MXREAD,RANDIN,NFS,NMS,LS,NMR,1, NID,ID,ITYPE,  TAPMOD 00033
     1            LRS, D, M,N,PARM,IRR)                                 TAPMOD 00034
      IF(IRR.NE.0) GO TO 6020                                           TAPMOD 00035
      CALL RDINIT                                                       TAPMOD 00036
      ITYPE = 5HMIXED                                                   TAPMOD 00037
      CALL READMX(INTAPE,MXREAD,RANDIN,NFS,NMS,LS,NMR,1, NID,ID,ITYPE,  TAPMOD 00038
     1            LRS, S, M,N,PARM,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         FIRST 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
      NC = MYBW + 1
      NCH = MYBW + MYBT
C
C         STORE DEFLECTIONS AND SLOPES
  725 CONTINUE
      ITS = 0
      DO 750 J=NC,NCH
      IST = IS(J)
      NK = NOC(J) + IST - 1
      JSUM = 0
      DO 730 I=1,IST
  730 JSUM = JSUM + JOC(I)
      JSUM = JSUM - JOC(IST) + 1
      DO 750 I=IST,NK
      ITS = ITS + 1
      ISUB = JSUM + J - NC + 1 - JS(I)
      DEFSL(1,ISUB) = D(ITS)
      DEFSL(2,ISUB) = S(ITS)
      JSUM= JSUM + JOC(I)
  750 CONTINUE
C
      RETURN
C
C         AN ERROR DURING READING A MATRIX FROM TAPE OR
C         DISK FILE OCCURRED. PRINT MESSAGES AND FLUSH
C
 6010 CONTINUE
 6020 CONTINUE
      WRITE (NT6,9020) MODESC,IRR
      WRITE (NT6,9021) NM
 6100 CONTINUE
      WRITE(NT6,9101) ID(1),ID(2)
      WRITE(NT6,9102) PARM,IPARM
      WRITE(NT6,9103) NFS,NMS
      WRITE(NT6,9104) ITYPE,M,N
      WRITE(NT6,9900)
C
      CALL FLUSH(1)
C
 9020 FORMAT(54H0*** ERROR - WHILE READING FROM THE MODE SCRATCH FILE
     1 A10,15H, ERROR CODE = I4,4H *** )
 9021 FORMAT(5X,39HAN ATTEMPT WAS MADE TO READ MODE SHAPE I3,//)
 9101 FORMAT(5X,*MATRIX ID = *, A10, I10)
 9102 FORMAT(5X,*PARAMETERS *,10E11.3, /10X,*(INTEGER)*, I7, 9I11 )
 9103 FORMAT(5X,*FILE SPACING = *,I3,* MATRIX SPACING = *,I3)
 9104 FORMAT(5X,*MATRIX TYPE -*,A10,*, DIMENSIONED (*I4,2H X,I4,1H) )
 9900 FORMAT(*0    ERROR OCCURRED IN MODES SECTION (SUBROUTINE TAPMOD).*
     1    )
      END
```

```
      SUBROUTINE ROPER                                                  ROPER   00002
C                                                                       ROPER   00003
C        SUBROUTINE TO DETERMINE THE BOXES ON EACH ROW THAT SHOULD      ROPER   00004
C        HAVE MODE 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
      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 /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                                          ROPER   00054
      NCH = MYBW + MYBT                                                 ROPER   00055
      JCH = MYBW + 1                                                    ROPER   00056
      IFBT = (IXBT-IXBW)/NSUBDV +1                                      ROPER   00057
      IOVLAP = 0                                                        ROPER   00058
      IF(IFBT.LE.MXBW)  IOVLAP = MXBW - IFBT + 1                        ROPER   00059
      DO 1000 I=IFBT,MXBT                                               ROPER   00060
      IX = I + IOVLAP                                                   ROPER   00061
      JS(IX) = 0                                                        ROPER   00062
      JOC(IX)= 0                                                        ROPER   00063
      DO 800 J=JCH,NCH                                                  ROPER   00064
      IF(I.LT.IS(J)) GO TO 800                                          ROPER   00065
      ILAST = IS(J) + NOC(J) -1                                         ROPER   00066
      IF(I.GT.ILAST) GO TO 800                                          ROPER   00067
      IF(JS(IX).EQ.0) JS(IX) = J-MYBW                                   ROPER   00068
      JV = J- MYBW                                                      ROPER   00069
      JOC(IX) = JV - JS(IX) + 1                                         ROPER   00070
  800 CONTINUE                                                          ROPER   00071
  825 CONTINUE                                                          ROPER   00072
 1000 CONTINUE                                                          ROPER   00073
 1500 CONTINUE                                                          ROPER   00074
      RETURN                                                            ROPER   00075
      END                                                               ROPER   00076
```

```
      SUBROUTINE FITTER(M,N,X,Y,Z,C,CN,IDIM)
      DIMENSION X(100), Y(100), Z(IDIM,100), C(IDIM,66)
      DIMENSION AI(68),A(66,68),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 COEFFIENTS
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.
      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 + 1
      XM2= XM/2.
      NC = XM*XM2 + XM2 + EPS
      IF(NC.LE.N) GO TO 25
      M = M-1
      GO TO 15
   25 CONTINUE
C
      NAC = NC
C
C        DETERMINE THE MAXIMUM DEGREE THAT CAN BE COMPUTED IN
C        EACH DIRECTION AND SET UP ORDER OF SOLUTION.
C
      NDV = 1
      NDX = 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(NDV-1.EQ.M) GO TO 65
   55 CONTINUE
   60 CONTINUE
      MDX = NDV -1
   65 CONTINUE
C
      NDV = 1
      MDY = M
      VS(1) = Y(1)
      DO 80 I=1,N
      DO 70 J=1,NDV
      IF(Y(I).EQ.VS(J)) GO TO 75
   70 CONTINUE
      NDV = NDV + 1
      VS(NDV) = Y(I)
      IF(NDV-1.EQ.M) GO TO 85
   75 CONTINUE
   80 CONTINUE
      MDY = NDV - 1
   85 CONTINUE
C
      ITOT =NC +1
      ITOT1 = ITOT
      IF(COMPLX) ITOT = ITOT + 1
C
C         ZERO OUT THE A ARRAY
C
      DO 95 I=1,NC
      C(1,I) = 0.0
      IF(.NOT.COMPLX) GO TO 90
      C(2,I) = 0.0
   90 CONTINUE
      DO 95 J=1,ITOT
   95 A(I,J) = 0.0
C
C         DETERMINE DEVIATION EQUATION AND SQUARE THE EQUATION
C
      AI(1) =1.0
      XP(1) =1.0
      YP(1) =1.0
      MM = M + 1
      DO 200 K=1,N
      DO 10 L=2,MM
      XP(L) = XP(L-1)*X(K)
      YP(L) = YP(L-1)*Y(K)
   10 CONTINUE
C
      I = 1
      DO 40 L=2,MM
      DO 20 LL=1,L
      IL= L - LL +1
      IF(LL-1.GT.MDY) GO TO 30
      IF(IL-1.GT.MDX) GO TO 20
      I = I + 1
      AI(I) = XP(IL)*YP(LL)
   20 CONTINUE
   30 CONTINUE
```

```
   40 CONTINUE
      AI(I+1) = Z(1,K)
      IF(COMPLX) AI(I+2) = Z(2,K)
      IF(K.GT.1) GO TO 45
      NC = I
      ITOT = NC + 1
      ITOT1 = ITOT
      IF(COMPLX) ITOT = ITOT + 1
   45 CONTINUE
C
      DO 1100 I=1,NC
      DO 1100 J=I,ITOT
      ASAV = AI(I)*AI(J)
      A(I,J)=A(I,J)+ASAV
 1100 CONTINUE
  200 CONTINUE
C
C         SQUARE ROOT METHOD
C         INTERMEDIATE MATRIX
      DO 1200 I=1,NC
      IM1 = I-1
      TMP=0.0
      IF(I.EQ.1) GO TO 1150
      DO 1120 L=1,IM1
 1120 TMP= TMP+ A(L,I)**2
 1150 CONTINUE
      T = A(I,I) - TMP
      IF(T.GT.EPS) GO TO 4
      A(I,I) = 0.0
      GO TO 1200
    4 CONTINUE
      A(I,I) = SQRT(T)
      IF(A(I,I).GT.EPS) GO TO 1155
      A(I,ITOT) = 0.0
      GO TO 1200
 1155 CONTINUE
C
      JS = I+1
      DO 1180 J = JS,ITOT
      TMP= 0.0
      IF(I.EQ.1) GO TO 1175
      DO 1160 L=1,IM1
 1160 TMP = TMP + A(L,I)*A(L,J)
 1175 A(I,J) =(A(I,J)-TMP)/A(I,I)
 1180 CONTINUE
 1200 CONTINUE
C
C
C         BACK SUBSTITUTE FOR COEFFICIENTS
      DO 1400 K=1,NC
      I = NC - K + 1
      IP1=I+1
      TMP1  = 0.0
      TMP2  = 0.0
      IF(A(I,I).GT.EPS) GO TO 1325
      C(1,I) = 0.0
      IF(COMPLX) C(2,I) = 0.0
```

```
      GO TO 1400                                                        FITTER  00173
 1325 CONTINUE                                                          FITTER  00174
      IF(I.EQ.NC)GO TO 1375                                             FITTER  00175
      DO 1350 L=IP1,NC                                                  FITTER  00176
      TMP1  = TMP1 + A(I,L)*C(1,L)                                      FITTER  00177
      IF(.NOT.COMPLX) GO TO 1350                                        FITTER  00178
      TMP2  = TMP2 + A(I,L)*C(2,L)                                      FITTER  00179
 1350 CONTINUE                                                          FITTER  00180
 1375 CONTINUE                                                          FITTER  00181
      C(1,I) =(A(I,ITOT1)-TMP1)/A(I,I)                                  FITTER  00182
      IF(.NOT.COMPLX) GO TO 1400                                        FITTER  00183
      C(2,I) =(A(I,ITOT) -TMP2)/A(I,I)                                  FITTER  00184
 1400 CONTINUE                                                          FITTER  00185
C                                                                       FITTER  00186
C                                                                       FITTER  00187
C         REORDER THE COEFFICIENTS IN CORRECT POWERS                    FITTER  00188
C         OF X AND Y.                                                   FITTER  00189
C                                                                       FITTER  00190
      IF(NAC.EQ.NC) GO TO 1475                                          FITTER  00191
C                                                                       FITTER  00192
      IZ = 1                                                            FITTER  00193
      I  = 1                                                            FITTER  00194
      DO 1440 L=2,MM                                                    FITTER  00195
      DO 1420 LL=1,L                                                    FITTER  00196
      IL = L -LL +1                                                     FITTER  00197
      I = I +1                                                          FITTER  00198
      IF(LL-1.LE.MDY.AND.IL-1.LE.MDX) GO TO 1410                        FITTER  00199
      X(I) = 0.0                                                        FITTER  00200
      Y(I) = 0.0                                                        FITTER  00201
      GO TO 1420                                                        FITTER  00202
 1410 CONTINUE                                                          FITTER  00203
      IZ = IZ + 1                                                       FITTER  00204
      X(I) = C(1,IZ)                                                    FITTER  00205
      IF(COMPLX) Y(I) = C(2,IZ)                                         FITTER  00206
 1420 CONTINUE                                                          FITTER  00207
 1440 CONTINUE                                                          FITTER  00208
C                                                                       FITTER  00209
      DO 1450 I=2,NAC                                                   FITTER  00210
      C(1,I) = X(I)                                                     FITTER  00211
      IF(COMPLX) C(2,I) = Y(I)                                          FITTER  00212
 1450 CONTINUE                                                          FITTER  00213
 1475 CONTINUE                                                          FITTER  00214
C                                                                       FITTER  00215
C         ELIMINATE THE SCALE FACTOR FROM THE COEFFICIENTS.             FITTER  00216
C                                                                       FITTER  00217
      IF(CN.EQ.1.0) GO TO 1700                                          FITTER  00218
      I=1                                                               FITTER  00219
      CP= 1.0/CN                                                        FITTER  00220
      DO 1600 L1=2,MM                                                   FITTER  00221
      DO 1500 L2=1,L1                                                   FITTER  00222
      I = I+1                                                           FITTER  00223
      C(1,I) = C(1,I)*CP                                                FITTER  00224
      C(2,I) = C(2,I)*CP                                                FITTER  00225
 1500 CONTINUE                                                          FITTER  00226
      CP= CP/CN                                                         FITTER  00227
 1600 CONTINUE                                                          FITTER  00228
 1700 CONTINUE                                                          FITTER  00229
```

```
C                                                                 FITTER  00230
C         THE C ARRAY NOW CONTAINS THE COEFFICIENTS.              FITTER  00231
C                                                                 FITTER  00232
      RETURN                                                      FITTER  00233
      END                                                         FITTER  00234
```

```
      SUBROUTINE MODOUT(DEFSL,JS,JOC,NROWS,NM,IOVLAP)                   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,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,NOUTP,         FILES   00002
     1                IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC          FILES   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
      DIMENSION SLOMAT(3)                                               MODOUT  00008
      EQUIVALENCE  (SLOMAT(2), IMAT)                                    FTNXX   00046
      DATA SLOMAT/10H(I3,3X,    ,10H          0,10HP,15F7.3)  /         MODOUT  00009
      DATA IMAT1,IMAT2 /10H          0, 10H         -0 /                MODOUT  00011
C                                                                       MODOUT  00012
C        MODOUT WILL PRINT THE MODE SHAPES OUT IN ROW/COLUMN FORM       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        IOVLAP - NUMBER OF BOXES OVERLAP BETWEEN PLANFORMS FOR         MODOUT  00021
C                 NON-COPLANAR SURFACES                                 MODOUT  00022
C                                                                       MODOUT  00023
      IMAT = 10H         0                                              MODOUT  00024
      NSETS = 0                                                         MODOUT  00025
      DO 50 I=1,NROWS                                                   MODOUT  00026
      JL = JS(I)+JOC(I)-1                                               MODOUT  00027
      IF(NSETS.LT.JL) NSETS=JL                                          MODOUT  00028
   50 CONTINUE                                                          MODOUT  00029
      JSETS = (NSETS-1)/15 + 1                                          MODOUT  00030
      DO 1000 NP=1,2                                                    MODOUT  00031
C                                                                       MODOUT  00032
C        FIND LARGEST VALUE                                             MODOUT  00033
C                                                                       MODOUT  00034
      VALUE = 0.0                                                       MODOUT  00035
      DO 100 L=1,500                                                    MODOUT  00036
      AVAL = ABS(DEFSL(NP,L))                                           MODOUT  00037
      IF(AVAL.GT.VALUE) VALUE = AVAL                                    MODOUT  00038
  100 CONTINUE                                                          MODOUT  00039
      POW = 1                                                           MODOUT  00040
      TSCALE = 10.                                                      MODOUT  00041
      IF(VALUE.GE.10.) TSCALE = 0.1                                     MODOUT  00042
      DO 110 N=1,8                                                      MODOUT  00043
      POW = POW *TSCALE                                                 MODOUT  00044
      TTEN = VALUE * POW                                                MODOUT  00045
      IF(TTEN.GE.10.) GO TO 110                                         MODOUT  00046
      IF(TTEN.LT.1. ) GO TO 110                                         MODOUT  00047
      NPA = N                                                           MODOUT  00048
      GO TO 115                                                         MODOUT  00049
  110 CONTINUE                                                          MODOUT  00050
      NPA= 0                                                            MODOUT  00051
```

```
  115 CONTINUE
      IF(VALUE.LE.1.00) GO TO 120
C
C           THE ARRAY MUST BE SCALED DOWN.
C
      NPX = -NPA
      IMAT = IMAT2
      GO TO 122
C
C           THE ARRAY MUST BE SCALED UP.
C
  120 CONTINUE
      NPX = NPA
      IMAT = IMAT1
  122 CONTINUE
      IMAT = IMAT + NPA
      IF(NP.EQ.2) GO TO 124
      WRITE (NT6,9005) NPX
      GO TO 125
  124 CONTINUE
      WRITE (NT6,9010) NPX
  125 CONTINUE
      DO 900 JPS=1,JSETS
      JBASE = (JPS-1)*15
      DO 150 JC=1,15
  150 JPH(JC) = JC + JBASE
      JL=15
      IF(JPS.EQ.JSETS) JL= NSETS - 15*(JSETS-1)
      WRITE(NT6,9015) (JPH(J) ,J=1,JL)
      IOV = 0
      ITOT = 0
      DO 800 I=1,NROWS
C
C           ZERO OUT PRINT ARRAY
      DO 200 J=1,50
  200 DS(J) = 0.0
C
C           PUT THE VALUES INTO PRINT ARRAY
      JI = JS(I)
      IF(JI.EQ.0) GO TO 800
      JL = JS(I) +JOC(I) -1
      DO 300 J=JI,JL
      ITOT = ITOT+1
  300 DS(J) = DEFSL(NP,ITOT)
C
C           PRINT THE ONES IN THIS SET
      JIP=(JPS-1)*15 +1
      JIL= JIP +14
      IF(JL.LT.JIP) GO TO 800
      IF(JI.GT.JIL) GO TO 800
      IF(JIL.GT.JL) JIL=JL
      M = I
      IF(IOVLAP.EQ.0) GO TO 350
      IF(I.LE.MXBW)   GO TO 350
      M = I-IOVLAP
      IF(IOV.NE.0) GO TO 350
      IOV = 1
```

```
      WRITE(NT6,9015)
  350 CONTINUE
      WRITE(NT6,SLOMAT) M,(DS(J),J=IIP,JIL)
  800 CONTINUE
  900 CONTINUE
 1000 CONTINUE
      RETURN
 9005 FORMAT(1H0,46X,*DEFLECTIONS X 1.0E *,I2,/ 47X,21(1H-) )
 9010 FORMAT(1H0,//47X,*   SLOPES X 1.0E *,I2,/ 47X,21(1H-) )
 9015 FORMAT(1H0,5X,15(I4,3X))
 9020 FORMAT(I3,3X,15F7.3)
      END
```

```
      SUBROUTINE PRECOF(IDEG,A,IFR)                                     PRECOF  00002
      COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUTP,         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        IDEG - 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(IDEG.EQ.0) GO TO 550                                           PRECOF  00021
      IDEX = 2                                                          PRECOF  00022
      DO 520 I=1,IDEG                                                   PRECOF  00023
      NCL = I+1                                                         PRECOF  00024
      LDEX = IDEX + I                                                   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(1H0,6(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(1H0,12X,*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 = 5HMIXED                                                   VICMAIN 00095
      MXARRY = 6HATAB                                                   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(NV.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
      NBIZ = 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
      NBIZ = 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.NBIZ) 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
      NBIZ = 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
C         CALCULATE 2 PLANAR AICS IF SUBDIVISION IS APPLIED.
      NPK = 1
      IF(NSUBDV.GT.1) NPK = 2
      NVCS = NSPATK + NPK
      DO 1000 NV = 1,NVCS
C
C         SET K1 VALUE IF SUBDIVIDED AIC
C
      IF(NPK.EQ.2.AND.NV.EQ.NVCS) K1= K1/FLOAT(NSUBDV)
C
      LENZ = LKERNL
      IF(NV.EQ.NVCS.AND.NPK.EQ.2) LENZ = LSKERN
      DO 100 I=1,LENZ
      C(I) = (0.,0.)
      W(I) = (0.,0.)
      V(I) = (0.,0.)
  100 CONTINUE
C
  110 CONTINUE
      IF(NV.GT.NVCS-NPK) GO TO 10
C
C         READ MUAIC ARRAY FROM IGEOSC. FIRST FILE MUST BE SKIPPED
C         PRIOR TO FIRST READ.
C
      CALL RDINIT
      IF(NV.NE.1) GO TO 200
      REWIND IGEOSC
      NFS = 1
  200 CONTINUE
      ITYPE = 5HMIXED
      MXARRY = 6HMUAIC
      CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
     1            LRS,MUAIC,M,NROWS,PARM,IRR)
      IF(IRR.NE.0) GO TO 6010
C
      YBAR = PARM(4)
      EL = PARM(5)
      NN = 1
C
C         DETERMINE IF SPATIAL AICS ARE ON TAPE AND GET THEM
C         IF POSSIBLE.
C
      NKST = 0
      IF(.NOT.OSPAIC) GO TO 25
C
C         AICS ARE ON TAPE. GET TABLE OF CONTENTS.
      REWIND NSPAIC
      CALL RDINIT
      NFS = 1
      ITYPE = 5HMIXED
      MXARRY = 6HTAB
      CALL READMX(NSPAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,500,NID,ID,ITYPE,
     1            LRS,  TAB,NKST,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 6060
C
      CALL RDINIT
```

```
C
C        CALCULATE 2 PLANAR AICS IF SUBDIVISION IS APPLIED.
      NPK = 1
      IF(NSUBDV.GT.1) NPK = 2
      NVCS = NSPATK + NPK
      DO 1000 NV = 1,NVCS
C
C        SET K1 VALUE IF SUBDIVIDED AIC
C
      IF(NPK.EQ.2.AND.NV.EQ.NVCS) K1= K1/FLOAT(NSUBDV)
C
      LENZ = LKERNL
      IF(NV.EQ.NVCS.AND.NPK.EQ.2) LENZ = LSKERN
      DO 100 I=1,LENZ
      C(I) = (0.,0.)
      W(I) = (0.,0.)
      V(I) = (0.,0.)
  100 CONTINUE
C
  110 CONTINUE
      IF(NV.GT.NVCS-NPK) GO TO 10
C
C        READ MUAIC ARRAY FROM IGEOSC. FIRST FILE MUST BE SKIPPED
C        PRIOR TO FIRST READ.
C
      CALL RDINIT
      IF(NV.NE.1) GO TO 200
      REWIND IGEOSC
      NFS = 1
  200 CONTINUE
      ITYPE = 5HMIXED
      MXARRY = 6HMUAIC
      CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
     1            LRS,MUAIC,M,NROWS,PARM,IRR)
      IF(IRR.NE.0) GO TO 6010
C
      YBAR = PARM(4)
      EL = PARM(5)
      NN = 1
C
C        DETERMINE IF SPATIAL AICS ARE ON TAPE AND GET THEM
C        IF POSSIBLE.
C
      NKST = 0
      IF(.NOT.OSPAIC) GO TO 25
C
C        AICS ARE ON TAPE. GET TABLE OF CONTENTS.
      REWIND NSPAIC
      CALL RDINIT
      NFS = 1
      ITYPE = 5HMIXED
      MXARRY = 6HTAB
      CALL READMX(NSPAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,500,NID,ID,ITYPE,
     1            LRS,  TAB,NKST,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 6060
C
      CALL RDINIT
```

```
C
C        THERE IS A MATRIX THAT CAN BE ENLARGED.
      AMACH(IVAL) = -AMACH(IVAL)
      WRITE (NT6,9030) IVAL, AERR(IVAL),ISIZE(IVAL),NROWS
C
C        SPACE TO CORRECT ARRAY ON TAPE
      CALL RDINIT
      NMS = (IVAL-1)*4
      ITYPE = 5HMIXED
      MXARRY = 6HMUTWO
      CALL READMX(NSFAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR, 2 ,NID,ID,ITYPE,
     1             LRS,MUTWO,M,N,FARM,IRR)
      IF(IRR.NE.0) GO TO 6060
C
      CALL RDINIT
      ITYPE = 5HMIXED
      MXARRY = 6H C
      CALL READMX(NSFAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
     1             LRS, C, M,N,FARM,IRR)
      IF(IRR.NE.0) GO TO 6060
C
      CALL RDINIT
      ITYPE = 5HMIXED
      MXARRY = 6H W
      CALL READMX(NSFAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
     1             LRS, W, M,N,FARM,IRR)
      IF(IRR.NE.0) GO TO 6060
C
      CALL RDINIT
      ITYPE = 5HMIXED
      MXARRY = 6H V
      CALL READMX(NSFAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
     1             LRS, V, M,N,FARM,IRR)
      IF(IRR.NE.0) GO TO 6060
C
      GO TO 25
  800 CONTINUE
C
C        THERE IS A GOOD MATRIX ON TAPE. READ THE TAPE, PRINT MESSAGE,
C        MAIL RESUME ON THIS ONE ANYWAY.
C
C        SET NSIZ  EQUAL TO NROWS SO THAT MATRIX WILL NOT BE WRITTEN
C        ON TAPE
C
      NSIZ = NROWS
      CALL RDINIT
      NMS = (I-1)*4
      ITYPE = 5HMIXED
      MXARRY = 6HMUTWO
      CALL READMX(NSFAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2, NID,ID,ITYPE,
     1             LRS,MUTWO,M,N,FARM,IRR)
      IF(IRR.NE.0) GO TO 6060
C
      CALL RDINIT
      ITYPE = 5HMIXED
      MXARRY = 6H C
      CALL READMX(NSFAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
```

```
     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 = 5HMIXED                                                   VICMAIN 00213
      MXARRY = 6H 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 = 5HMIXED                                                   VICMAIN 00220
      MXARRY = 6H 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         LTOBKN = 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 CARDC USE THAT             VICMAIN 00254
C         FIND IF THE PLANFORM LIMITS THE SIZE OF THE EFFECTIVE AREA.   VICMAIN 00255
C                                                                       VICMAIN 00256
      IF(NROWEA.GT.20) NROWEA = 20                                      VICMAIN 00257
      IF(NROWEA.NE.0) MXSKRN = NSUBDV * NROWEA                          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 325                                    VICMAIN 00265
```

```
      NBOXES = NBOXES + NBOXRW
      NBOXRW = NBOXRW - 2
      IF(NBOXRW.LE.0) GO TO 330
      NSROW  = NSROW  + 2
      GO TO 300
  325 CONTINUE
      NSROW = NSROW - 1
      IF (NBOXES-NBOXRW/2 .GT. LSDW) NSROW = NSROW - 1
      GO TO 335
  330 CONTINUE
      NSROW = MXSKRN
  335 CONTINUE
      XNA = MXSKRN
      LSKERN = XNA*(XNA/2.) + (XNA/2.) + 0.001
      ISUB = MXSKRN/NSUBDV
      IF (NSROW .LT. MXSKRN) ISUB =  NSROW/NSUBDV
      SUB    = ISUB
      LTOBKN = SUB*(SUB/2.) + (SUB/2.) + 0.001
      IST    = LSKERN - LTOBKN
  340 CONTINUE
      IPKERN = IST + 1
      MAXL = IST + LUKERN
      IF(MAXL.LE.LKERNL) GO TO 21
      WRITE (NT6,9305) MAXL,LKERNL
 9305 FORMAT(59HO*** ERROR - THE SIZE OF THE AIC ARRAY FOR THIS PLANFORM
     1 IS,I5,29H, THE MAXIMUM SIZE ALLOWED IS,I5, 5H. *** )
      CALL FLUSH(1)
C
   21 CONTINUE
C
      IF(NPK.EQ.2.AND.NV.EQ.NVCS-1) GO TO 22
      NN = 1
      NROWS = MXSKRN
      GO TO 23
   22 CONTINUE
      NN = IPKERN
      NROWS = NPLKRN
   23 CONTINUE
      DO 24 I=1,NROWS
      MUAIC(1,I) = I
      MUAIC(2,I) = I + I - 1
   24 CONTINUE
C
C
C         DETERMINE IF KERNELS EXIST ON TAPE AND GET THEM FROM TAPE
C         IF POSSIBLE.
C
      NKOT = 0
      IF(.NOT.OPLAIC) GO TO 25
C
C         KERNELS ARE ON TAPE. GET TABLE OF CONTENTS
      REWIND NPLAIC
      CALL RDINIT
      NFS = 1
      ITYPE = 5HMIXED
      MXARRY = 6HTAB
      CALL READMX(NPLAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,500,NID,ID,ITYPE,
```

```
    2  31X,1HC,34X,1HW,34X,1HV,/3X,2HNU,3X,2HMU,7X,*VELOCITY POTENTIAL
    3COEFFICIENT* 10X,*UPWASH COEFFICIENT*,15X,*SIDEWASH COEFFICIENT* /
    4 2(3X,2H--),7X, 32(1H-),2X,32(1H-),2X,32(1H-) //)
C
      IF(.NOT.PRNT) GO TO 52
      K=0
      KN = 0
      IF(NPK.EQ.2.AND.NV.EQ.NVCS-1) KN = IPKERN -1
C
      DO 50 I=1,NROWS
      M= I-1
      MH = M/2
      JS = 2*(I-1) +1
      IF(YBAR.NE.0.0) JS = JS+1
      IF(EL.EQ.0.0) JS = I
      DO 50 J=1,JS
      K=K+1
      KN = KN + 1
      N = I - J
      IF(EL.EQ.0.0) N = 1- J
      IF(YBAR.LT.0.0) N = -N
      WRITE (NT6,9210) M,N,C(KN),W(K),V(K)
 9210 FORMAT(2I5,5X,6E17.8)
   50 CONTINUE
      GO TO 53
C
   52 CONTINUE
      RN = NROWS
      K = RN* (RN/2.) + (RN/2.) + 1.0E-05
      IF(EL.EQ.0.) GO TO 53
      K = K+K
      IF(YBAR.EQ.0.)  K = K - NROWS
   53 CONTINUE
C
C
      IF (NSPATK .EQ. 0 .OR. NV .GT. NVCS-NPK) GO TO 55
      IF(NV.EQ.1) REWIND IAICSC
C
C         WRITE THE SPATIAL AICS ON A SCRATCH FILE
C
      CALL RDINIT
      PARM(1) = K1
      PARM(2) = XMACH
      PARM(4) = YBAR
      PARM(5) = EL
      ITYPE = 5HMIXED
      MXARRY = 6HMUAIC
      CALL WRTEMX(IAICSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
     1             MUAIC,ITYPE,2,NROWS,PARM,IRR)
      IF(IRR.NE.0) GO TO 6170
C
      MXARRY = 6H C
      CALL WRTEMX(IAICSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
     1             C,     ITYPE, 2,K,     PARM, IRR)
      IF(IRR.NE.0) GO TO 6170
C
      MXARRY = 3H W
```

```
      AMACH(IVAL) = -AMACH(IVAL)
      WRITE (NT6,9030) IVAL, AERR(IVAL),ISIZE(IVAL),NPLKRN
 9030 FORMAT(46H0THE PROGRAM IS GOING TO ENLARGE AIC ARRAY NO. I3,
     1   29H, GENERATED AT AN ACCURACY OF F8.5 /19H IT IS NECESSARY TO
     2   29H ENLARGE THE SAVED ARRAY FROM I3, 3H TO, I3 )
C
C        SPACE TO CORRECT ARRAY ON TAPE
      REWIND NPLAIC
      CALL RDINIT
      NMS = IVAL-1
      ITYPE = 5HMIXED
      MXARRY = 6H C
      CALL READMX(NPLAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
     1            LRS,C(NN), M, N, PARM, IRR)
      IF(IRR.NE.0) GO TO 6050
C
C        SET MUAIC ARRAY FOR EXPANSION AIC CALCULATION
      DO 1700 I=1,NSIZE
      MUAIC(1,I) = 0
      MUAIC(2,I) = 0
 1700 CONTINUE
      GO TO 25
 1800 CONTINUE
C
C        THERE IS A GOOD MATRIX ON TAPE. READ THE TAPE, PRINT MESSAGE,
C        DO NOT MAIL RESUME.
C        SET NSIZE EQUAL TO NPLKRN SO THAT MATRIX WILL NOT BE WRITTEN
C        ON TAPE.
C
      NSIZE = NPLKRN
      CALL RDINIT
      REWIND NPLAIC
      NMS = I-1
      ITYPE = 5HMIXED
      MXARRY = 6H C
      CALL READMX(NPLAIC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,  NID,ID,ITYPE,
     1            LRS,C(NN), M, N, PARM, IRR)
      IF(IRR.NE.0) GO TO 6050
C
      WRITE (NT6,6005) I, AERR(I)
      GO TO 35
   25 CONTINUE
C
      CALL KERNEL(XMACH,K1,ERR,C(NN),W,V)
C
   35 CONTINUE
      PRNT = .FALSE.
      IF (NV.GT.NVCS-NPK) GO TO 40
      IF(PRSAIC) PRNT = .TRUE.
      GO TO 45
   40 CONTINUE
      IF(PRPAIC) PRNT = .TRUE.
   45 CONTINUE
      IF (PRNT) WRITE (NT6,9005) TITLE,XMACH,K1,ERR,EL,YBAR
 9005 FORMAT(1H1, 8A10, // 40X, *AIC CALCULATIONS*, ///
     X       17X, *MACH =*, F10.5, 5X,  *K1 =*,
     1  F10.7,5X,*ERR =*,E12.5,5X,*EL =*, F6.2,5X,*YBAR =*,F6.2,//
```

```
     2  31X,1HC,34X,1HW,34X,1HV,/3X,2HNU,3X,2HMU,7X,*VELOCITY POTENTIAL
     3COEFFICIENT* 10X,*UPWASH COEFFICIENT*,15X,*SIDEWASH COEFFICIENT* /
     4 2(3X,2H--),7X, 32(1H-),2X,32(1H-),2X,32(1H-) //)
C
      IF(.NOT.PRNT) GO TO 52
      K=0
      KN = 0
      IF(NPK.EQ.2.AND.NV.EQ.NVCS-1) KN = IPKERN -1
C
      DO 50 I=1,NROWS
      M= I-1
      MH = M/2
      JS = 2*(I-1) +1
      IF(YBAR.NE.0.0) JS = JS+1
      IF(EL.EQ.0.0) JS = I
      DO 50 J=1,JS
      K=K+1
      KN = KN + 1
      N = I - J
      IF(EL.EQ.0.0) N = 1- J
      IF(YBAR.LT.0.0) N = -N
      WRITE (NT6,9210) M,N,C(KN),W(K),V(K)
 9210 FORMAT(2I5,5X,6E17.8)
   50 CONTINUE
      GO TO 53
C
   52 CONTINUE
      RN = NROWS
      K = RN* (RN/2.) + (RN/2.) + 1.0E-05
      IF(EL.EQ.0.) GO TO 53
      K = K+K
      IF(YBAR.EQ.0.)  K = K - NROWS
   53 CONTINUE
C
C
      IF (NSPATK .EQ. 0 .OR. NV .GT. NVCS-NPK) GO TO 55
      IF(NV.EQ.1) REWIND IAICSC
C
C         WRITE THE SPATIAL AICS ON A SCRATCH FILE
C
      CALL RDINIT
      PARM(1) = K1
      PARM(2) = XMACH
      PARM(4) = YBAR
      PARM(5) = EL
      ITYPE = 5HMIXED
      MXARRY = 6HMUAIC
      CALL WRTEMX(IAICSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
     1              MUAIC,ITYPE,2,NROWS,PARM,IRR)
      IF(IRR.NE.0) GO TO 6170
C
      MXARRY = 6H C
      CALL WRTEMX(IAICSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
     1              C,    ITYPE, 2,K,     PARM, IRR)
      IF(IRR.NE.0) GO TO 6170
C
      MXARRY = 3H W
```

```
      CALL WRTEMX(IAICSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
     1            W,     ITYPE, 2,K,      PARM, IRR)
      IF(IRR.NE.0) GO TO 6170
C
      MXARRY = 3H V
      CALL WRTEMX(IAICSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
     1            V,     ITYPE, 2,K,      PARM, IRR)
      IF(IRR.NE.0) GO TO 6170
C
      IF(NV.NE.NVCS-NPK) GO TO 55
      END FILE IAICSC
      REWIND IAICSC
   55 CONTINUE
      IF(NPLAIC.EQ.0.AND.NSPAIC.EQ.0) GO TO 1000
C
C         WRITE THE KERNEL ON TAPE
      IF(NV.LE.NVCS-NPK) GO TO 900
C
C         WRITE ON THE PLANAR KERNEL TAPE
      IF(NPLAIC.EQ.0) GO TO 1000
C
C         DETERMINE IF A PREVIOUS MATRIX WAS ON TAPE.
C         IF NKOT = 0 IT IS A NEW TAPE AND THERE ARE NO OLD ONES
C         IF NSIZE IS LESS THAN NROWS A MATRIX WAS EXPANDED OR THERE
C         WAS NONE WITH CORRESPONDING PARAMETERS
C
      IF(NKOT.EQ.0) GO TO 60
      IF(NSIZE.GE.NROWS) GO TO 1000
C
   60 CONTINUE
      NKOT = NKOT + 1
      AMACH(NKOT) = XMACH
      AKVAL(NKOT) = K1
      AERR(NKOT)  = ERR
      ISIZE(NKOT) = NROWS
      REWIND NPLAIC
      CALL RDINIT
      NMS = NKOT - 1
      ITYPE = 5HMIXED
      MXARRY = 6H C
      CALL WRTEMX(NPLAIC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
     1            C(NN), ITYPE, 2, K, PARM, IRR)
      IF(IRR.NE.0) GO TO 6150
C
      END FILE NPLAIC
      CALL RDINIT
      ITYPE = 5HMIXED
      MXARRY = 6HTAB
      CALL WRTEMX(NPLAIC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,500,ID,
     1            TAB,ITYPE,NKOT,3,PARM,IRR)
      IF(IRR.NE.0) GO TO 6150
C
      MXARRY = 6HISIZE
      CALL WRTEMX(NPLAIC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,  LWS,1,ID,
     1            ISIZE,ITYPE,1,NKOT,PARM,IRR)
      IF(IRR.NE.0) GO TO 6150
      END FILE NPLAIC
```

```
      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
C          THERE WAS NO OLD KERNEL TAPE THEREFORE SKIP TO NV-NFK AND
C          ADD TO TABLE OF CONTENTS.
C
      NKST = NV
      GO TO 920
  910 CONTINUE
C
C          IF NSIZ IS LESS THAN NROWS A MATRIX WAS EXPANDED OR THERE
C          WAS NONE WITH CORRESPONDING PARAMETERS.
C
      IF(NSIZ.GE.NROWS) GO TO 1000
C
C          THERE WAS AN OLD KERNEL TAPE THEREFORE SKIP TO NKST+1 AND
C          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
      REWIND NSPAIC
      CALL RDINIT
      ITYPE = 5HMIXED
      NMS =(NKST - 1)*4
      MXARRY = 6HMUAIC
      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 = 5HMIXED
      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
```

```
C
      MXARRY = 6H V
      CALL WRTEMX(NSPAIC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
     1            V, ITYPE,2,K,PARM,IRR)
      IF(IRR.NE.0) GO TO 6160
C
      END FILE NSPAIC
C
C           WRITE TABLE OF CONTENTS ARRAYS
      ITYPE = 5HMIXED
      MXARRY = 6HTAB
      CALL WRTEMX(NSPAIC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,500,ID,
     1            TAB,ITYPE,NKST,3,PARM,IRR)
      IF(IRR.NE.0) GO TO 6160
C
      MXARRY = 6HATAB
      CALL WRTEMX(NSPAIC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,500,ID,
     1            ATAB,ITYPE,NKST,3,PARM,IRR)
      IF(IRR.NE.0) GO TO 6160
C
      END FILE NSPAIC
      END FILE NSPAIC
      IF(NV.EQ.MVCS-NFK) OSPAIC = .TRUE.
 1000 CONTINUE
      RETURN
C
 6010 CONTINUE
      WRITE (NT6,9010) IGEOSC,IRR
      WRITE (NT6,9011) MXARRY
      GO TO 6100
C
 6050 CONTINUE
      WRITE (NT6,9050) NPLAIC,IRR
      WRITE (NT6,9011) MXARRY
      GO TO 6100
C
 6060 CONTINUE
      WRITE (NT6,9060) NSPAIC,IRR
      WRITE (NT6,9011) MXARRY
      GO TO 6100
C
 6150 CONTINUE
      WRITE (NT6,9150) NPLAIC,IRR
      WRITE (NT6,9151) MXARRY
      GO TO 6100
C
 6160 CONTINUE
      WRITE (NT6,9160) NSPAIC,IRR
      WRITE (NT6,9151) MXARRY
      GO TO 6100
C
 6170 CONTINUE
      WRITE (NT6,9160) IAICSC,IRR
      WRITE (NT6,9151) MXARRY
C
 6100 CONTINUE
      WRITE (NT6,9101) ID(1),ID(2)
```

```
      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
 6005 FORMAT(*0AIC 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 A6,8H 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(*0    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                IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC          FILES   00003
      COMMON /VICPAR/ YBAR,EL,NROWS,MUAIC(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  CTM(2), WTM(2), VTM(2)                                 KERNEL  00007
      EQUIVALENCE  (CSV,CTM), (WSV,WTM), (VSV,VTM)                      KERNEL  00008
      COMMON /BESFUN/ XIB(5), A(50,5)                                   BCSAICB 00001
      REAL K1,K1BAR                                                     KERNEL  00010
      COMPLEX ZERO                                                      KERNEL  00011
      DATA  EPS / 1.0E-4 /                                              KERNEL  00012
C                                                                       KERNEL  00013
C        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
      K1BAR = (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 + I                                                   KERNEL  00037
C                                                                       KERNEL  00038
   75 CONTINUE                                                          KERNEL  00039
      IF(MUAIC(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 105 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  00055
  106 CONTINUE                                                          KERNEL  00056
```

```
      IWARN = 0
      DO 108 J=1,5
      TAU = SQRT(XIB(J)*XIB(J) - EL2)
      TAUKM = (K1BAR/XMACH) * TAU
      XIB(J) = TAUKM
      CALL RANGE(TAUKM,N)
      IF(N.LE.100) GO TO 107
      WRITE (NT6,9005) N
 9005 FORMAT( 99H0*** THE ARGUMENT FOR A BESSEL FUNCTION YIELDS AN ORDER
     1 GREATER THAN 100. ORDER REDUCED TO 100. *** )
      N = 100
      IF(IWARN.EQ.1) GO TO 107
      IWARN = 1
  107 CONTINUE
      CALL BESSEL(TAUKM,A(1,J),N)
  108 CONTINUE
C
C         THERE ARE BOXES ON THIS ROW. FIND LEFT MOST BOX AND PROCEED
C         FROM LEFT TO RIGHT.
      ULEFT = SQRT(VBARU*VBARU - EL2)
      URIGHT= -ULEFT
      IHALF = (NTP+1)/2
      REM = ABS(YBAR) - 0.5
      IL   = ULEFT - REM
      IL =IHALF -IL
      IR = REM - URIGHT + 1.0
      IR = IHALF + IR
      IF(IL.LT.MUAIC(1,I)) GO TO 110
C
C         HYPERBOLA IS LESS THAN ALLOWED, REDUCE LIMITS.
      MUAIC(1,I) = IL
      GO TO 120
C
C         HYPERBOLA CROSSED A BOUNDARY, REDUCE CALCULATIONS.
  110 IL = MUAIC(1,I)
  120 CONTINUE
C
C         TEST RIGHT SIDE
      IF(IR.GT.MUAIC(2,I)) GO TO 130
C
C         HYPERBOLA IS LESS THAN ALLOWED, REDUCE LIMITS.
      MUAIC(2,I) = IR
      GO TO 140
C
C         HYPERBOLA CROSSES A BOUNDARY, REDUCE CALCULATIONS.
  130 IR = MUAIC(2,I)
  140 CONTINUE
C
C         DETERMINE INTEGRALS FOR BOXES IL TO IR.
      IF (IL .GT. IR) GO TO 950
      DO 900 ID=IL,IR
      U = IHALF - ID
      IU = U
      ULEFT = U + 0.5 + ABS(YBAR)
      URIGHT= ULEFT -1.0
      YMUBAR = ULEFT - 0.5
C
```

```
      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 C. .ER 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) = CBARL                                 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
C           ONLY HAS 2 SEGMENTS
      NINT = 2
      XIL(1) = ABS(EL)
      XIL(2) = CBARR
      XIL(3) = VBARU
      IUC(1)=3
      IUC(2)=1
      GO TO 700
C
  440 CONTINUE
C
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
      NINT = 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
C           TWO SEGMENTS
      NINT = 2
      XIL(3) = VBARU
      IUC(2)=0
      GO TO 700
C
C           THREE SEGMENTS
  450 NINT = 3
      XIL(3) = CBARL
      IUC(2)=1
      XIL(4) = VBARU
      IUC(3)=0
      GO TO 700
C
C           CENTER LINE BOX, BUT NOT APEX, HAS THREE SEGMENTS
  475 CONTINUE
      IF(CBARR.LE.VBARL) GO TO 220
      NINT = 3
      XIL(1) = VBARL
      XIL(2) = CBARR
```

```
      XIL(3) = CBARL
      XIL(4) = VBARU
      IUC(1) = 3
      IUC(2) = 1
      IUC(3) = 0
      GO TO 700
C
C           FULL BOX, ONLY ONE SEGMENT TO INTEGRATE
  500 CONTINUE
      NINT = 1
      XIL(1) = VBARL
      XIL(2) = VBARU
      IUC(1) = 0
C
C           LIMITS AND TYPES FOR ALL SEGMENTS ARE COMPLETED. INTEGRATE.
  700 CONTINUE
      IDX = ID
      IF(EL.EQ.0.0) IDX = ID - I + 1
      IX = IS + IDX - 1
      IF(C(IX).NE.0) GO TO 900
      DO 800 INT=1,NINT
      CSV = (0.,0.)
      WSV = (0.,0.)
      VSV = (0.,0.)
      IFLAG=0
C
C           CALL ROMBERG INTEGRATION FOR REAL PART
      CALL ROMBER(XIL(INT),XIL(INT+1),IUC(INT),ERR,IFLAG,K1BAR,YMUBAR,
     1            EL,XMACH,CTM(1),WTM(1),VTM(1) )
      IF(K1.EQ.0.0) GO TO 750
      IFLAG = 1
C
C           CALL ROMBERG INTEGRATION FOR IMAGINARY PART
      CALL ROMBER(XIL(INT),XIL(INT+1),IUC(INT),ERR,IFLAG,K1BAR,YMUBAR,
     1            EL,XMACH,CTM(2),WTM(2),VTM(2)  )
  750 CONTINUE
      C(IX) = C(IX) + CSV
      W(IX) = W(IX) + WSV
      V(IX) = V(IX) + VSV
  800 CONTINUE
C
  900 CONTINUE
      GO TO 1000
C
  950 CONTINUE
      MUAIC(1,I) = 0
      MUAIC(2,I) = 0
C
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ROMBER(XILL,XILU,IUC,ERR,IFLAG,K1BAR,YMUBAR,EL ,XMACH,
     1                  C, W, V )
      DIMENSION  XI(512),FXIC(512),FXIW(512),FXIV(512)
      DIMENSION  A(11,11), AW(11,11), AV(11,11), VT(2)
      REAL       K1BAR
      PIE = 3.141592654
C
C         XILL   - XI LOWER LIMIT OF INTEGRATION
C         XILU   - XI UPPER LIMIT OF INTEGRATION
C         IUC    - FLAG INDICATING TYPE OF BOX OR EDGE CONDITION OF
C                  INTERVAL TO BE INTEGRATED.
C                  IUC = 0, FULL BOX
C                      = 1, LEFT SIDE OF INTERVAL IS EDGE OF MACH HYP.
C                      = 2, RIGHT SIDE OF INTERVAL IS EDGE OF MACH HYP.
C                      = 3, BOTH SIDES OF INTERVAL IS EDGE OF MACH HYP.
C         ERR    - CONVERGENCE TEST CRITERIA
C         IFLAG  - INDICATOR OF REAL OR IMAGINARY PARTS
C                  IFLAG = 0, REAL PART
C                        = 1, IMAGINARY PART
C         K1BAR  - FUNCTION OF REDUCED FREQUENCY, MACH NUMBER
C         YMUBAR - COORDINATE HORIZONTALLY OF PULSE SENDING BOX
C         EL     - DISTANCE OF RECEIVING BOX ABOVE SENDING PLANE
C         XMACH  - MACH NUMBER
C         C      - C COEFFICIENT
C         V      - V COEFFICIENT
C         W      - W COEFFICIENT
C
C
C         CALCULATE INITIAL VALUES AT END POINTS
C
      EL2 = EL*EL
      VK = -XMACH/(PIE*K1BAR)
      XI(1) = XILL
      XI(2) = XILU
      CALL FUNCT(2,XI,FXIC,FXIW,IFLAG,K1BAR,EL ,YMUBAR,IUC,XMACH,BESSW)
      IF(EL.EQ.0.0) GO TO 101
      IF(K1BAR.EQ.0.0) GO TO 101
      IF(IUC.EQ.3) GO TO 101
      CALL VFUNC (2, XI,FXIV,IFLAG,K1BAR,EL,YMUBAR,IUC,XMACH,1,VT)
  101 CONTINUE
      TERM1 = FXIC(1)/XI(1)
      TERM2 = FXIC(2)/XI(2)
      HINT = 0.5*(XILU-XILL)
      TC = HINT *(FXIC(1)+FXIC(2))
      TW = HINT *(FXIW(1)+FXIW(2))
      A(1,1)  = TC
      AW(1,1) = TW
      IF(EL.EQ.0.0) GO TO 102
      IF(K1BAR.EQ.0.0) GO TO 102
      IF(IUC.EQ.3) GO TO 102
      TV = HINT *(FXIV(1)+FXIV(2))
      AV(1,1) = TV
  102 CONTINUE
C
      DO 30 M=2,11
      I= 2**(M-1)
      H = (XILU - XILL)/I
```

```
C
C        DETERMINE XI LOCATIONS TO EVALUATE FUNCTION AT
      DO 3 J=2,I,2
      JJ = J/2
      XI(JJ) = XILL + (J-1)*H
    3 CONTINUE
      CALL FUNCT(JJ,XI,FXIC,FXIW,IFLAG,K1BAR,EL  ,YMUBAR,IUC,XMACH,DUMMY)
      IF(EL.EQ.0.0) GO TO 103
      IF(K1BAR.EQ.0.0) GO TO 103
      IF(IUC.EQ.3) GO TO 103
      CALL VFUNC (JJ,XI,FXIV,IFLAG,K1BAR,EL,YMUBAR,IUC,XMACH,0,DUMMY)
  103 CONTINUE
C
C        DETERMINE TRAPEZOIDAL AREA WITH THE NEW FUNCTION EVALUATIONS
      TMNC = 0.0
      TMNW = 0.0
      TMNV = 0.0
      DO 5 J=1,JJ
      TMNC = TMNC + FXIC(J)
      TMNW = TMNW + FXIW(J)
      IF(EL.EQ.0.0) GO TO 5
      IF(K1BAR.EQ.0.0) GO TO 5
      IF(IUC.NE.3)TMNV = TMNV + FXIV(J)
    5 CONTINUE
      TC = 0.5*TC + H*TMNC
      TW = 0.5*TW + H*TMNW
C
C        PUT THE NEW AREAS INTO THE ARRAY AND PERFORM EXTRAPOLATION
      A(M,1) = TC
      AW(M,1)= TW
      IF(EL.EQ.0.0) GO TO 104
      IF(K1BAR.EQ.0.0) GO TO 104
      IF(IUC.EQ.3) GO TO 104
      TV = 0.5*TV + H*TMNV
      AV(M,1) = TV
  104 CONTINUE
      DO 10 N=2,M
      A(M,N) = ((4**(N-1))*A(M,N-1)-A(M-1,N-1))/(4**(N-1)-1)
      AW(M,N)= ((4**(N-1))*AW(M,N-1)-AW(M-1,N-1))/(4**(N-1)-1)
      IF(EL.EQ.0.0) GO TO 10
      IF(K1BAR.EQ.0.0) GO TO 10
      IF(IUC.EQ.3) GO TO 10
      AV(M,N)= ((4**(N-1))*AV(M,N-1)-AV(M-1,N-1))/(4**(N-1)-1)
   10 CONTINUE
C
C        DETERMINE IF THE TECHNIQUE HAS REACHED SUFFICIENT CONVERGENCE
      C = A(M,M)
      W = AW(M,M)
      IF(EL.EQ.0.0) GO TO 105
      IF(K1BAR.EQ.0.0) GO TO 105
      IF(IUC.EQ.3) GO TO 105
      V = AV(M,M)
  105 CONTINUE
      RERR = ABS(ERR*C)
      MM1 = M-1
      UDIF = ABS(A(M,MM1)-A(MM1,MM1))
      RDIF = ABS(A(M,M)  -A(M  ,MM1))
```

```
      DELS = 0.5*(UDIF+RDIF)
      IF(DELS.GT.RERR) GO TO 30
C
C         C HAS CONVERGED,  TEST FOR W CONVERGENCE
      IF(EL.EQ.0.0) GO TO 50
      RERR = ABS(ERR*W)
      UDIF = ABS(AW(M,MM1) - AW(MM1,MM1))
      RDIF = ABS(AW(M,M)   - AW(M  ,MM1))
      DELS = 0.5 *(UDIF + RDIF)
      IF(DELS.GT.RERR) GO TO 30
C
C         C AND W HAVE CONVERGED, TEST FOR V CONVERGENCE
C
      IF(K1BAR.EQ.0.0 .OR. IUC.EQ.3) GO TO 50
      RERR = ABS(ERR*V)
      UDIF = ABS(AV(M,MM1) - AV(MM1,MM1))
      RDIF = ABS(AV(M,M)   - AV(MM1,MM1))
      DELS = 0.5 *(UDIF + RDIF)
      IF(DELS.LE.RERR) GO TO 50
C
C         HAS NOT CONVERGED MAKE ANOTHER LOOP.
   30 CONTINUE
   50 CONTINUE
      C = -C/PIE
      IF(EL .EQ.0.0) GO TO 70
      W  = (EL/PIE)*(W*TERM2-TERM1  + BESSW )
      IF(K1BAR.EQ.0.0) GO TO 65
      IF(IUC.EQ.3) GO TO 70
      V = VK*(V +VT(2)-VT(1))
      GO TO 70
   65 CONTINUE
      ETAL = YMUBAR - 0.5
      IF(IUC.EQ.2) ETAL = - SQRT(XILU*XILU - EL2)
      ETAU = YMUBAR + 0.5
      IF(IUC.EQ.1) ETAU =   SQRT(XILU*XILU - EL2)
      XILU2 = XILU * XILU
      XILL2 = XILL * XILL
      ETAU2 = ETAU * ETAU
      ETAL2 = ETAL * ETAL
      SU = ETAU2 + EL2
      SL = ETAL2 + EL2
      S1 = XILU2 - SU
      S2 = XILU2 - SL
      S3 = XILL2 - SU
      S4 = XILL2 - SL
      V1 = 0.0
      V2 = 0.0
      V3 = 0.0
      V4 = 0.0
      IF(S1.GT.0.0) V1 = ALOG((XILU+SQRT(S1))/SQRT(SU))
      IF(S2.GT.0.0) V2 = ALOG((XILU+SQRT(S2))/SQRT(SL))
      IF(S3.GT.0.0) V3 = ALOG((XILL+SQRT(S3))/SQRT(SU))
      IF(S4.GT.0.0) V4 = ALOG((XILL+SQRT(S4))/SQRT(SL))
      V = (-1.0/PIE)*(V1-V2-V3+V4)
   70 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FUNCT(K,XI,FXIC,FXIW,IFLAG,K1BAR,EL ,YMUBAR,IUC,
     1                 XMACH,BESSY)
      DIMENSION  XI(512),FXIC(512),FXIW(512),A(50)
C
C         K      - NUMBER OF FUNCTIONS TO EVALUATE
C         XI     - VARIABLE OF INTEGRATION
C         FXIC   - FUNCTIONAL VALUE FOR C EQUATION
C         FXIW   - FUNCTIONAL VALUE FOR W EQUATION
C         IFLAG  - INDICATOR OF REAL OR IMAGINARY PARTS
C                  IFLAG = 0, REAL PART
C                  IFLAG = 1, IMAGINARY PART
C         K1BAR  - FUNCTION OF REDUCED FREQUENCY AND MACH NUMBER
C         EL     - DISTANCE OF RECEIVING BOX ABOVE SENDING PLANE
C         YMUBAR - COORDINATE HORIZONTALLY OF PULSE SENDING BOX
C         IUC    - FLAG INDICATING TYPE OF BOX OR EDGE CONDITION OF
C                  INTERVAL TO BE INTEGRATED.
C         XMACH  - MACH NUMBER
C         BESSY  - EVALUATION OF END POINTS FOR W COEFFICIENTS.
C
      REAL    K1BAR
      PIE = 3.141592654
      PIE2 = 1.570796327
      EL2 = EL*EL
      BESSY = 0.0
C
      DO 1000  I=1,K
C
C         SET UP CONSTANTS
      TAU = SQRT(XI(I)*XI(I) - EL2)
      TAUKM = (K1BAR/XMACH)*TAU
      EPOW = K1BAR*XI(I)
      IF(ABS(TAU).LT.1.0E-06) GO TO 25
      THETAU = (YMUBAR + 0.5)/TAU
      THETAL = (YMUBAR-0.5)/TAU
      GO TO 50
   25 CONTINUE
      THETAL = 0.0
      THETAU = 0.0
   50 CONTINUE
C
      IF(IFLAG.EQ.0) GO TO 100
C
C          IMAGINARY PART
      EXPN = -SIN(EPOW)
      EXPNW = (EPOW*COS(EPOW) - SIN(EPOW))/(XI(I)*XI(I))
      GO TO 200
C
C          REAL PART
  100 EXPN = COS(EPOW)
      EXPNW =(COS(EPOW) + EPOW*SIN(EPOW))/(XI(I)*XI(I))
C
  200 CONTINUE
      CALL BFUNC(TAUKM,A,N)
      IF(EL .EQ.0.0) GO TO 250
      IF(I.GT.1) GO TO 250
      IF(IUC.NE.3) GO TO 250
      EXL =  ABS(EL) - XI(1)
```

```
      IF(ABS(EXL).GT.1.0E-05) GO TO 250                                FUNCT  00059
      BESSY = (EXPN*PIE)/EL                                            FUNCT  00060
  250 CONTINUE                                                         FUNCT  00061
      BESSO = A(1)                                                     FUNCT  00062
      BTERM = 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                                                     FTNXX  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                                            FTNXX  00050
      PTERM = PSIGN/R                                                  FUNCT  00102
      BTERM = BTERM + PTERM*A(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) = EXPN *(BESSO*(C1-C2) + BTERM )                         FUNCT  00107
      FXIW(I) = 0.0                                                    FUNCT  00108
      IF(EL2.EQ.0.0) GO TO 1000                                        FUNCT  00109
      FXIW(I) = EXPNW*(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(I) = AV(II) * C                                                BESSEL  00035
      I = I + 1                                                        BESSEL  00036
      IF(I.EQ.50) GO TO 99                                             BCSAICA 00001
   90 CONTINUE                                                         BESSEL  00037
   99 CONTINUE                                                         BESSEL  00038
      RETURN                                                           BESSEL  00039
      END                                                              BESSEL  00040
```

```
      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
```

```
      SUBROUTINE VFUNC(K,XI,FXIV,IFLAG,K1BAR,EL,YMUBAR,IUC,XMACH,IND,VT) VFUNCT  00002
      DIMENSION XI(256),FXIV(256),VT(2)                                VFUNCT  00003
      REAL K1BAR                                                       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       K1BAR  - 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 = K1BAR*XI(I)                                               VFUNCT  00031
         XI2  = XI(I)*XI(I)                                            VFUNCT  00032
      FREQM = K1BAR/XMACH                                              VFUNCT  00033
      YMUP2 = (YMUBAR + 0.5)*(YMUBAR + 0.5)                            VFUNCT  00034
      YMUM2 = (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                                                       VFUNCT  00059
      C2R = XI2 - YMUM2 - EL2                           VFUNCT  00060
      IF(ABS(C2R).LT.EPS) GO TO 400                     VFUNCT  00061
      C2 = SIN(FREQM*SQRT(C2R))                         VFUNCT  00062
C                                                       VFUNCT  00063
  400 CONTINUE                                          VFUNCT  00064
      FXIV(I) = EXPN * (C1-C2)                          BCSAICA 00006
      IF(IND.EQ.1) VT(I) =(EXPNV/XI(I)) * (C1-C2)       BCSAICA 00007
  500 CONTINUE                                          VFUNCT  00067
      RETURN                                            VFUNCT  00068
      END                                               VFUNCT  00069
```

```
      SUBROUTINE BFUNC(X,AV,N)                                          BFUNC   00002
C                                                                       BFUNC   00003
C          X   - ARGUMENT FO THE BESSEL FUNCTION                        BFUNC   00004
C          AV - EVALUATION OF BESSEL FUNCTION                           BFUNC   00005
C          N   - MAX ORDER OF BESSEL FUNCTION                           BFUNC   00006
C                                                                       BFUNC   00007
      COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUTP,         FILES   00002
     1                IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC          FILES   00003
      COMMON /BESFUN/ XIB(5), A(50,5)                                   BCSAICB 00006
      DIMENSION   AV(1)                                                 BFUNC   00010
C                                                                       BFUNC   00011
C          FIND INTERVAL X IS IN                                        BFUNC   00012
      IF(X.GE.XIB(1).AND.X.LE.XIB(5)) GO TO 50                          BCSAICB 00007
      EPS = 1.0E-04                                                     BFUNC   00014
      I = 1                                                             BFUNC   00015
      IF (ABS(X-XIB(I)) .LE. EPS*XIB(I)) GO TO 150                      BFUNC   00016
      I= 5                                                              BCSAICB 00008
      IF (ABS(X-XIB(I)) .LE. EPS*XIB(I)) GO TO 150                      BFUNC   00018
      WRITE (NT6,9005) X,XIB(1),XIB(5)                                  BCSAICB 00009
 9005 FORMAT( 68H0*** ERROR - THE ARGUMENT FOR A BESSEL FUNCTION IS OUT BFUNC   00020
     1OF RANGE. ***/  14H0ARGUMENT    = F11.6/14H LOWER LIMIT = F11.6 / BFUNC   00021
     2    14H UPPER LIMIT = F11.6 )                                     BFUNC   00022
      CALL FLUSH(1)                                                     BFUNC   00023
   50 CONTINUE                                                          BFUNC   00024
      I = 2                                                             BFUNC   00025
  100 IF(X-XIB(I)) 200,150,125                                          BFUNC   00026
  125 CONTINUE                                                          BFUNC   00027
      I = I + 1                                                         BFUNC   00028
      GO TO 100                                                         BFUNC   00029
  150 CONTINUE                                                          BFUNC   00030
C                                                                       BFUNC   00031
C          X EQUALS XIB(I)    DO NOT INTERPOLATE                        BFUNC   00032
C                                                                       BFUNC   00033
      N = 1                                                             BFUNC   00034
  160 CONTINUE                                                          BFUNC   00035
      AV(N) = A(N,I)                                                    BFUNC   00036
      IF(A(N+1,I).EQ.0) GO TO 400                                       BFUNC   00037
      N = N + 1                                                         BFUNC   00038
      GO TO 160                                                         BFUNC   00039
  200 CONTINUE                                                          BFUNC   00040
      DXX =  (X-XIB(I-1))/(XIB(I) -XIB(I-1))                            BFUNC   00041
      N = 1                                                             BFUNC   00042
  300 CONTINUE                                                          BFUNC   00043
      AV(N) = A(N,I-1) + DXX * (A(N,I) - A(N,I-1))                      BFUNC   00044
      IF(A(N+1,I-1).EQ.0.) GO TO 400                                    BFUNC   00045
      N = N+ 1                                                          BFUNC   00046
      GO TO 300                                                         BFUNC   00047
  400 CONTINUE                                                          BFUNC   00048
      RETURN                                                            BFUNC   00049
      END                                                               BFUNC   00050
```

```
      OVERLAY(AFMBOX,1,5)                                               NWVPMBX 00002
      PROGRAM NWVPMBX                                                   NWVPMBX 00003
C                                                                       NWVPMBX 00004
C        THIS SECTION CONTROLS THE COMPUTATION OF BOX NORMAL WASH       NWVPMBX 00005
C        VALUES AND VELOCITY POTENTIAL DIFFERENCES.  THE NORMAL WASH    NWVPMBX 00006
C        VALUES MAY BE PRINTED, BUT OTHERWISE ARE NOT SAVED.  THE       NWVPMBX 00007
C        VELOCITY POTENTIAL DIFFERENCES ARE PLACED ON SCRATCH FILE      NWVPMBX 00008
C        IVPSC, TWO MATRICES PER MODE SHAPE                             NWVPMBX 00009
C                                                                       NWVPMBX 00010
      COMMON PKERNL(1640)                                               NWVPMBX 00011
      COMPLEX PKERNL                                                    NWVPMBX 00012
      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,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 / KERN / ERR,MXSKRN,IPKERN,NPLKRN,NSPATK,NROWEA            KERN    00002
      COMMON /KVAL  / IKVAL,XKVAL(20), XKS(20)                          KVAL    00002
      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,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       BCSPRB  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                        BCSPRB  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,MTYPEW,MTYPET                            MODCOM  00002
      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
      COMMON /SAMPLW/ ISMPLW,ICHORD(10),IBOXF(10),IBOXL(10),ZLOC(10)    SAMPLW  00002
      COMMON /PAICS / NMAK,  NTTK,  NRWTK,  XLVTK,   PAIC(4,50)         PAICS   00002
      INTEGER PAIC                                                      PAICS   00003
      DIMENSION NK(4)                                                   PAICS   00004
      EQUIVALENCE (NMAK,NK(1))                                          PAICS   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 /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(1080), TVP(250), TEXLOC(250), FEXLOC(250), DELTAP  00003
     1               IPNTRM(2,100),NPNTRS, IOVLAP                       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
      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),IOVLAPN   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(LBXCDW,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
C                                                                       NWVPMBX 00033
C                IPNTRM(2,NROWS),IPNTDW(2,NROWS),IPNTSD(2,NSROWS)       NWVPMBX 00034
C                                                                       NWVPMBX 00035
      COMMON /LROT  / LROT                                              NWVPMBX 00036
      COMMON /CHECKPR/ DPPCPR,GEOCPR,MODCPR,AICCPR,NWSCPR,SMCPR,GAFCPR  CHECKPR 00002
      LOGICAL  DPPCPR, GEOCPR, MODCPR, AICCPR,NWSCPR,  SMCPR, GAFCPR    CHECKPR 00003
      LOGICAL CHECKPR                                                   NWVPMBX 00038
      EQUIVALENCE (CHECKPR,NWSCPR)                                      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                                                                       NWVPMBX 00048
      PSIDIF = PSIT - PSIW                                              NWVPMBX 00049
      IPNTLS = LPNTSD                                                   NWVPMBX 00050
      LIMDWS = LSDW                                                     NWVPMBX 00051
      LROT = NSUBDV + NSUBCN                                            NWVPMBX 00052
      NTVP  = (MYBSW + MYBST)                                           NWVPMBX 00053
      PARMW(2) = B1                                                     NWVPMBX 00054
      PARMW(3) = XMACH                                                  NWVPMBX 00055
      IPARMW(4) = NMODES                                                NWVPMBX 00056
C                                                                       NWVPMBX 00057
C                                                                       NWVPMBX 00058
      REWIND IGEOSC                                                     NWVPMBX 00059
      REWIND MODESC                                                     NWVPMBX 00060
      REWIND IVPSC                                                      NWVPMBX 00061
C                                                                       NWVPMBX 00062
      XKVL = XKVAL(IKVAL)                                               NWVPMBX 00063
C                                                                       NWVPMBX 00064
C           READ GEOMETRIC INFORMATION                                  NWVPMBX 00065
C              WING BOX CODES                                           NWVPMBX 00066
      MXARRY = 9HBOX CODES                                              NWVPMBX 00067
      CALL RDINIT                                                       NWVPMBX 00068
      ITYPE = 5HMIXED                                                   NWVPMBX 00069
      K = LBXCDW                                                        NWVPMBX 00070
      CALL READMX (IGEOSC, MXRD, .F., NFS,NMS,LS, NMR, K, NID,ID, ITYPE, NWVPMBX 00071
```

```
     1           LRS,  IBOXW,  M,N,  PARM, IRR)
      IF (IRR .NE. 0) GO TO 910
      MXBBS = M
      IF (NSURF .EQ. 1 .OR. COPLAN) GO TO 20
C
C           TAIL BOX CODES
      MXARRY = 10HTAIL CODES
      CALL RDINIT
      ITYPE = 5HMIXED
      CALL READMX (IGEOSC, MXRD, .F., NFS,NMS,LS, NMR, K, NID,ID, ITYPE,
     1           LRS,  IBOXW(MXBBS+1,1), M,N,  PARM, IRR)
      IF (IRR .NE. 0) GO TO 910
      ISUBT = MXBBS + 2 - IXBST
C           = SUBSCRIPT FOR IBOXW TO GET TAIL CODES
C
   20 CONTINUE
      MXARRY   10H    FEXLOC
      CALL RDINIT
      ITYPE = 5HMIXED
      CALL READMX(IGEOSC, MXRD,.F., NFS,NMS,LS, NMR, 1, NID,ID, ITYPE,
     1    LRS, FEXLOC, M,N, PARM, IRR)
      IF (IRR .NE. 0) GO TO 910
C
      MXARRY =10H    TEXLOC
      CALL RDINIT
      ITYPE = 5HMIXED
      CALL READMX(IGEOSC, MXRD,.F.,  NFS,NMS,LS, NMR, 1, NID,ID, ITYPE,
     1    LRS, TEXLOC, M,N, PARM, IRR)
      IF (IRR .NE. 0) GO TO 910
      IF(M .NE. 1) GO TO  930
C
      IF (PSIW .NE. 0 .AND. DIHW) GO TO 30
      IF (ISMFLW .NE. 0) GO TO 30
      IF (NSURF .EQ. 1) GO TO 40
      IF (CAPL .NE. 0) GO TO 30
      IF (PSIDIF .NE. 0) GO TO 30
      IF (PSIT .NE. 0 .AND. DIHT) GO TO 30
      GO TO 40
C
C         READ THE ARRAY OF AIC TABLE OF CONTENTS
   30 CONTINUE
      MXARRY = 9HSPAT. TOC
      CALL RDINIT
      NMS = 2
      K = 4
      CALL READMX (IGEOSC, MXRD, .F., NFS,NMS,LS, NMR, K, NID,ID, ITYPE,
     1           LRS, PAIC, M,N, PARM, IRR )
      IF (IRR .NE. 0) GO TO 910
      DO 35 I = 1,4
      NK(I) = IPARM(I+2)
   35 CONTINUE
   40 CONTINUE
      REWIND IGEOSC
C
C         GET POINTER ARRAY FOR MODES
      CALL RDINIT
      ITYPE = 5HMIXED
```

```
      CALL READMX(MODESC, MXRD,.F.,  NFS,NMS,LS, NMR, 2, NID,ID, ITYPE,
     1    LRS, IPNTRM, M,N, PARM, IRR)
      IF (IRR .NE. 0) GO TO 912
      IF (M .NE. 2) GO TO 931
      NPNTRS = N
      IOVLAP = IPARM(3)
C         IOVLAP = NUMBER OF ROWS TO ALLOW FOR TAIL OVERLAP (TAIL ONLY)
C         NPNTRS = TOTAL NUMBER OF ROWS ON BOTH SURFACES, + 1 .
C                  (INCLUDES OVERLAP IF SPATIAL)
C
C         SET UP POINTER ARRAY FOR UNSUBDIVIDED DOWNWASHES
      IP = 1
      MYBB = MYBBW
      IF (COPLAN) MYBB = MAX0(MYBB,MYBBT)
      MYBBS = MYBB*NSUBDV
      MXBB = MXBBW
      IF (COPLAN) MXBB = MXBT
      CALL POINTR( 1,MXBB,MYBB,  .F., .T., IBOXW,LBXCDW, LPNTDW,1,
     1            , IP, IPNTDW)
      MXB = MXBW
      IF (COPLAN) MXB = MXBT
      MYB = MYBW
      IF(COPLAN) MYB = MYBT
      IOVLAPN = 0
      IF (NSURF .EQ. 1 .OR. COPLAN) GO TO 50
      MXB = MXBT
      IXBUT = (IXBT-IXBW)/NSUBDV + 1
      IP = MXBBW + 1
      IPNT = IPNTDW(1,IP)
      CALL POINTR(IXBUT, MXBT-IXBUT+1, MYBBT, .F.,.T., IBOXW(ISUBT,1),
     1            LBXCDW,LPNTDW, IPNT, IP, IPNTDW)
      IF (MXBBW .GE. IXBUT) IOVLAPN = MXBBW - IXBUT + 1
   50 CONTINUE
C
C         LOOP ON MODE SHAPES
      DO 500 IMODE = 1,NMODES
C
C         ZERO OUT THE DOWNWASH AND VELOCITY POTENTIAL ARRAYS
      LIM = IPNTDW(1,IP) - 1
      DO 80 I = 1,LIM
      DWRUS(I) = XINIT
      DWRLS(I) = XINIT
   80 CONTINUE
      LIM = IPNTRM(1,NPNTRS) - 1
      DO 85 I = 1,LIM
      DELPHI(I) = (0.,0.)
   85 CONTINUE
C
C
C         READ IN MODE SHAPE
      CALL RDINIT
      MXARRY = 10HMODE SHAPE
      ITYPE = 4HREAL
      CALL READMX(MODESC, MXRD,.F., NFS,NMS,LS, NMR, 2, NID, ID, ITYPE,
     1    LRS, DEFSL,  M,N, PARM, IRR)
      IF (IRR .NE. 0) GO TO 912
      IF (M .NE. 2) GO TO 931
```

```
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. PRNWD ) 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,MXBB,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, ENRLS,1, 1,MXBB,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,MXBBW+1) - 1                                       NWVPMBX 00220
      DO 120 I = 1,LIM                                                  NWVPMBX 00221
      ENRUS(I) = ENRLS(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(ISUBT,1),LBXCDW, PKERNL(IPKERN), PKERNL,      NWVPMBX 00227
     1              .F., DIHT)                                          NWVPMBX 00228
C        OPTIONAL PRINT OF NORMAL WASHES                                NWVPMBX 00229
      IF ( .NOT. (CHECKPR .OR. PRNWD ) 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,IMODE, ENRLS,1, IXBUT,MXBT,MYBBT,
     1             IPNTDW(1,IOVLAPN + 1)  )
      TITL(1) = 10HTAIL LOWER
      CALL PRINTR( TITL,IMODE, ENRLS,1, IXBUT,MXBT,MYBBT,
     1             IPNTDW(1,IOVLAPN + 1)  )
  135 CONTINUE
C
C        WRITE THE RESULTS ON SCRATCH FILES
  140 CONTINUE
      PARMW(1) = XKVL
      DO 210 I = 1,4
  210 PARM(I) = PARMW(I)
C
C   XX    WRITE DELTA PHIS, TEMPORARILY AS A REAL MATRIX, WRTETP    XX
      M = 2
      ITYPE = 4HREAL
      N = IPNTRM(1,MXB+IOVLAP+1) - 1
      CALL WRTEMX(IVPSC, .F., .F., NFS,NMS,LS, NMR,LWS, 2, ID, DELPHI,
     1     ITYPE, M,N,PARM, IRR)
      IF (IRR .NE. 0) GO TO 920
C
      M = 2
      N = NTVP
      CALL WRTEMX(IVPSC, .F., .F., NFS,NMS,LS, NMR, LWS,  2, ID, TVP,
     1     ITYPE, M,N,PARM, IRR)
      IF (IRR .NE. 0) GO TO 920
C
      IF (.NOT. PRVP) GO TO 230
      TITL(1) = 8H    WING
      TITL(2) = 10HVELOCITY P
      TITL(3) = 10HOTENTIALS
      M = MXBW
      IF (.N. COPLAN) GO TO 220
      TITL(1) = 10HWING/TAIL
      M = NPNTRS - 1
  220 CALL PRINTR( TITL, IMODE, DELPHI,1, 1,M, MYB, IPNTRM)
      IF (NSURF .EQ. 1 .OR. COPLAN) GO TO 230
      TITL(1) = 8H    TAIL
      CALL PRINTR( TITL, IMODE, DELPHI, 1, IXBUT,MXBT, MYBT,
     1             IPNTRM(1,IOVLAP+1)  )
  230 CONTINUE
C        ARE SAMPLE WASHES DESIRED -
      IF (NSURF .EQ. 2 .OR. ISMPLW .EQ. 0) GO TO 500
C        YES.  IS SAMPLE WASH PRINTOUT DESIRED
      IF (.NOT.(PRDW .OR. PRSW) ) GO TO 500
C        LOOP ON CHORDS FOR WHICH SAMPLE-WASH IS DESIRED
      DO 300 JCHRD = 1,ISMPLW
      JT = ICHORD(JCHRD)
      IFRST = IBOXF(JCHRD)
      ILAST = IBOXL(JCHRD)
      CALL SMPLW( IBOXW,LBXCDW,JCHRD,JT,IFRST,ILAST)
  300 CONTINUE
C
  500 CONTINUE
C         END OF LOOP ON MODE SHAPES, FROM STATEMENT 50*
C
      RETURN
```

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

```
     1    6H FILE ,A10, 4H ***)                                          NWVPMBX 00357
 9300 FORMAT(1H0,   48H*** MATRIX READ ERROR.  THE  M  DIMENSION SHOULD  NWVPMBX 00358
     1   4H BE ,I2, 4H ***)                                              NWVPMBX 00359
 9500 FORMAT(16H0 *** ERROR CODE ,I5, 28H WHILE READING THE FOLLOWING    NWVPMBX 00360
     1  11H MATRIX *** )                                                 NWVPMBX 00361
 9520 FORMAT(16H0 *** ERROR CODE ,I5, 28H WHILE WRITING THE FOLLOWING    NWVPMBX 00362
     1  11H MATRIX *** )                                                 NWVPMBX 00363
 9600 FORMAT( 5X,*MATRIX ID = *,10A10 / (20X,10A10)  )                   NWVPMBX 00364
 9614 FORMAT( 5X,22HMATRIX INDEX (NAME) = ,I5,2H (A10,1H) /              NWVPMBX 00365
     1        5X,33HLEVEL NUMBER READ (OR WRITTEN) = O2,2H (,O2,1H)  )   FTNX1   00056
 9620 FORMAT( 5X,*MATRIX ID = *, A10, I10 )                              NWVPMBX 00367
 9622 FORMAT( 5X,11HPARAMETERS, 10E11.3 /10X, 9H(INTEGER), I7,9I11)      NWVPMBX 00368
 9624 FORMAT( 5X,15HFILE SPACING = ,I3, 19H, MATRIX SPACING = ,I3 )      NWVPMBX 00369
 9630 FORMAT( 5X,A10,21H ARRAY WAS BEING USED )                          NWVPMBX 00370
 9640 FORMAT( 5X,*MATRIX TYPE - *,A10, *, DIMENSIONED (*I4,* X*,I4,*)*)  NWVPMBX 00371
 9900 FORMAT(54H0*** ERROR OCCURRED DURING VELOCITY POTENTIAL CALCULAT   NWVPMBX 00372
     1    8HIONS *** )                                                   NWVPMBX 00373
C                                                                        NWVPMBX 00374
      END                                                                NWVPMBX 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 COPLANAR, .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,INFSP,NPLAIC,NSPAIC,NOUTP,         FILES   00002
     1                IOUFSP,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,KDLPHI, 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,MXBBW,MYBW,MYBBW,MXBSW,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,CAPL                            GEOM2   00003
      COMMON / MODES/ SYM,SYMT,MTYPEW,MTYPET                            MODCOM  00002
C                                                                       VELPOT  00032
      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 /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, IOVLAP                      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 /NWASHES/ IPNTDW(2,100),ENRUS(1275), ENRLS(1275),IOVLAPN   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(LBXCDW,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,NWSCPR,SHCPR,GAFCPR     CHECKPR 00002
      LOGICAL  DPPCPR, GEOCPR, MODCPR, AICCPR,NWSCPR,  SHCPR, GAFCPR       CHECKPR 00003
      EQUIVALENCE (CHECKPR,NWSCPR)                                          VELPOT 00040
      LOGICAL CHECKPR                                                       VELPOT 00041
C                                                                           VELPOT 00042
      COMPLEX   AZERO, DELPH,  DELPHA,DELPHB,DELPHC, B, SDELPH,COEF         VELPOT 00043
      COMPLEX   ENRULU(50),ENRLLL(50),ENRURW(50),ENRULW(50), DPHIL(50)      VELPOT 00044
      COMPLEX   ENDIF, 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                                                         FTNX1  00057
      DIMENSION XINITX(2)                                                   FTNX1  00058
      EQUIVALENCE (XINIT,XINITX)                                            FTNX1  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 /                              FTNX1  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        MXBS   = LAST SUBDIVIDED ROW TO CONSIDER (TO LAST CONTROL PNT)     VELPOT 00065
      IF (WING) GO TO 80                                                    VELPOT 00066
      IOVLP = IOVLAP                                                        VELPOT 00067
      IOVLPN = IOVLAPN                                                      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
   80 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
      MXBS = (MXB-1)*NSUBDV + IXBW
      MXBS = MAX0(MXBS,MXBSW)
      GO TO 100
   85 CONTINUE
      MYBB  = MAX0(MYBBW,MYBBT)
      MYBBS = MAX0(MYBBSW,MYBBST)
   90 CONTINUE
      MXB   = MXBT
      MXBS  = MXBST
C
  100 CONTINUE
      PSIS2 = 2*PSIS
      PSISUM = PSIW + PSIT
      CPSIS2 = COS(PSIS2)
      SPSIS2 = SIN(PSIS2)
      CPSISM = COS(PSISUM)
      SPSISM = SIN(PSISUM)
      IRFB = IXB - NSUBD2
C           =  ROW OF FIRST SUBDIVIDED BOX IN THE FIRST ROW OF THE
C              UNSUBDIVIDED BOXES
      MYBBSX = ((MYBBS+NSUBD2)/NSUBDV)*NSUBDV
      IF (NSUBDV .EQ. 1) GO TO 110
      SUBOFF = .F.
      MYBBP1 = MYBB+1
      DO 105 I = 1,MYBBP1
      ENRULU(I) = (0.,0.)
      ENRLLL(I) = (0.,0.)
      ENRURW(I) = (0.,0.)
      ENRULW(I) = (0.,0.)
      FULLBX(I) = .T.
  105 CONTINUE
      GO TO 120
  110 SUBOFF = .T.
      CROW = .T.
      LROW = .T.
      FBOX = .T.
      CBOX = .T.
      LBOX = .T.
C
C          LOOP ON ALL (SUBDIVIDED) ROWS OF THE SURFACE
C
  120 CONTINUE
      FLIROW = FLOAT(IXBS) - 1.0
      DO 1300 IROW = IXBS,MXBS
      FLIROW = FLIROW + 1.0
C
C          SET FLAGS FOR FIRST, CENTER AND LAST SUBDIVIDED ROW IN UN-
C          SUBDIVIDED ROW
      IF (SUBOFF) GO TO 270
      FROW = .F.
      IF (IROW - IXB) 230,220,210
  210 IF (MOD(IROW-IXB, NSUBDV) .NE. 0) GO TO 240
  220 CROW = .T.
      GO TO 250
  230 CROW = .F.
      IF (IROW .EQ. IXBS .OR. IROW .EQ. IRFB) FROW = .T.
```

```
      GO TO 250
  240 CROW = .F.
      I = MOD(IROW+NSUBCN - IXB, NSUBDV)
      IF (I - 1) 260,245,250
  245 FROW = .T.
  250 LROW = .F.
      IF (IROW .NE. MXBS) GO TO 270
  260 LROW = .T.
C
C        GET THE SUBDIVIDED ROW NUMBER FOR THE CENTER OF THE ASSOCIATED
C        UNSUBDIVIDED BOX,  ICENT
  270 CONTINUE
      IF (CROW) GO TO 340
      IF (IROW - IRFB) 310,320,330
C        NO FULL ROW
  310 ICENT  = 0
      IUCENT = 0
      NCDCEN = 0
      CBOX   = .F.
      LBOX   = .F.
      GO TO 355
  320 ICENT = IXB
      GO TO 350
  330 ICENT =((IROW-IRFB)/NSUBDV)*NSUBDV + IXB
      IF (ICENT .GT. MXBS) ICENT = ICENT - NSUBDV
      GO TO 350
  340 ICENT = IROW
C
  350 CONTINUE
      IUCENT = (ICENT-IXBW)/NSUBDV + 1
  355 CONTINUE
      JEXLOC = 1
      IF (.NOT. WING) JEXLOC = MYBSW + 1
C
C        LOOP ON ALL (SUBDIVIDED) CHORDS FOR THE SURFACE AND DIAPHRAGM
C
      DO 1200 JCOL = 1,MYBBSX
C
C        GET THE CURRENT (SUBDIVIDED) BOX CODE
      CALL DCODER(IBOX,LBXCD, IROW,JCOL, IROW,JCOL, .T., NCDBOX)
      IF (SUBOFF) GO TO 480
C
C     - - - - - - - - - - - - - - - - - - - - - - -
C
C       GET INFORMATION ABOUT POSITION WITHIN UNSUBDIVIDED BOX
C
C         ICENT  = I-LOCATION (SUBDIVIDED) OF THE CONTROL POINT
C         JCENT  = J-LOCATION (SUBDIVIDED) OF THE CONTROL POINT
C         IUCENT = UNSUBDIVIDED I-LOCATION OF ASSOCIATED CONTROL POINT
C         JUCENT = UNSUBDIVIDED J-LOCATION OF ASSOCIATED CONTROL POINT
C         IPCENT = UNSUBDIVIDED I-LOCATION OF THE NEAREST PLANFORM
C                  CONTROL POINT, IF THE SUBDIVIDED BOX IS ON-PLANFORM
C         CBOX   = .T., CURRENT BOX IS A CENTER BOX
C         LBOX   = .T., THIS IS THE LAST BOX ASSOCIATED WITH THE CONTROL
C                    POINT
C         NCDCEN = CODE FOR CONTROL POINT
C         NCDBOX = CODE FOR THE CURRENT SUBDIVIDED BOX
```

```
C          IICENT = I-LOCATION (SUBDIVIDED) FOR THE NEAREST PLANFORM
C                    CONTROL POINT
C
      LBOX = .F.
      JUCENT = (JCOL-NSUBCN)/XSUBDV + 1.5
      JCENT  = NSUBDV * JUCENT - NSUBD2
      IF (.NOT. CROW) GO TO 410
      IF (JCENT .NE. JCOL) GO TO 410
      CBOX = .T.
      NCDCEN = NCDBOX
      IICENT = IROW
      IF (NCDBOX) 450,1100,450
C          GET CENTER BOX CODE, NCDCEN
  410 CBOX = .F.
      NCDCEN = 0
      IF (ICENT .LE. 0) GO TO 414
      CALL DCODER(IBOX,LBXCD, ICENT,JCENT, ICENT,JCENT, .T., NCDCEN)
      IICENT = ICENT
      GO TO 418
  414 IICENT = IXB - NSUBDV
C
  418 CONTINUE
      IF (NCDBOX - 1) 420,424,450
C          SUBDIVIDED BOX IS NOT CONSIDERED (CODE = 0).  IF CENTER CODE
C          IS ALSO ZERO, LOOP TO NEXT BOX.  OTHERWISE, CHECK FOR LAST BOX
  420 IF (NCDCEN) 450,1100,450
C
C          SUBDIVIDED BOX CODE = 1.  CHECK WHETHER ITS CONTROL PT = 1
  424 CONTINUE
      IF (.NOT. COPLAN) GO TO 431
      IF (TEXLOC(JEXLOC) .LT. FLIROW) JEXLOC = JEXLOC + MYBSW
      IF (FEXLOC(JEXLOC) .GT. FLIROW) JEXLOC = JEXLOC - MYBSW
C          DETERMINE WHETHER SUBDIVIDED BOX IS ON SAME PLANFORM AS
C          IICENT  (LOCATION OF NEAREST CONTROL POINT)
      IF (JEXLOC .EQ. JCOL) GO TO 428
C          SUBDIVIDED BOX IS ON THE TAIL.  IS IICENT ON THE WING -
      IF (FLOAT(IICENT) .LE. TEXLOC(JCENT) ) GO TO 432
C          NO.  CHECK FOR IICENT OFF-PLANFORM.
      GO TO 431
C          SUBDIVIDED BOX IS ON THE WING.  IS IICENT AFT OF THE WING T.E.
  428 IF (FLOAT(IICENT) .GT. TEXLOC(JCENT) ) GO TO 432
C          NO.  CHECK FOR IICENT OFF- PLANFORM (L.E. DIAPHRAGM)
  431 CONTINUE
C          CHECK CODE AT IICENT (NEAREST CONTROL POINT)
      IF (NCDCEN .EQ. 1) GO TO 450
C          SUBDIVIDED ON-PLANFORM BOX DOES NOT LIE WITHIN AN UNSUBDIVIDED
C          BOX WHOSE CONTROL POINT IS ON PLANFORM.  SEARCH FORE AND AFT
C          FOR THE NEAREST CONTROL POINT ON THE SURFACE.
  432 CONTINUE
      IFCOMR = IFIX(FEXLOC(JEXLOC)) + 1
      ILCOMR =        TEXLOC(JEXLOC)
      IMAX = 2*NSUBDV
      DO 436 I = NSUBDV,IMAX,NSUBDV
      IICENT = ICENT+I
      IF (IICENT .GT. ILCOMR) GO TO 434
      CALL DCODER(IBOX,LBXCD, IICENT,JCENT, IICENT,JCENT, .T., NCD)
      IF (NCD .EQ. 1) GO TO 440
```

```
  434 CONTINUE
      IICENT = ICENT-1
      IF (IICENT .LT. IFCOMR) GO TO 438
      CALL DCODER(IBOX,LBXCD, IICENT,JCENT, IICENT,JCENT, .T., NCD)
      IF (NCD .EQ. 1) GO TO 440
  438 CONTINUE
C          NO CENTER BOX FOUND.  A WARNING DIAGNOSTIC WILL BE PRINTED,
C          THEN COMPUTATION WILL CONTINUE AT 420
      GO TO 3010
C          A BOX WAS FOUND
  440 CONTINUE
C
C          THE ASSOCIATED CONTROL POINT HAS BEEN FOUND.  GET THE UNSUB-
C          DIVIDED SUBSCRIPT.
  450 CONTINUE
      IPCENT = (IICENT-IXBW)/NSUBDV + 1
      IF (LROW) GO TO 470
      IF (.NOT. FROW) GO TO 460
      IF (JCOL .NE. JCENT-NSUBD2) GO TO 460
      FBOX = .T.
      GO TO 500
  460 CONTINUE
      FBOX = .F.
      GO TO 500
  470 CONTINUE
      IF (JCOL .EQ. JCENT+NSUBD2) LBOX = .T.
      FBOX = .F.
      GO TO 500
C
C     - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C          SET UP VALUES FOR AN UNSUBDIVIDED CASE
C          TEST FOR NON-ZERO BOX CODE -
  420 CONTINUE
      IF (NCDBOX .EQ. 0) GO TO 1100
      ICENT = IROW
      IICENT = ICENT
      IUCENT = ICENT
      JCENT  = JCOL
      JUCENT = JCOL
      CBOX   = .T.
      LBOX   = .T.
      NCDCEN = NCDBOX
      FBOX   = .T.
      IPCENT = ICENT
C
  500 CONTINUE
      IF (NCDBOX .GT. 0) GO TO 510
      FULLBX(JUCENT) = .F.
      IF (LBOX) GO TO 1040
      IF (FBOX) GO TO 515
      GO TO 1100
C
C          THE BOX IS TO BE CONSIDERED.  ARE N-HAT TERMS NECESSARY -
  510 CONTINUE
C          ARE N-HAT TERMS ALREADY AVAILABLE -
      IF (FBOX) GO TO 515
```

```
      IF (NCDBOX .NE. 1) GO TO 860
      IF (.NOT. WING) GO TO 810
      IF (IPCENT .NE. IUCENT) GO TO 830
      GO TO 850
C
C     - - - - - - - - - - - - - - - - - - - - - - - -
C
C         COMPUTE N-HAT TERMS FOR THIS (UNSUBDIVIDED) BOX
C
  515 CONTINUE
      ENRULU(JUCENT) = (0.,0.)
      ENRLLL(JUCENT) = (0.,0.)
      DPHIL(JUCENT) = (0.,0.)
C         ARE LEFT SURFACE CONTRIBUTIONS POSSIBLE -
      IF (IUCENT-IXBU .LT. JUCENT) GO TO 600
      IF (PSIS .EQ. 0 .OR. .NOT. DIHS) GO TO 600
C        GET AIC ARRAYS W AND V FOR LEFT SURFACE INFLUENCE ON RT SURFACE
      CALL GETAIC(JUCENT,WW, 0, IR)
      IF (IR .NE. 0) GO TO 600
      NUBMIN = JUCENT
      NUBMAX = IUCENT - IXBU
      I = IUCENT - JUCENT
      YMUBAR = COS(2*PSIS)* (JUCENT-.5)
      JBAR   = YMUBAR + 1
C         GET REFERENCE LOCATION IN AIC ARRAYS
      IF (YBAR) 520,525,530
  520 JINCR = 1
      GO TO 535
  525 IAIC = NUBMIN**2
      INCAIC = 2*NUBMIN + 1
      JINCR = -1
      GO TO 540
  530 JINCR = -1
  535 IAIC = NUBMIN**2 + NUBMIN
      INCAIC = 2*NUBMIN + 2
C
C         LOOP FORWARD OF BOX FOR WING/WING (TAIL/TAIL) N-HAT TERMS
  540 CONTINUE
      DO 590 NUBAR = NUBMIN,NUBMAX
      MUAIC1 = MUAIC(1,NUBAR+1)
      MUAIC2 = MUAIC(2,NUBAR+1)
      IF (MUAIC2 .EQ. 0) GO TO 585
      IF (YBAR .GE. 0) GO TO 550
      JCOLL = -JBAR - NUBAR + MUAIC1
      GO TO 560
  550 JCOLL = -JBAR + NUBAR - MUAIC1 + 2
  560 CONTINUE
C
C         LOOP LEFT OF RECEIVING CHORD TO GET LEFT SURFACE CONTRIBUTIONS
      DO 580 MUAI = MUAIC1,MUAIC2
      IF (JCOLL .LE. 0) GO TO 570
      CALL DCODER(IBOX,LBXCD, I,JCOLL, I,JCOLL, .F., ICD)
      IF (ICD .EQ. 0) GO TO 570
C         A CONTRIBUTING BOX HAS BEEN FOUND.  GET THE AIC LOCATION
      KAIC = IAIC + MUAI
C         GET LOCATION IN N  ARRAYS FOR THE VALUES AT BOX (I,JCOLL)
      ICS = LOCSDW(I,JCOLL, IPNTDW,LPNTDW, 1, LPNTDW)
```

```
C                                                                        VELPOT  00373
      COEF =  ( CPSIS2*W(KAIC) - SPSIS2*V(KAIC) ) * SYMTY                VELPOT  00374
      ENRULU(JUCENT) = ENRUS(IDS)*COEF + ENRULU(JUCENT)                  VELPOT  00375
      ENRLLL(JUCENT) =-ENRLS(IDS)*COEF + ENRLLL(JUCENT)                  VELPOT  00376
      DPHIL(JUCENT) = (ENRUS(IDS) - ENRLS(IDS) ) * C(KAIC) * SYMTY +     VELPOT  00377
     1                DPHIL(JUCENT)                                      VELPOT  00378
  570 CONTINUE                                                           VELPOT  00379
C                                                                        VELPOT  00380
      JCOLL = JCOLL + JINCR                                              VELPOT  00381
  580 CONTINUE                                                           VELPOT  00382
C         END OF LOOP FOR LEFT ROW CONTRIBUTIONS                         VELPOT  00383
C                                                                        VELPOT  00384
  585 CONTINUE                                                           VELPOT  00385
      I = I - 1                                                          VELPOT  00386
      IF (I .LT. IXBU) GO TO 600                                         VELPOT  00387
      IAIC = INCAIC + IAIC                                               VELPOT  00388
      INCAIC = INCAIC + 2                                                VELPOT  00389
  590 CONTINUE                                                           VELPOT  00390
C         END OF LOOP FORWARD ON ROWS, TO COMPUTE LEFT SURFACE OUT-OF-   VELPOT  00391
C         PLANE EFFECTS, FROM 540                                        VELPOT  00392
C                                                                        VELPOT  00393
C         IF THIS IS AN ON-PLANFORM TAIL BOX, THERE ARE WING-TAIL        VELPOT  00394
C         CONTRIBUTIONS                                                  VELPOT  00395
  600 CONTINUE                                                           VELPOT  00396
      IF (WING) GO TO 830                                                VELPOT  00397
      ENRURW(JUCENT) = (0.,0.)                                           VELPOT  00398
      ENRULW(JUCENT) = (0.,0.)                                           VELPOT  00399
      IF (NCUCEN .NE. 1) GO TO 860                                       VELPOT  00400
C                                                                        VELPOT  00401
C         COMPUTE THE RIGHT WING CONTRIBUTION TO THE TAIL BOX            VELPOT  00402
      II = 1                                                             VELPOT  00403
      IF (PSIW .EQ. PSIT) II = 2                                         VELPOT  00404
      CALL GETAIC(JUCENT,RWT, II, IR)                                    VELPOT  00405
      IF (IR .NE. 0) GO TO 700                                           VELPOT  00406
      NUBMIN =  ABS(EL) + .5                                             VELPOT  00407
      NUBMAX = IUCENT - 1                                                VELPOT  00408
      I = IUCENT - NUBMIN                                                VELPOT  00409
      YMUBAR = (JUCENT - .5) * COS(PSIDIF) + CAPL * SIN(PSIW)            VELPOT  00410
      JBAR  = YMUBAR                                                     VELPOT  00411
      IF (YMUBAR .GE. 0) JBAR = JBAR + 1                                 VELPOT  00412
      IF (YBAR) 620,625,630                                              VELPOT  00413
  620 JINCR = -1                                                         VELPOT  00414
      GO TO 635                                                          VELPOT  00415
  625 IAIC = NUBMIN**2                                                   VELPOT  00416
      INCAIC = 2*NUBMIN + 1                                              VELPOT  00417
      JYNCR = 1                                                          VELPOT  00418
      GO TO 640                                                          VELPOT  00419
  630 JINCR = 1                                                          VELPOT  00420
  635 IAIC = NUBMIN**2 + NUBMIN                                          VELPOT  00421
      INCAIC = 2*NUBMIN + 2                                              VELPOT  00422
  640 CONTINUE                                                           VELPOT  00423
C                                                                        VELPOT  00424
C         LOOP FORWARD OVER THE RIGHT WING                               VELPOT  00425
      DO 690 NUBAR = NUBMIN,NUBMAX                                       VELPOT  00426
      MUAIC1 = MUAIC(1,NUBAR+1)                                          VELPOT  00427
      MUAIC2 = MUAIC(2,NUBAR+1)                                          VELPOT  00428
      IF (MUAIC2 .EQ. 0) GO TO 685                                       VELPOT  00429
```

```
      IF (YBAR .GE. 0) GO TO 650
      JCOLR = JBAR + NUBAR - MUAIC2 + 1
      GO TO 660
  650 JCOLR =JBAR - NUBAR + MUAIC1 - 1
  660 CONTINUE
C
C        LOOP ON A ROW OF WING BOXES, COMPUTING RIGHT HAND WING-TAIL
C        CONTRIBUTIONS
      DO 680 MUAI = MUAIC1,MUAIC2
      IF (JCOLR .LE. 0) GO TO 670
      CALL DCODER(IBOXW,LBXCDW, I,JCOLR, I,JCOLR, .F., ICD)
      IF (ICD .EQ. 0)    GO TO 670
C        A CONTRIBUTING BOX HAS BEEN FOUND. GET THE AIC LOCATION
      KAIC = IAIC + MUAI
C        GET THE NORMAL-WASH LOCATION
      IDS  = LOCSDW( I, JCOLR,  IPNTDW,LPNTDW, 1, LPNTDW)
C        ADD THIS CONTRIBUTION TO  N-HAT
      IF (II .EQ. 2) GO TO 665
      ENRURW(JUCENT) = (COS(PSIDIF)*W(KAIC) - SIN(PSIDIF)*V(KAIC))
     1                * ENRUS(IDS)  +  ENRURW(JUCENT)
      GO TO 670
  665 CONTINUE
      ENRURW(JUCENT) =   COS(PSIDIF)*W(KAIC) * ENRUS(IDS) +
     1                   ENRURW(JUCENT)
  670 CONTINUE
      JCOLR = JCOLR + JINCR
  680 CONTINUE
C        END OF LOOP FOR RIGHT WING ROW CONTRIBUTIONS
C
  685 CONTINUE
      I = I - 1
      IF (I .LE. 0) GO TO 700
      IAIC = IAIC + INCAIC
      INCAIC = INCAIC + 2
  690 CONTINUE
C        END OF LOOP FORWARD ON ROWS, TO COMPUTE RIGHT WING OUT-OF-
C        PLANE EFFECTS ON THE TAIL,  FROM 640
C
C        DETERMINE WHETHER LEFT WING INFLUENCE IS TO BE COMPUTED
  700 CONTINUE
      IF (SYM .EQ. 0) GO TO 800
C        GET AIC ARRAYS W AND V FOR LEFT WING INFLUENCE ON TAIL
      II = 1
      IF ( -PSIW .EQ. PSIT) II = 2
      CALL GETAIC(JUCENT, LWT, II, IR)
      IF (IR .NE. 0) GO TO 800
      NUBMIN = ABS(EL) + .5
      NUBMAX = IUCENT - 1
      I = IUCENT - NUBMIN
      YMUBAR = - COS(PSIW + PSIT)*(JUCENT-.5)  +  ZAPL*SIN(PSIW)
      JBAR = YMUBAR
      IF (YMUBAR .GE. 0) JBAR = JBAR + 1
      IF (YBAR) 720,725,730
  720 JINCR = 1
      GO TO 735
  725 IAIC = NUBMIN**2
      INCAIC = 2*NUBMIN + 1
```

```
      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 MUAI = MUAIC1,MUAIC2                                       VELPOT 00507
      IF (JCOLL .LE. 0) GO TO 770                                       VELPOT 00508
      CALL DCODER(IBOXW,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 + MUAI                                                VELPOT 00512
C          GET THE NORMAL WASH LOCATION                                 VELPOT 00513
      IDS = LOCSDW( I, JCOLL, IPNTDW, LPNTDW, 1, LPNTDW)                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 .LT. 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          PLANFORM.  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
  810 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+IOVLP, 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, XKVL*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+IOVLPN, 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
  830 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, XKVL*DFL) + ENRLLL(JUCENT)) * 2.0  VELPOT  00573
C                                                                       VELPOT  00574
      IF (IPCENT .NE. IUCENT) 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-PLANFORM            VELPOT  00580
  850 CONTINUE                                                          VELPOT  00581
      IF (.N. FBOX) LOCNW = LOCSDW(IUCENT+IOVLPN, 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 PLANFORM   VELPOT  00590
C         BOX WITH NO ASSOCIATED PLANFORM CONTROL POINT                 VELPOT  00591
  832 CONTINUE                                                          VELPOT  00592
      FULLBX(JUCENT) = .FALSE.                                          VELPOT  00593
      DELPHA = (ENSUM + ENDIF) * 0.5                                    VELPOT  00594
      DELPHB = (ENSUM - ENDIF) * 0.5                                    VELPOT  00595
  854 CONTINUE                                                          VELPOT  00596
      ENSRUS = CMPLX( REAL(DELPHA)/XSUBDV,  AIMAG(DELPHA)/XSUBDV +       VELPOT  00597
     1              XKVL*(IROW-IICENT)*REAL(DELPHA) )                   VELPOT  00598
      ENSRLS = CMPLX( REAL(DELPHB)/XSUBDV,  AIMAG(DELPHB)/XSUBDV +       VELPOT  00599
     1              XKVL*(IROW-IICENT)*REAL(DELPHB) )                   VELPOT  00600
```

```
      IF (CBOX) GO TO 870
      GO TO 1030
  855 CONTINUE
      ENSRUS = ENRUS(LOCNW)
      ENSRLS = ENRLS(LOCNW)
      GO TO 870
  860 CONTINUE
      FULLBX(JUCENT) = .F.
C
C     - - - - - - - - - - - - - - - - - - - - - - - - -
C
C          DETERMINE THE VELOCITY POTENTIAL CONTRIBUTIONS FROM
C          BOXES LYING  A H E A D  OF THE CURRENT BOX
C
  870 CONTINUE
      IF (PSIS .NE. 0 .AND. DIHS) GO TO 880
C          THE SURFACE IS PLANAR.  GET FULL SURFACE CONTRIBUTIONS
      DELPH = B(IROW,JCOL, PKERNL,SKERNL, IBOX,LBXCD, WING, .F. )
      GO TO 890
C
C          DIHEDRAL ANGLE IS TO BE ACCOUNTED FOR.  GET THE PLANAR
C          (SUBDIVIDED) CONTRIBUTION OF THE RIGHT SURFACE
  880 CONTINUE
      DELPH = B(IROW,JCOL, PKERNL,SKERNL, IBOX,LBXCD, WING, .T. )
C          ADD THE SPATIAL LEFT SURFACE CONTRIBUTIONS
      DELPH = DELPH + DPHIL(JUCENT)
  890 CONTINUE
      IF (NCDBOX - 2) 910,1000,980
C
C     - - - - - - - - - - - - - - - - - - - - - - - - -
C
C          THE BOX IS ON-PLANFORM, CENTER.  COMPLETE THE CALCULATION OF
C          THE VELOCITY POTENTIAL
C
  910 CONTINUE
      IDS  = LOCSTW(IUCENT+IOVLP,  JUCENT, IPNTRM,LMODES, 1, LMODES)
      IF (SUBOFF) GO TO 915
      DELPHI(IDS)  = (ENSRUS-ENSRLS) * SKERNL(1) + DELPH
      GO TO 920
  915 CONTINUE
      DELPHI(IDS)  = (ENSRUS-ENSRLS) * PKERNL(1) + DELPH
  920 CONTINUE
C
C          COMPUTE ANY TRAILING EDGE VELOCITY POTENTIALS ASSOCIATED
C          WITH THIS UNSUBDIVIDED BOX
C
C          IS THIS A TRAILING EDGE BOX -
      IF (.NOT. COPLAN) GO TO 930
      IF (TEXLOC(JEXLOC) .LT. FLIROW) JEXLOC = JEXLOC + MYBSW
      IF (FEXLOC(JEXLOC) .GT. FLIROW) JEXLOC = JEXLOC - MYBSW
  930 CONTINUE
      JJ = JEXLOC - NSUBD2
      TEXMIN = TEXLOC(JJ)
      IF (NSUBDV .EQ. 1) GO TO 935
      DO 932 J = 2,NSUBDV
      JJ = JJ + 1
      TEXMIN = AMIN1(TEXMIN,TEXLOC(JJ))
```

```
  932 CONTINUE
  933 IF ( TEXMIN .GE. FLIROW + XSUBDV) GO TO 1030
C         YES.  GET THE BASIC VELOCITY POTENTIAL
      DELPHB = DELPHI(IDS)
C         TEST UNSUBDIVIDED BOX AHEAD OF CURRENT ONE -
      CALL DCODER(IBOX,LBXCD, IROW-NSUBDV,JCOL, IROW-NSUBDV,JCOL,
     1            .T.,  NCDA)
      IF (NCDA .NE. 1) GO TO 940
C         IT IS ON PLANFORM.  TEST FOR TIP CHORD -
      IF ( JEXLOC .EQ. JCOL .AND. JUCENT .LT. MYBW) GO TO 950
      IF ( JEXLOC .GT. JCOL .AND. JUCENT .LT. MYBT) GO TO 950
C         BOX IS ON THE TIP CHORD.  CHECK FOR THIRD TIP BOX -
      IR = IROW - 2*NSUBDV
      IF ( IR  .LT. IXB) GO TO 940
      CALL DCODER(IBOX,LBXCD, IR, JCOL,  IR,JCOL, .T., NCDC)
      IF (NCDC .EQ. 1) GO TO 950
C         TRY MACH RAY EXTRAPOLATION.  ARE THE 2 RAY BOXES ON-PLANFORM -
  940 CONTINUE
      JC = JCOL - NSUBDV
      CALL DCODER(IBOX,LBXCD, IROW,JC, IROW,JC, .T., NCDD)
      IF (NCDD .NE. 1) GO TO 945
      JC = JC - NSUBDV
      CALL DCODER(IBOX,LBXCD, IROW-NSUBDV,JC, IROW-NSUBDV,JC, .T.,NCDD)
      IF (NCDD .NE. 1) GO TO 945
C
C         MACH RAY EXTRAPOLATION, FOLLOWED BY CHORDWISE LINEAR EXTRA-
C              POLATION
      IDPHM2 = LOCSDW( IUCENT-1+IOVLP,JUCENT-2, IPNTRM,LMODES,1,LMODES)
      IDPHM1 = LOCSDW( IUCENT+IOVLP,  JUCENT-1, IPNTRM,LMODES,1,LMODES)
      DELPHA = DELPHI(IDPHM2)
      DELPHC = DELPHI(IDPHM1)
      SDELPH = (2.0*DELPHC - DELPHA - DELPHB)/XSUBDV
      GO TO 955
C
C         MACH RAY UNAVAILABLE.  ARE THERE 2 BOXES ON THE CHORD -
  945 IF (NCDA .NE. 1) GO TO 3020
C
C         CHORDWISE LINEAR EXTRAPOLATION
C
  950 CONTINUE
      IDPHM1 = LOCSDW(IUCENT-1+IOVLP, JUCENT, IPNTRM,LMODES, 1,LMODES)
      SDELPH = (DELPHB - DELPHI(IDPHM1))/XSUBDV
C
C         LOOP TO COMPUTE AND STORE TRAILING EDGE VELOCITY POTENTIALS
  955 CONTINUE
      JA = JEXLOC - NSUBD2
      JB = JEXLOC + NSUBD2
      DO 960 JJ = JA,JB
      XINCR = TEXLOC(JJ) - IROW
      IF (XINCR .LT. -XSUBDV/2.0) GO TO 960
      TVP(JJ) = DELPHB + XINCR*SDELPH
  960 CONTINUE
C         ALL TRAILING EDGE VALUES HAVE BEEN COMPUTED FOR THIS
C         UNSUBDIVIDED BOX.
      GO TO 1030
C
C     - - - - - - - -   - - - - - - - - - - - - - - -
```

```
C
C        BOX IS IN THE TRAILING EDGE DIAPHRAGM AREA.  COMPUTE THE WAKE
C        VELOCITY POTENTIAL CONTRIBUTION
  980 CONTINUE
      IF (COPLAN) GO TO 985
      JJ = JEXLOC
      GO TO 990
  985 JJ = JCOL
      IF (JJ .GT. MYBST) GO TO 990
      JJT = JJ + MYBSW
      IF (FLOAT(IROW) .GT. TEXLOC(JJT) ) JJ = JJT
  990 CONTINUE
C        COMPUTE (X-DISTANCE/B1) * K1
      XDKVL = (FLOAT(IROW)-TEXLOC(JJ) ) * XKVL/XSUBDV
      AZERO =  TVP(JJ) * CMPLX( COS(XDKVL), -SIN(XDKVL) )
      IF (.NOT. CBOX) GO TO 1010
      IF (IUCENT+IOVLP .GT. NPNTRS) GO TO 1010
C        SET DELPHI VALUE TO ZERO, TO CLEAN UP LEFT OVER MODE SHAPES
      I = LOCSDW(IUCENT+IOVLP,JUCENT, IPNTRM, LMODES,1,LMODES)
      IF (I .NE. 0) DELPHI(I) = (0.,0.)
      GO TO 1010
C
C        BOX IS IN A LEADING EDGE OR TIP DIAPHRAGM AREA
 1000 CONTINUE
      AZERO = (0.,0.)
C
C        COMPUTE NORMAL-WASH VALUES FOR A (SUBDIVIDED) DIAPHRAGM BOX
C
 1010 CONTINUE
      IF (SUBOFF) GO TO 1015
      ENDIF = (AZERO - DELPH)/ SKERNL(1)
      ENSUM = -(ENRULU(JUCENT) + ENRLLL(JUCENT) )
      ENSUM =  CMPLX(REAL(ENSUM), AIMAG(ENSUM) + XKVL*XSUBDV*(IROW -
     1             ICENT)*REAL(ENSUM) ) / XSUBDV
      GO TO 1020
 1015 CONTINUE
      ENDIF = (AZERO - DELPH)/ PKERNL(1)
      ENSUM = -(ENRULU(JUCENT) + ENRLLL(JUCENT) )
C
 1020 CONTINUE
      ENSRUS =  0.5*(ENSUM + ENDIF)
      ENSRLS =  0.5*(ENSUM - ENDIF)
      IF (SUBOFF) GO TO 1090
      GO TO 1035
C
C     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C        STORE THE NORMAL WASH VALUES
C
 1030 CONTINUE
      IF (SUBOFF) GO TO 1100
C
C        STORE THE COMPUTED SUBDIVIDED NORMAL WASHES
 1035 CONTINUE
      CALL STOSDW (IROW,JCOL, ENSBD, IBOX,LBXCD, IXBS,MXBS,MYBBS,IRR)
      IF (IRR .NE. 0) GO TO 3030
      IF (.NOT. LBOX) GO TO 1100
```

```
C
C         DETERMINE THE UNSUBDIVIDED NORMAL WASH VALUES
 1040 IF (NCDCEN - 1) 1100,1050,1060
C         IF THE BOX IS COMPLETE, THE VALUE IS ALREADY STORED -
 1050 IF (FULLBX(JUCENT)) GO TO 1100
C         THE UNSUBDIVIDED VALUE EQUALS THE AVERAGE OF ALL ITS
C         SUBDIVIDED BOXES
 1060 CONTINUE
      II = ICENT - NSUBD2
      JJJ = JCENT + NSUBD2
      ENSRUS = (0.,0.)
      ENSRLS = (0.,0.)
      DO 1080 I = 1,NSUBDV
      JJ  = JCENT - NSUBD2
      CALL DCODER(IBOX,LBXCD, II,JJ, II,JJJ, .T., ICODE )
      DO 1075 J = 1,NSUBDV
      IF (ICODE(J) .EQ. 0) GO TO 1070
      LOCDW = LOCSDW( II,JJ, IPNTSD,IPNTIN,IPNTOT,IPNTLS)
      ENSRUS = ENSUBD(1,LOCDW) + ENSRUS
      ENSRLS = ENSUBD(2,LOCDW) + ENSRLS
 1070 CONTINUE
      JJ = JJ + 1
 1075 CONTINUE
      II = II + 1
 1080 CONTINUE
      ENSRUS = ENSRUS/XSUBDV
      ENSRLS = ENSRLS/XSUBDV
C
C         RESTORE THE PARTIAL BOX FLAG FOR THE NEXT ROW
      FULLBX(JUCENT) = .T.
C
C         STORE THE UNSUBDIVIDED NORMAL WASHES
 1090 CONTINUE
      LOCNW = LOCSDW( IUCENT+IOVLPN, JUCENT, IPNTDW, LPNTDW,1,LPNTDW )
      IF (LOCNW .EQ. 0 ) GO TO 3040
      ENRUS(LOCNW) = ENSRUS
      IF (WING) GO TO 1095
      ENRLS(LOCNW       ) = ENSRLS
      GO TO 1100
 1095 ENRLS(LOCNW) = ENSRLS
 1100 CONTINUE
      JEXLOC = JEXLOC + 1
C
C
 1200 CONTINUE
C         END OF LOOP ON (SUBDIVIDED) CHORDS,  STARTING AT 355*
C
C
 1300 CONTINUE
C         END OF LOOP ON (SUBDIVIDED) ROWS,   STARTING AT  320
C
C
      RETURN
C
C     - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C         DIAGNOSTICS - ALL CALL FLUSH
```

```
C
 3010 WRITE (NT6,9010)
      WRITE (NT6,9999) IROW,JCOL, IUCENT,JUCENT
      GO TO 420
 3020 WRITE (NT6,9020)
      GO TO 3999
 3030 WRITE (NT6,9030)
      GO TO 3999
 3040 WRITE (NT6,9040)
      GO TO 3999
C
 3999 WRITE (NT6,9999)  IROW,JCOL, IUCENT,JUCENT
      CALL FLUSH(1)
C
C
 9010 FORMAT(56H0*** WARNING - NO PLANFORM CONTROL POINT FOUND FOR SUBDI
     1 52HVIDED BOX DURING VELOCITY POTENTIAL CALCULATIONS *** )
 9020 FORMAT(56H0*** ERROR - THE TIP BOX PATTERN DOES NOT ALLOW TRAILING
     1 44H EDGE VELOCITY POTENTIALS TO BE COMPUTED *** )
 9030 FORMAT(56H0*** ERROR - FAILURE IN STORING SUBDIVIDED NORMAL-WASHES
     1 4H *** )
 9040 FORMAT(53H0*** ERROR - FAILURE IN STORING CONTROL POINT NORMAL-
     1    10HWASHES *** )
 9999 FORMAT(14X,16HSUBDIVIDED BOX (,I3,1H,I3,19H),  CONTROL POINT (,
     1     I2,1H,I2,1H)  )
C
      END
```

```
      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 DIHDRL   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            MXB    = 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            LROT   = 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             LIMDWS = DIMENSION OF SUBDIVIDED NORMAL-WASH ARRAYS        B       00041
C                                                                        B       00042
      COMMON /GEOMTY/ COPLAN,NSUBDV,XSUBDV,NSUPDC,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,YLAZ,PSIT,MXBT,MYBT,MYBBT,MXBST,MYBST,        GEOM2   00002
     1                MYBBST,IXBT,IXBST,CAPL                             GEOM2   00003
      COMMON / MODES/ SYM,SYMT,MTYPEW,MTYPET                             MODCOM  00002
      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 /NWASHES/ IPNTDW(2,100),ENRUS(1275), ENRLS(1275),IOVLAPN    NWASHES 00002
      COMPLEX                       ENRUS,        ENRLS                  NWASHES 00003
      COMMON / KERN / ERR,MXSKRN,IPKERN,NPLKRN,NSPATK,NROWEA             KERN    00002
C              DELPHI(LMODES),TYP(LTYP),TEXLOC(LTYP)                     DELTAP  00002
      COMMON /DELTAP/ DELPHI(1080), TYP(250), TEXLOC(250), FEXLOC(250),  DELTAP  00003
```

```
     1               IPNTRW(2,100),NPNTRS, IOVLAP                    DELTAP  00004
      COMPLEX        DELPHI,         TVP                             DELTAP  00005
C                 DEFSL(2,LMODES)                                    DELTAP  00006
      DIMENSION   DEFSL(2,1000)                                      DELTAP  00007
      EQUIVALENCE    (DELPHI(61), DEFSL)                             DELTAP  00008
C               ARRAYS DELPHI AND DEFSL ARE                          DELTAP  00009
C               EQUIVALENCED TO GIVE A 2 ROW UN-OVERLAPPED SECTION   DELTAP  00010
      COMMON /LROT  / LROT                                           B       00050
C                                                                    B       00051
C                                                                    B       00052
      COMPLEX PKERNL(1),SKERNL(1)                                    B       00053
      LOGICAL WING,DIH                                               B       00054
C                                                                    B       00055
      DIMENSION IBXCD(150)                                           B       00056
      LOGICAL LEFT, LSIDE                                            B       00057
C                                                                    B       00058
      IA = IROW                                                      B       00059
      B = (0.,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
      IOVLPN = IOVLAPN                                               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 = MYBBSW                                                 B       00076
      IOVLPN = 0                                                     B       00077
      SYMTY = SYM                                                    B       00078
   25 CONTINUE                                                       B       00079
      LSIDE = SYMTY .NE. 0 .AND. .NOT. DIH                           B       00080
C                                                                    B       00081
C          IS SUBDIVISION REQUESTED -                                B       00082
      IF (NSUBDV .EQ. 1) GO TO 410                                   B       00083
C                                                                    B       00084
C          YES.  DETERMINE THE NUMBER OF ROWS WHICH CAN BE HANDLED (MXB1) B   00085
      NSRWM1 = IPNTIN - IPNTOT - 1                                   B       00086
      IF (NSRWM1 .LT. 0) NSRWM1 = NSRWM1 + IPNTLS                    B       00087
      NSRWM1 = MIN0(NSRWM1,MXSKRN-1)                                 B       00088
      IF (IA - IXBS .GT. NSRWM1) 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 - NSRWM1 - IXB                                          B       00095
      I = MOD(I,NSUBDV)                                              B       00096
      I = LROT -I                                                    B       00097
      MXB1 = NSRWM1 - MOD(I,NSUBDV)                                  B       00098
C                                                                    B       00099
```

```
C        DETERMINE THE CONTRIBUTION TO B FROM A FORWARD CONE OF SUB-
C        DIVIDED BOXES
C            NU = ROW NUMBER OF CONTRIBUTING BOX RELATIVE TO RECEIVING
C                BOX.  NU OF RECEIVING BOX = 0.
C            IA = ACTUAL ROW LOCATION OF CONTRIBUTING BOX, RELATIVE TO
C                SUBDIVIDED GRID.
C
  200 CONTINUE
      IF (MXB1 .EQ. 0) GO TO 310
      DO 300 NU = 1,MXB1
      IA = IA - 1
C        HAS THE FORWARD EDGE OF THE PATTERN BEEN REACHED -
      IF (IA .LT. IXBS) GO TO 600
C        NO. GET BOX TYPE CODES FOR CURRENT ROW.
      LEFT = .T.
      IIA = MOD(IA-1,IPNTLS) + 1
      IIAP1 = MOD(IA,IPNTLS) + 1
      IPNTP1 = IPNTSD(1,IIAP1)
      IF (IPNTP1 .LE. 1) GO TO 208
      NYB = IPNTP1         - IPNTSD(1,IIA) + IPNTSD(2,IIA) - 1
      GO TO 210
  208 NYB = MYBBS
  210 CONTINUE
      CALL DCODER (IBOX,LBXCD, IA,1, IA,NYB, .T., IBXCD )
      IF (NYB .    MYBBS) GO TO 215
      DO 212 I = 1,MYBBS
      IF (IBXCD(NYB) .NE. 0) GO TO 215
      NYB = NYB - 1
  212 CONTINUE
  215 CONTINUE
C            IBXCD  = ROW OF BOX CODES
C            NYB    = NUMBER FOUND
C
C        GET LOCATION IN THE SUBDIVIDED DOWNWASH ARRAY FOR BOX(IA,JCOL)
      IDW = LOCSDW(IA,JCOL, IPNTSD,IPNTIN,IPNTOT,IPNTLS)
      N = (NU*(NU+1))/2 + 1
C        KERNEL(NU,MU) = SKERNL((NU*(NU+1))/2 + ABS(MU) + 1) , SO
C            N      = SUBSCRIPT FOR KERNEL (STARTING WITH NU,0 )
C
C        CENTER BOX OF ROW IN CONE
      IF (JCOL .GT. NYB) GO TO 220
      IF (IBXCD(JCOL) .EQ. 0) GO TO 220
      B = B + SKERNL(N) * (ENSUBD(1,IDW)- ENSUBD(2,IDW) )
  220 CONTINUE
C
C
C        GOING OUT FROM CENTER CHORD OF CONE IN BOTH DIRECTIONS
C          IDWR,IDWL = POINTERS IN DOWNWASH ARRAY FOR RIGHT,LEFT SIDES
C          IBXR,IBXL = POINTERS IN BOX CODES ARRAY, AS ABOVE
C            L       = LEFT SIDE POINTER INCREMENTER (CHANGES SIGN WHEN
C                      THE PLANFORM CENTER-LINE IS ENCOUNTERED)
C            E       = LEFT SIDE MULTIPLIER, USED TO DETERMINE SYM/ANTI
C                      SYM. AFTER PLANFORM CENTER-LINE ENCOUNTERED.
      IDWR = IDW+1
      IDWL = IDW-1
      IBXR = JCOL+1
      IBXL = IBXR-2
```

```
      N = N + 1
      L = -1
      E = 1.0
C
      DO 280 MU = 1,NU
C
C         RIGHT SIDE
      IF (IBXR .GT. NYB) GO TO 230
      IF (IBXCD(IBXR) .EQ. 0) GO TO 230
      B = B + SKERNL(N) * (ENSUBD(1,IDWR) - ENSUBD(2,IDWR) )
  230 CONTINUE
C
C         LEFT SIDE
C
      IF (.NOT. LEFT) GO TO 270
C         HAS PLANFORM CENTER LINE BEEN ENCOUNTERED -
      IF (IBXL .GT. 0) GO TO 250
C         YES.  SET PARAMETERS TO SWEEP BACK ACROSS RIGHT HALF AS A
C         SYMMETRIC/ANTISYMMETRIC IMAGE OF THE LEFT SIDE.
      LEFT = LSIDE
      IF (.NOT. LEFT) GO TO 270
      E  = SYMTY
      L = 1
      IBXL = 1
      IDWL = IDWL + 1
      GO TO 260
C         IF THE CENTER LINE HAS PREVIOUSLY BEEN ENCOUNTERED, IBXL WILL
C         BE INCREASING. IF IBXL HAS EXCEEDED THE NUMBER OF BOXES ON
C         THIS ROW, THIS ROW IS COMPLETE, TRANSFER TO LOOP ON NU.
  250 IF (IBXL .GT. NYB) GO TO 270
  260 IF (IBXCD(IBXL) .EQ. 0) GO TO 270
      B = B + SKERNL(N) * (ENSUBD(1,IDWL) - ENSUBD(2,IDWL) ) * E
  270 CONTINUE
C
C         SET COUNTERS FOR NEXT STEP OUTWARD
      IDWR = IDWR+1
      IDWL = IDWL+L
      IBXR = IBXR+1
      IBXL = IBXL+L
      N = N+1
  280 CONTINUE
C         END OF LOOP ON MU (SUBDIVIDED COLUMNS OUTBOARD)
C
  300 CONTINUE
C         END OF LOOP ON NU (SUBDIVIDED ROWS FORWARD)  FROM 200
C
C         IS THERE AT LEAST ONE FULL UNSUBDIVIDED ROW LEFT AHFAD OF
C         CURRENT POSITION -
  310 CONTINUE
      IF (IA .LT. IXB   ) GO TO 800
C
C
C         UNSUBDIVIDED BOXES
C
C         DETERMINE ROW AND COLUMN NUMBERS IN SUBDIVIDED ARRAYS CORRES-
C         PONDING TO UNSUBDIVIDED BOX CENTERS.
C              IA      = ROW LOCATION OF CONTRIBUT.. SUBDIVIDED BOX
```

```
C              IIA    = ROW LOCATION OF UNSUBDIVIDED BOX
C              INU     = FIRST ROW OF UNSUBDIVIDED BOXES TO USE, COUNTING
C                          OUTWARD.
C              JJJ     = UNSUBDIVIDED CHORD NUMBER OF RECEIVING BOX
C
      IA = IA + NSUBD2
      IIA = (IA - IXBW)/NSUBDV + 1
      INU = MXB1  /NSUBDV + 1
      JJJ = (JCOL-1)/NSUBDV + 1
      GO TO 420
C
C          SET UP POINTERS IF NO SUBDIVISION WAS REQUESTED
C
  410 CONTINUE
C     IA WAS SET TO IROW UPON ENTRY
      IIA = IA
      INU = 1
      JJJ=JCOL
C
C          DETERMINE THE CONTRIBUTION TO B FROM A FORWARD CONE OF UNSUB-
C          DIVIDED BOXES, STARTING WHERE SUBDIVISION LEFT OFF.
C              NU = ROW NUMBER OF CONTRIBUTING BOX RELATIVE TO RECEIVING
C                   BOX.
C
  420 DO 500 NU = INU,MXB
      IA = IA - NSUBDV
      IIA = IIA -1
C          HAS THE FORWARD EDGE OF THE PATTERN BEEN REACHED -
      IF (IA .LT. IXB) GO TO 600
C          NO.  GET BOX TYPE CODES FOR CURRENT ROW, UNSUBDIVIDED BOX
C          CENTERS ONLY
      LEFT = .T.
      IIAVLPN = IIA + IOVLPN
      NYB = IPNTDW(1,IIAVLPN+1) - IPNTDW(1,IIAVLPN) + IPNTDW(2,IIAVLPN)
     1        - 1
      CALL DCODER(IBOX,LBXCD, IIA,1, IIA,NYB, .F., IBXCD)
C              IBXCD = ROW OF BOX CODES
C              NYB   = NUMBER FOUND
C
C          GET LOCATION IN UNSUBDIVIDED DOWNWASH ARRAY FOR BOX(IIA,JJJ)
      IDW  = LOCSDW(IIAVLPN,JJJ, IPNTDW, IIAVLPN+1,1,IIAVLPN+2)
C
      N = (NU*(NU+1))/2 + 1
C              N      = UNSUBDIVIDED KERNEL SUBSCRIPT FOR MU = 0.
C
C          CENTER BOX
      IF (JJJ .GT. NYB) GO TO 425
      IF (IBXCD(JJJ) .NE. 0) B = B + FKERNL(N)*(ENRUS(IDW)-EDRLS(IDW) )
  425 CONTINUE
C
C          GOING OUT FROM CENTER CHORD IN BOTH DIRECTIONS
      IDWR = IDW+1
      IDWL = IDW-1
      IBXR = JJJ+1
      IBXL = IBXR-2
      L = -1
      E =  1.0
```

```
      N =  N + 1
C
      DO 480 MU = 1,NU
C
C           RIGHT SIDE
      IF (IBXR .GT. NYB) GO TO 430
      IF (IBXCD(IBXR) .EQ. 0) GO TO 430
      B = B + PKERNL(N) * (ENRUS(IDWR) - ENRLS(IDWR) )
  430 CONTINUE
C
C           LEFT SIDE
C
      IF (.NOT. LEFT) GO TO 470
C           HAS PLANFORM CENTER LINE BEEN ENCOUNTERED -
      IF (IBXL .GT. 0) GO TO 450
C           YES.  SET PARAMETERS TO SWEEP BACK ACROSS RIGHT SIDE
      LEFT = LSIDE
      IF (.NOT. LEFT) GO TO 470
      E  = SYMTY
      L = 1
      IBXL = 1
      IDWL = IDWL + 1
C           TEST FOR ROW COMPLETE, AS IN SUBDIVIDED LOGIC
  450 IF (IBXL .GT. NYB) GO TO 470
  460 IF (IBXCD(IBXL) .EQ. 0) GO TO 470
      B = B + PKERNL(N) * (ENRUS(IDWL) - ENRLS(IDWL) )  * E
  470 CONTINUE
C           SET COUNTERS FOR NEXT STEP OUTWARD
      IDWR = IDWR+1
      IDWL = IDWL+L
      IBXR = IBXR+1
      IBXL = IBXL+L
      N = N + 1
  480 CONTINUE
C           END OF LOOP ON MU (CHORDS OUTWARD)
C
  500 CONTINUE
C           END OF LOOP ON NU (ROWS FORWARD) FROM  420
C
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,NSFAIC,NOUTP,         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 / NWWK,  NTTK,  NRWTK,  NLWTK,   PAIC(4,50)         PAICS   00002
      INTEGER PAIC                                                      PAICS   00003
      DIMENSION NK(4)                                                   PAICS   00004
      EQUIVALENCE (NWWK,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,LPNTSD,KSDW,LSDW,KPNTDW,LPNTDW,     ARRAYS  00004
     3                KDW,LDW,KTVP,LTVP                                 ARRAYS  00005
      LOGICAL MXWRIT,RANDOU                                             GETAIC  00022
      DATA  MXWRIT,RANDOU / .F.,.F. /                                   GETAIC  00023
      DATA IPAIC,IPAICL,IPNT,   IFLAG,IFLAGL, ICODEP / 6*0 /            FTNX1   00061
C                                                                       GETAIC  00024
      IR = 0                                                            GETAIC  00025
      IF (IPNT .NE. 0) GO TO 160                                        GETAIC  00026
C                                                                       GETAIC  00027
C         INITIAL CALL.  SET UP FILES AND POINTERS                      GETAIC  00028
      REWIND IAICSC                                                     GETAIC  00029
      IPNT = 1                                                          GETAIC  00030
C         EXPAND PAIC ARRAY                                             GETAIC  00031
      I = 4                                                             GETAIC  00032
      IF (NRWTK   .EQ. 0) I = 3                                         GETAIC  00033
      IF (NTTK    .EQ. 0) I = I - 1                                     GETAIC  00034
      IF (NWWK    .EQ. 0) I = I - 1                                     GETAIC  00035
      IF (NLWTK   .EQ. 0) GO TO 120                                     GETAIC  00036
      IF (I .EQ. 4) GO TO 140                                           GETAIC  00037
      DO 110 J = 1,NLWTK                                                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 (NRWTK .EQ. 0) GO TO 130                                       GETAIC  00044
      IF (I .EQ. 3) GO TO 140                                           GETAIC  00045
```

```
      DO 125 J = 1, NRWTK
      PAIC(3,J) = PAIC(I,J)
      PAIC(I,J) = 0
  125 CONTINUE
      I = I - 1
  130 CONTINUE
      IF (NTTK .EQ. 0) GO TO 140
      IF (I .EQ. 2) GO TO 140
      DO 135 J = 1,NTTK
      PAIC(2,J) = PAIC(I,J)
      PAIC(I,J) = 0
  135 CONTINUE
  140 CONTINUE
C           ZERO OUT THE AIC ARRAYS
      DO 150 I = 1,LKERNL
      C(I) = (0.,0.)
      W(I) = (0.,0.)
      V(I) = (0.,0.)
  150 CONTINUE
C
C           GET THE AIC LOCATION
  160 CONTINUE
      IF (NC(ITPE) .LT. JUCENT) GO TO 290
      ILOC = PAIC(ITPE,JUCENT)
C           ARE THE DESIRED ARRAYS ALREADY IN CORE -
      IF (ILOC .EQ. IPAIC) GO TO 300
      IPAIC = ILOC
      ILOC = (ILOC-1)*4 + 1
C
C           GET THE MUAIC ARRAY  FROM THE NON-PLANAR AIC SCRATCH FILE
C
C           SPACE AND READ MUAICS
      CALL RDINIT
      IF (ILOC - IPNT) 200,220,210
C           MUAICS ARE BEHIND CURRENT LOCATION
  200 REWIND IAICSC
      NNG = ILOC -1
      GO TO 220
C
C           REQUIRED MUAICS ARE AHEAD OF CURRENT POSITION
  210 CONTINUE
      NNB =  ILOC - IPNT
C
C           READ MUAICS FROM IAICSC
  220 CONTINUE
      MXARRY = 9H    MUAIC
      K = 2
      CALL READMX( IAICSC, MXWRIT,RANDOU, NFB,NNB, LS,NNR, K, NID, ID,
     1              ITYPE,  LRS,  MUAIC, M,N, PARM, IRR)
      IF (IRR .NE. 0) GO TO 3000
      NROWB = N
      IPNT  = ILOC + 1
      EL    = PARM(5)
      YBAR  = PARM(4)
      ICD   = IPARM(6)
C
      CALL RDINIT
```

```
C         IS THE C ARRAY DESIRED -
      IF (ICODE .NE. 0) GO TO 250
C         YES.  IS IT AVAILABLE  -
      IF (ICD .EQ. 0) GO TO 240
C         NO.  SET THE ERROR FLAG AND CONTINUE
      IR = 1
      GO TO 250
C         READ THE  C  ARRAY FROM IAICSC
  240 CONTINUE
      MXARRY = 9HSPATIAL C
      CALL READMX( IAICSC, MXWRIT,RANDOU, NFS,NMS, LS,NMR, K, NID, ID,
     1              ITYPE,  LRS,  C,  M,N,  PARM, IRR)
      IF (IRR .NE. 0) GO TO 3000
      CALL RDINIT
      GO TO 260
  250 CONTINUE
      NMS = 1
  260 CONTINUE
      IPNT = IPNT + 1
C         READ THE  W  ARRAY FROM IAICSC
      MXARRY = 9HSPATIAL W
      CALL READMX( IAICSC, MXWRIT,RANDOU, NFS,NMS, LS,NMR, K, NID, ID,
     1              ITYPE,  LRS,  W,  M,N,  PARM, IRR)
      IF (IRR .NE. 0) GO TO 3000
      IPNT = IPNT + 1
      CALL RDINIT
C         IS THE V ARRAY DESIRED -
      IF (ICODE .EQ. 2) GO TO 300
C         YES.  IS IT AVAILABLE -
      IF (ICD .NE. 2) GO TO 280
C         NO.  SET ERROR FLAG
      IR = IR + 1
      GO TO 300
C         READ THE  V  ARRAY FROM IAICSC
  280 CONTINUE
      MXARRY = 9HSPATIAL V
      CALL READMX( IAICSC, MXWRIT,RANDOU, NFS,NMS, LS,NMR, K, NID, ID,
     1              ITYPE,  LRS,  V,  M,N,  PARM, IRR)
      IF (IRR .NE. 0) GO TO 3000
      IPNT = IPNT + 1
      GO TO 300
C
C         NO AICS CAN BE FOUND OF THE TYPE DESIRED FOR THIS CHORD
  290 IR = 3
C
  300 CONTINUE
      RETURN
C
C         DIAGNOSTIC
C
 3000 CONTINUE
      WRITE (NT6,9000) IAICSC,IRR
      WRITE (NT6,9192) MXARRY, M,N
      CALL FLUSH(1)
 9000 FORMAT(49H*** ERROR - WHILE READING FROM SPATIAL AIC FILE ,A10,
     1      14H, ERROR CODE = I4, 4H *** )
 9192 FORMAT(14X,A10,    20HARRAY, DIMENSIONED (I4,1H,I4,11H) WAS BEING
```

```
1         5H READ )                                          GETAIC  00160
 END                                                         GETAIC  00161
```

```
      SUBROUTINE STOSDW(IROW,JCOL, EN, IBOX,LBXCD, IXB,MXBS,MYB, IRR)
C
C        STORES A COMPUTED DOWNWASH VALUE IN THE END-AROUND SUBDIVIDED
C        DOWNWASH ARRAY, AND UPDATES POINTERS WHEN NECESSARY
C
C             IROW   = BOX CHORDWISE LOCATION
C             JCOL   = BOX SPANWISE LOCATION
C             EN     = COMPLEX NORMAL-WASHES TO BY STORED
C             IBOX   = ARRAY OF BOX CODES
C             LBXCD  = LENGTH OF BOX CODE ARRAY
C             IXB    = FIRST SUBDIVIDED ROW OF THE PLANFORM
C             MXBS   = MAXIMUM CHORD LENGTH OF SUBDIVIDED PATTERN
C             MYB    = MAXIMUM ROW LENGTH
C        RETURNS -
C             IRR    = ERROR RETURN, 0 = SUCCESSFUL
C                    = 1, FUNCTION LOCSDW FOUND THE POINTER OUTSIDE
C                         THE DEFINED SET OF DOWNWASHES
C             ENSUBD = SUBDIVIDED NORMAL-WASH ARRAY WITH ADDED VALUE
C
C        COMMON PARAMETERS USED
C             MXSKRN = MAXIMUM SIZE OF THE SUBDIVIDED KERNEL
C             LSDW   = DIMENSION OF SUBDIVIDED NORMAL WASH ARRAY
C             IPNTSD = POINTER ARRAY FOR SUBDIVIDED NORMAL WASH ARRAY
C             IPNTIN = NEXT AVAILABLE CELL IN IPNTSD
C             IPNTOT = FIRST CURRENTLY VALID CELL IN IPNTSD
C             IPNTLS = DIMENSION OF IPNTSD
C
      COMMON / KERN / ERR,MXSKRN,IPKERN,NFLKRN,NEPATK,NRCWEA
      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,LTVP
      COMMON /SNWASH/ IPNTSD(2,50), ENSUBD(2,600),  IPNTIN,IPNTOT,IPNTLS
C                    IPNTSD(LPNTSD), ENSUBD(2*LSDW)
      COMPLEX ENSUBD
      COMMON /CHECKPR/ DPPCPR,GEOCPR,MODCPR,AICCPR,NWSCPR,SMCPR,GAFCPR
      LOGICAL  DPPCPR, GEOCPR, MODCPR, AICCPR,NWSCPR,  SMCPR, GAFCPR
      EQUIVALENCE (CHECKPR,NWSCPR)
      LOGICAL CHECKPR
      DIMENSION TITL(3)
C
C
      COMPLEX EN(2)
C
      IRR = 0
C         IS THIS THE INITIAL CALL -
      IF (IROW .EQ. IXB .AND. JCOL .EQ. 1) GO TO 700
C         NO.  IS A NEW ROW BEING CONSIDERED -
      IF (IROW .GT. IOROW) GO TO 200
C         NO.  GET THE LOCATION FOR THE VALUE IN THE SUBDIVIDED DOWN-
C              WASH ARRAY
      IJ = LOCSDW(IROW,JCOL, IPNTSD,IPNTIN,IPNTOT,IPNTLS)
      IF (IJ) 900,900,550
C
C         MUST UPDATE POINTERS AND ADD A ROW TO THE SUBDIVIDED BOX ARRAY
  200 CONTINUE
      IOROW = IROW
```

```
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. LSDW + 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
      IPI = 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
  440 CONTINUE
C         HAS THE ARRAY BEEN OVER-WRITTEN IN GOING END-AROUND -
      IF (IPO .GT. IPI) GO TO 500
C         YES.  MOVE IPNTOT UNTIL CLEAR.
      IPNTOT = MOD(IPNTOT,IPNTLS) + 1
      IPO = IPNTSD(1,IPNTOT)
      IF (IPO .NE. 1) GO TO 440
C
C         ALL POINTERS HAVE BEEN RESET.  GET LOCATION
  500 CONTINUE
      IJ = LOCSDW(IROW,JCOL, IPNTSD,IPNTIN,IPNTOT,IPNTLS)
      IF (IJ .EQ. 0) GO TO 900
C
C         STORE THE DOWNWASH VALUE
  550 CONTINUE
      ENSUBD(1,IJ) = EN(1)
      ENSUBD(2,IJ) = EN(2)
  600 RETURN
C
C
C         INITIAL CALL
  700 CONTINUE
      IPNTOT = MOD(IXB-1,IPNTLS) + 1
      IPNTIN = IPNTOT
      MX     = MIN0(MXBS-IXB+1,  MXSKRN,  IPNTLS-1)
C         SET UP POINTER ARRAY FOR FIRST PASS
      CALL POINTR(IXB, MX,MYB, .T., .T., IBOX,LBXCD, IPNTLS,
     1     1, IPNTIN, IPNTSD)
  720 CONTINUE
      IF (IPNTSD(1,IPNTIN) .LE. LSDW   )   GO TO 730
      IPNTIN = IPNTIN - 1
      MX = MX - 1
      GO TO 720
  730 CONTINUE
      IOROW = MX + IXB - 1
C
      ENSUBD(1,1) = EN(1)
      ENSUBD(2,1) = EN(2)
      GO TO 600
C
C         ERROR
  900 IRR = 1
      GO TO 600
C
      END
```

```
      SUBROUTINE PRINTR( TITL,IMODE,ARRAY,K,IXB,MXB,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/ 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 /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUTP,          FILES   00002
     1                 IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC          FILES   00003
      COMMON /KVAL  / IKVAL,XKVAL(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    / 10HPAGE CONTI,4HNUED /                              FTNXX1  00063
      DATA   BLANK / 1H  /                                               FTNXX1  00064
      DATA   XINIT / -1.0 /                                              FTNXX1  00065
      K1 = XKVAL(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,MXB                                                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) TITLE,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 800                                                          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
      IF(JE.EQ.0) GO TO 1400
      DO 1000 J=JS,JE
      S(J) =  REAL(ARRAY(1,IDX) )
      D(J) = AIMAG(ARRAY(1,IDX) )
      IDX = IDX +1
 1000 CONTINUE
      DO 1200 J=N,M
      IF(S(J)) 1300,1100,1300
 1100 CONTINUE
      IF(D(J)) 1300,1200,1300
 1200 CONTINUE
      GO TO 1400
 1300 WRITE (NT6,9013) I,(S(J),D(J),J=N,M)
      LINE = LINE + 1
 1400 CONTINUE
      M = M+4
      N = N+4
      IF(N.GT.MYB ) GO TO 1500
      IF(M.GT.MYB ) M=MYB
      IF(LINE.GT.45) GO TO 100
      WRITE(NT6,6006) (BLANK,J,J=N,M)
      WRITE(NT6,6007) (BLANK,  J=N,M)
      LINE = LINE+3
      GO TO 200
 1500 CONTINUE
      RETURN
 9001 FORMAT(1H1,20X,8A10/ 50X,3A10/ 46X, 7H( MACH F5.3,5X,10HRED. FREQ.
     1     *=*,F8.5, * )* /52X,*MODE   SHAPE* ,I3)
 9005 FORMAT(44X,42(1H-),20X,A10,A4)
 6006 FORMAT(4HOROW, A1,14X,5HCHORD,I3, 3(A1,22X,5HCHORD,I3) )
 6007 FORMAT(3X, 4(A1,9X,4HREAL,8X,9HIMAGINARY) )
 9013 FORMAT(I4,8E16.8)
      END
```

```
      SUBROUTINE DCODER(IBOX,LBOX, IA,JA,  IL,JL,  SUBD, ICODE)
      DIMENSION  IBOX(LBOX,1), ICODE(1)
C
C         IBOX   - ARRAY OF BOX CODES IN PACKED WORD FORMAT
C         LBOX   - ROW DIMENSION OF BOX CODES ARRAY
C         IA     - I-TH INDEX OF FIRST CODE TO RETRIEVE
C         JA     - J-TH INDEX OF FIRST CODE TO RETRIEVE
C         IL     -  LAST BOX CODE ON THE JA-TH CHORD TO RETRIEVE
C         JL     -  LAST BOX ON THE IA-TH ROW TO RETRIEVE
C         SUBD   - .T., SUBDIVIDED BOX CODES DESIRED, .F. UNSUBDIVIDED.
C         ICODE  - ARRAY INTO WHICH BOX CODE WILL BE STORED.
C
C         COMMENT ON USAGE
C             BOX CODES CAN BE RETRIEVED FOR ONE BOX, A ROW OR PART OF
C         A ROW, OR A COLUMN OR PART OF A COLUMN. A ROW AND COLUMN CAN
C         NOT BE RETRIEVED AT THE SAME TIME. IF ONLY 1 BOX IS DESIRED
C         SET  IL = IA AND  JL = JA.   IF BOTH  IL .NE. IA AND  JL .NE.
C         JA, ONE ROW WILL BE RETURNED,  IL BEING IGNORED.
C
      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
      LOGICAL SUBD
      INTEGER SHIFT
      DATA NBWRD /20/
      MASK = 7
      IB = 1
      IF (SUBD) GO TO 50
      I = NSUBDV * (IA-1) + IXBW
      J = NSUBDV * (JA-1) + NSUBCN
      ISKIP = NSUBDV
      IEND = NSUBDV * (IL-1) + IXBW
      JEND = NSUBDV * (JL-1) + NSUBCN
      GO TO 60
   50 CONTINUE
      I = IA
      J = JA
      ISKIP = 1
      IEND = IL
      JEND = JL
   60 CONTINUE
      IF (JL  .EQ. JA) GO TO 1100
C
C         PROGRAM WILL RETRIEVE NI BOXES FROM ROW I
  100 CONTINUE
      DO 1000 JJ = J,JEND,ISKIP
      JSB =(JJ-1)/NBWRD + 1
      IJWORD = IBOX(I,JSB)
      JB = (NBWRD - MOD(JJ,NBWRD) ) * 3
      IF(JB.EQ.60) JB = 0
C         JB = NUMBER OF BITS TO SHIFT LEFT.
      IJMASK= SHIFT(MASK,JB)
      IJCODE  =   IJWORD.AND.IJMASK
      NJB = -JB
      ICODE(IB) = SHIFT(IJCODE,NJB)
```

```
      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
      DO 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
```

```
      SUBROUTINE POINTR( IX,MX, MYB,          SUBD, DIAPH, IBOX,LBXCD,
     1                  MXIR,  IPOINT,IPNTIN,IPNTR )
C
C        GENERATES A POINTER ARRAY WHICH SERVES AS AN INDEX TO A
C        CONDENSED ARRAY OF BOX VALUES (MODES, DOWNWASHES, ETC.)
C
C        IX     = CENTER OF FIRST BOX TO USE
C        MX     = NUMBER OF ROWS TO PROCESS
C        MYB    = MAXIMUM ROW LENGTH
C        SUBD   = .T., SUBDIVIDED BOXES TO BE USED
C               = .F., UNSUBDIVIDED BOXES TO BE USED
C        DIAPH  = .T.,DIAPHRAGMS TO BE INCLUDED
C               = .F., ONLY PLANFORM BOXES
C        IBOX   = ARRAY OF BOX CODES
C        LBXCD  = ROW DIMENSION OF BOX CODES ARRAY
C        MXIR   = SIZE OF IPNTR ARRAY
C        IPOINT = VALUE TO BE USED FOR FIRST POINTER (NORMALLY 1)
C     IN/OUT -
C        IPNTIN = LOCATION OF NEXT ABAILABLE LOCATION IN IPNTR ARRAY
C                    (OVERLAP OF TAIL IS ACCOUNTED FOR HERE.)
C     OUTPUTS -
C        IPNTR  = POINTER ARRAY - OUTPUT FROM THE SUBROUTINE
C              IPNTR(1,I) = LOCATION OF THE FIRST VALUE FOR ROW I
C              IPNTR(2,I) = CHORD LOCATION (SUBSCRIPT J) OF THAT VALUE
C
C   COMMON VALUES USED -
C
C
      DIMENSION IBOX(LBXCD,1),  IPNTR(2,1), ICODE(150)
      LOGICAL   DIAPH,SUBD,WING
C
      ITOTBX = IPOINT
      IL = IX + MX - 1
      DO 100 IROW = IX,IL
      IPBSUM=0
      IUISUM=0
      IPBX=1
      IBXSUM = 0
      CALL DCODER(IBOX,LBXCD, IROW,1, IROW,MYB, SUBD, ICODE)
C
C         FIND LAST BOX ON ROW
      M = MYB
      IF (DIAPH) GO TO 20
C         ONLY PLANFORM BOXES DESIRED
      DO 15 JCOL = 1,MYB
      IF (ICODE(M) .EQ. 1) GO TO 30
      M = M - 1
   15 CONTINUE
      GO TO 92
C         PLANFORM AND DIAPHRAGM DESIRED
   20 CONTINUE
      DO 25 JCOL = 1,MYB
      IF (ICODE(M) .NE. 0) GO TO 30
      M = M - 1
   25 CONTINUE
      GO TO 92
C
```

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

```
      FUNCTION LOCSDW(IROW,JCOL,IPNTSD,IPNTIN,IPNTOT,IPNTLS)
C
C        RETURNS THE LOCATION OF THE WORD IN THE END-AROUND SUBDIVIDED
C        DOWNWASH ARRAY CORRESPONDING TO BOX(IROW,JCOL) OF THE SUB-
C        DIVIDED BOX ARRAY
C
C             IROW   = BOX CHORDWISE LOCATION
C             JCOL   = BOX SPANWISE LOCATION
C             IPNTSD = ARRAY OF POINTERS
C             IPNTIN = NEXT AVAILABLE (UNUSED) CELL IN IPNTSD (END-
C                      AROUND)
C             IPNTOT = FIRST CURRENTLY AVAILABLE CELL IN IPNTSD
C             IPNTLS = LAST CELL OF IPNTSD (LENGTH OF ARRAY)
C        RETURN -
C             LOCSDW = LOCATION OF DESIRED DOWNWASH, IF SUCCESSFUL
C                    = 0, IF LOCPNT LIES OUTSIDE THE DEFINED AREA.
C
      DIMENSION IPNTSD(2,IPNTLS)
C
      LOCPNT = MOD(IROW-1,IPNTLS) + 1
C             LOCPNT = LOCATION OF CELL IN IPNTSD WHICH WAS OR IS TO BE
C                      USED
      IF(IPNTIN - IPNTOT) 100, 300, 200
C        END AROUND HAS OCCURRED
  100 IF (LOCPNT - IPNTIN) 400, 300, 150
C        NOT IN UPPER PART. IS LOCPNT WITHIN BOTTOM PART -
  150 IF (LOCPNT - IPNTOT) 300, 400, 400
C
C        NO END AROUND, NORMAL SEQUENCE
  200 IF (LOCPNT - IPNTIN) 250, 300, 300
C        LESS THAN UPPER LIMIT. IS LOCPNT .GE. LOWER LIMIT -
  250 IF (LOCPNT .GE. IPNTOT) GO TO 400
C
C        ERROR OR INITIAL CONDITION ENCOUNTERED (SHOULD NEVER OCCUR)
  300 LOCSDW = 0
      GO TO 500
C
C        SUCCESSFUL, BOX HAS BEEN DEFINED
  400 IFB = IPNTSD(2,LOCPNT)
      IF(JCOL.LT.IFB) GO TO 300
      LOCSDW = IPNTSD(1,LOCPNT) + JCOL-IFB
C
  500 CONTINUE
      RETURN
      END
```

```
      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 /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 /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, MUAICL(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,NOUTP,         FILES  00002
     1                IOUFSP,MODESC,IVPSC,IGEOSC,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      BCSPRB 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                       BCSPRB 00002
      COMMON / MODES/ SYM,SYMT,MTYPEW,MTYPET                            MODCOM 00002
      COMMON /NWASHES/ IPNTUW(2,100),ENRUS(1275), ENRLS(1275),IOVLAPN   NWASHES 00002
      COMPLEX                          ENRUS,       ENRLS               NWASHES 00003
      COMMON /AICS   / XKVL,    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
      COMPLEX WSUM,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
      CPSI = COS(PSIW)                                                  SMPLW  00033
      SPSI = 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
C        GET THE NECESSARY AIC ARRAYS
      CALL GETAIC(JCHRD,RWT, 0, IR)
      IF (IR .NE. 0) GO TO 800
      YMUBAR = ( JT-.5 + EL*SPSI) / CPSI
      JBAR  = YMUBAR
      IF (YMUBAR .GE. 0) JBAR = JBAR + 1
      NUBMIN = ABS(EL) + .5
      IBX = 1
C
C        LOOP ON BOXES ALONG THE SAMPLE CHORD
      DO 200 IBXX = IFRST,ILAST
      NUBMAX = IBXX -1
      I = IBXX - NUBMIN
C
C        ZERO OUT THE SUMMATION VARIABLES
      WSUM = (0.,0.)
      VSUM = (0.,0.)
      PHISUM = (0.,0.)
C
      IF (YBAR) 120,125,130
  120 JINCR = -1
      GO TO 135
  125 IAIC = NUBMIN**2
      INCAIC = 2*NUBMIN + 1
      JINCR = 1
      GO TO 140
  130 JINCR = 1
  135 IAIC  = NUBMIN**2 + NUBMIN
      INCAIC = 2*NUBMIN + 2
  140 CONTINUE
C
C        LOOP FORWARD OVER THE RIGHT WING
      DO 190 NUBAR = NUBMIN,NUBMAX
      MUAIC1 = MUAIC(1,NUBAR+1)
      MUAIC2 = MUAIC(2,NUBAR+1)
      IF (MUAIC2 .EQ. 0) GO TO 185
      IF (YBAR .GE. 0) GO TO 150
      JCOLR = JBAR + NUBAR - MUAIC2 + 1
      GO TO 160
  150 JCOLR = JBAR - NUBAR + MUAIC1 - 1
  160 CONTINUE
C
C        LOOP ON A ROW OF WING BOXES, COMPUTING RIGHT WING CONTRIBUTION
      DO 180 MUAI = MUAIC1,MUAIC2
      IF (JCOLR .LE. 0) GO TO 170
      CALL DCODER(IBOX,LBXCD, I,JCOLR, I,JCOLR, .F., ICD)
      IF (ICD .EQ. 0)  GO TO 170
C        A CONTRIBUTING BOX HAS BEEN FOUND.  GET THE AIC LOCATION
      KAIC = IAIC + MUAI
C        GET THE NORMAL-WASH LOCATION
      IDS = LOCSDW( I,JCOLR,  IPNTDW,LPNTDW, 1, LPNTDW)
C        ADD THIS CONTRIBUTION TO THE SUMS
      IF (EL .LT. 0) GO TO 163
      EN  = ENRUS(IDS)
      GO TO 165
  163 EN  = ENRLS(IDS)
```

```
  165 CONTINUE
      WSUM   = WSUM   + W(KAIC) * EN
      VSUM   = VSUM   + V(KAIC) * EN
      PHISUM = PHISUM + C(KAIC) * EN
  170 CONTINUE
      JCOLR = JCOLR + JINCR
  180 CONTINUE
C         END OF LOOP FOR RIGHT WING ROW CONTRIBUTIONS
C
  185 CONTINUE
      I = I - 1
      IF (I .LE. 0) GO TO 195
      IAIC = IAIC + INCAIC
      INCAIC = INCAIC + 2
  190 CONTINUE
C         END OF LOOP FORWARD ON RIGHT WING ROWS,  FROM 140*
C
  195 CONTINUE
      DW(IBX) =  BINV * (CPSI*WSUM + SPSI*VSUM)
      SW(IBX) =  BINV * (CPSI*VSUM - SPSI*WSUM)
      PHI(IBX) = PHISUM
      IBX = IBX + 1
  200 CONTINUE
C         END OF LOOP ON RECEIVING BOXES, FOR RIGHT WING CONTRIBUTIONS
C
      NBXS = IBX - 1
C         IS LEFT WING CONTRIBUTION NEEDED -
      IF (SYM .EQ. 0) GO TO 310
C         YES.  GET THE AIC ARRAYS FOR LEFT WING CONTRIBUTIONS.
      CALL GETAIC(JCHRD, LWT, 0, IR)
      IF (IR .NE. 0) GO TO 800
      NUBMIN = ABS(EL) + .5
      IBX = 1
      YMUBAR = (-JT+.5 + EL*SPSI) / CPSI
      JBAR  = YMUBAR
      IF (YMUBAR .GE. 0) JBAR = JBAR + 1
C         LOOP ON BOXES ALONG THE SAMPLE CHORD
      DO 300 IBXX = IFRST,ILAST
      NUBMAX = IBXX - 1
      I = IBXX - NUBMIN
C          ZERO OUT THE SUMMATION VARIABLES
      WSUM = (0.,0.)
      VSUM = (0.,0.)
      PHISUM = (0.,0.)
C
      IF (YBAR) 220,225,230
  220 JINCR = 1
      GO TO 235
  225 IAIC = NUBMIN**2
      INCAIC = 2*NUBMIN + 1
      JINCR  = -1
      GO TO 240
  230 JINCR  = -1
  235 IAIC = NUBMIN**2 + NUBMIN
      INCAIC = 2*NUBMIN + 2
  240 CONTINUE
C
```

```
C         LOOP FORWARD OVER THE LEFT WING
      DO 290 NUBAR = NUBMIN,NUBMAX
      MUAIC1 = MUAIC(1,NUBAR+1)
      MUAIC2 = MUAIC(2,NUBAR+1)
      IF (MUAIC2 .LE. 0) GO TO 285
      IF (YBAR .GE. 0) GO TO 250
      JCOLL =  JBAR -NUBAR +MUAIC1 -1
      GO TO 260
  250 JCOLL =  JBAR +NUBAR -MUAIC1 +1
  260 CONTINUE
C
C         LOOP ON A ROW OF WING BOXES, COMPUTING LEFT WING CONTRIBUTIONS
      DO 280 MUAI = MUAIC1,MUAIC2
      IF (JCOLL .LE. 0) GO TO 270
      CALL DCODER(IBOX,LBXCD, I,JCOLL, I,JCOLL, .F., ICD)
      IF (ICD .EQ. 0) GO TO 270
C         A CONTRIBUTING BOX HAS BEEN FOUND.  GET THE AIC LOCATION
      KAIC = IAIC + MUAI
C         GET THE NORMAL-WASH LOCATION
      IDS = LOCSDW( I,JCOLL, IPNTDW, LPNTDW, 1, LPNTDW)
C         ADD THIS CONTRIBUTION TO THE SUMS
      IF (EL .LT. 0) GO TO 263
      EN = ENRUS(IDS)
      GO TO 265
  263 EN = ENRLS(IDS)
  265 CONTINUE
      WSUM   = WSUM   + W(KAIC) * EN
      VSUM   = VSUM   + V(KAIC) * EN
      PHISUM = PHISUM + C(KAIC) * EN
  270 CONTINUE
      JCOLL = JCOLL + JINCR
  280 CONTINUE
C         END OF LOOP FOR LEFT WING ROW CONTRIBUTIONS
C
  285 CONTINUE
      I = I - 1
      IF (I .LE. 0) GO TO 295
      IAIC = IAIC + INCAIC
      INCAIC = INCAIC + 2
  290 CONTINUE
C         END OF LOOP FORWARD ON LEFT WING ROWS,  FROM 240
C
C
  295 CONTINUE
      DW(IBX)  = DW(IBX) + BINV*(CPSI*WSUM - SPSI*VSUM) * SYM
      SW(IBX)  = SW(IBX) + BINV*(CPSI*VSUM + SPSI*WSUM) * SYM
      PHI(IBX) = PHI(IBX) + PHISUM*SYM
      IBX = IBX + 1
  300 CONTINUE
C         END OF LOOP ON RECEIVING BOXES, FOR LEFT WING CONTRIBUTIONS
C
C         DETERMINE WHAT TO PRINT
  310 CONTINUE
      WRITE (NT6,6001) TITLE, XMACH,XKVL, JT, IFRST,ILAST
      IF (.NOT. PRDW) GO TO 330
      WRITE (NT6,6010)
      WRITE (NT6,6015) (DW(I),I = 1,NBXS)
```

```
  330 CONTINUE
      IF (.NOT. PRSW) GO TO 340
      WRITE (NT6,6011)
      WRITE (NT6,6013) (SW(I),I = 1,NBXS)
  340 CONTINUE
      IF (.NOT. PRLW) GO TO 400
      IF (NBXS .LT. 2) GO TO 400
      ISUB = 0
      PHI(ISUB) = 2.*PHI(1) - PHI(2)
      PHI(NBXS+1) = 2.*PHI(NBXS) - PHI(NBXS-1)
      FACTOR = B1BETA/2.0 * BINV**2
      DO 350 I = 1,NBXS
      LW(I) = FACTOR*(PHI(I+1) - PHI(I-1))
  350 CONTINUE
      WRITE (NT6,6012)
      WRITE (NT6,6013) (LW(I),I=1,NBXS)
C
  400 RETURN
C
  800 WRITE (NT6,8000) IR
      GO TO 400
C
 6001 FORMAT(1H1,20X,8A10/ 51X,18HFLOWFIELD SAMPLING /40X,7H( MACH ,
     1       F5.3,5X,12HRED. FREQ. =,F8.5, 2H )/ 41X,16HSAMPLED AT CHORD
     2       I3, 6H, BOX I2, 8H TO BOX I2 / 1H0,2X, 4(10X,4HREAL,8X,
     3       9HIMAGINARY) )
 6010 FORMAT(15H0  UP-WASHES - )
 6011 FORMAT(15H0SIDE-WASHES - )
 6012 FORMAT(22H0LONGITUDINAL-WASHES - )
 6013 FORMAT(4X,8E16.8)
C
 8000 FORMAT(54H0*** WARNING - PROBLEMS ENCOUNTERED WHILE GETTING AICS
     1  39H FOR FLOW-FIELD SAMPLING.  ERROR CODE = ,I5, 4H *** )
      END
```

```
      PROGRAM SMTH
C
C        THIS PROGRAM WILL FIT A LEAST SQUARES POLYNOMIAL SURFACE
C        THROUGH THE VELOCITY POTENTIALS FOR A PLANFORM. A NEW SET OF
C        VELOCITY POTENTIALS WILL BE CALCULATED FROM THE POLYNOMIALS
C        AND WRITTEN ON THE IVPSC FILE.
C
      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,LTVP
      COMMON /FILES / NT5,NT6,IHTAPE,INFSP,NPLAIC,NSPAIC,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 /PROBLM/ XMACH,NMODES,NTSLOP,NKVALS,SMOOTH,NDEG,CRDFIT,
     1                EXAIC,SUBDV,PLYWOOD
      LOGICAL         SMOOTH,CRDFIT,EXAIC,SUBDV,PLYWOOD
      COMMON /KVAL  / IKVAL,XKVAL(20), XKS(20)
      COMMON /GEOMTY/ COPLAN,NSUBDV,XSUBDV,NSUBD2,NSUBCN,NSURF,
     1                B1,B1BETA,B1S,B1BTAS,WLAX,WLAZ,PSIW,
     2                MXCW,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 /TAPEIO/ NFS,NMS,LS,NMR,ID(20),NID,ITYPE,LRS,LWS,M,N,
     1                PARM(10),IRR
      DIMENSION IPARM(10)
      EQUIVALENCE (PARM,IPARM)
      COMMON /CHECKPR/ DPPCPR,GEOCPR,MODCPR,AICCPR,NWSCPR,SMCPR,GAFCPR
      LOGICAL  DPPCPR, GEOCPR, MODCPR, AICCPR,NWSCPR,  SMCPR, GAFCPR
      EQUIVALENCE (CHECKPR,SMCPR )
      LOGICAL CHECKPR
C
C              DELPHI(NBOXES),      TVP(NCOLS1 + NCOLS2 * NSUBDV)
      COMPLEX  DELPHI(1000),        TVP(250),    AVPS(1250)
C               X(NO. DELPHI + NO. TVP), Y(SAME)
      COMPLEX  SDELPH
      DIMENSION  X(1250), Y(1250)
C               A(NO. COEFF.)
      COMPLEX   A(66)
C
      COMMON /INDEX/ IS(100),NOC(100),JS(100),JOC(100)
C               XP(NO. COEFF. + 1), YP(SAME)
      DIMENSION  XP(11), YP(11)
C               FEXLOC((MYBW+MYBT)*NSUBDV),    TEXLOC(SAME)
      DIMENSION  FEXLOC(250),                  TEXLOC(250)
C               IPNTRM(2,NROWS*NSUBDV)
      DIMENSION  IPNTRM(2,150)
      DIMENSION  TITL(3)
C
```

```
      REAL K1
      COMPLEX VP
      LOGICAL  MXREAD,MXWRIT,RANDIN,RANDOU
      K1 = XKVAL(IKVAL)
      MXREAD = .FALSE.
      RANDIN = .FALSE.
      MXWRIT = .FALSE.
      RANDOU = .FALSE.
      MXB = MXBW
      IF(COPLAN) MXB = MXBT
C
C
C         PUT NAME OF SCRATCH FILE FOR SMOOTHED VALUES INTO PLACE
C
      NIVPSC = IAICSC
      REWIND NIVPSC
C
C         GET THE PLANFORM POINTERS FROM THE MODESC FILE
C
      REWIND MODESC
      CALL RDINIT
      ITYPE = 5HMIXED
      MXARRY = 6HIPNTRM
      CALL READMX(MODESC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
     1            LRS,IPNTRM,2,NPNTRS,PARM,IRR)
      IOVLAP = IPARM(3)
      IF(IRR.NE.0) GO TO 6020
C
      CALL RDINIT
      ITYPE = 5HMIXED
      NFS = 1
      MXARRY = 6HIS PT.
      CALL READMX(MODESC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,
     1            LRS, IS, M, N, PARM, IRR)
      IF(IRR.NE.0) GO TO 6020
C
C
C         READ THE FEXLOC AND TEXLOC ARRAYS FROM THE GEOMETRY SCRATCH
C         FILE.  THESE ARE NEEDED TO INTERPOLATE VELOCITY POTENTIALS AT
C         BOX EDGES.
C
      REWIND IGEOSC
      CALL RDINIT
      NMS = 2
      IF(NSURF.EQ.1.OR.COPLAN) NMS=1
      ITYPE = 5HMIXED
      MXARRY = 6HFEXLOC
      CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,
     1            LRS,FEXLOC,M,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 6010
C
      CALL RDINIT
      ITYPE = 5HMIXED
      MXARRY =6HTEXLOC
      CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,
     1            LRS,TEXLOC,M,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 6010
```

```
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 = 5HMIXED                                                   SMOOTH  00123
      CALL READMX(IVPSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,    SMOOTH  00124
     1            LRS,DELPHI,M,N,PARM,IRR)                              SMOOTH  00125
      IF(IRR.NE.0) GO TO 6040                                           SMOOTH  00126
C                                                                       SMOOTH  00127
      CALL RDINIT                                                       SMOOTH  00128
      ITYPE = 5HMIXED                                                   SMOOTH  00129
      CALL READMX(IVPSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,    SMOOTH  00130
     1        LRS,TVP,M,N,PARM,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 AVP 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 = MXBT
      IC   = 0
      NCH  = MYBW
      IUP = MXBW
      IF(COPLAN) IUP = IFBT - 1
      DO 125 I=1,IUP
      IC = IC + JOC(I)
  125 CONTINUE
  200 CONTINUE
      ICS = IC
      INO = 0
      DO 400 IX=IBEG,ILIM
      I = IX
      IF(NS.EQ.2) I = IX + IOVLAP
      JST = JS(I)
      JEND = JS(I) + JOC(I)- 1
      DO 400 J=JST,JEND
      IC = IC + 1
      IB = IS(J+NCH)
      IT = IB + NOC(J+NCH) -1
      IF(IX.LT.IB) GO TO 400
      IF(IX.GT.IT) GO TO 400
      INO = INO + 1
      AVPS(INO) = DELPHI(IC)
      X(INO) = I
      Y(INO) = J
  400 CONTINUE
C
C        ADD THE LEADING EDGE VELOCITY POTENTIAL TO THE AVP ARRAY
C        VEL. POT. = 0. UNLESS IT IS FOR SECOND PLANFORM IN COPLANAR
C        ANALYSIS
C
      JLAST = MYBW
      IF(NS.EQ.2) JLAST = MYBT
      DO 600 J=1,JLAST
      IB = IS(J+NCH)
      INO= INO +1
      X(INO) = FEXLOC(J)
      Y(INO) = J
      AVPS(INO) = (0.,0.)
      IF(NS.EQ.1) GO TO 600
      IF(.NOT.COPLAN) GO TO 600
      XDKVL = (FEXLOC(J+MYBW)-TEXLOC(J)) * K1
      JT = J
      IF(NSUBDV.NE.1) JT = NSUBDV * (J-1) + NSUBCN
      AVPS(INO) = TVP(JT) * CMPLX(COS(XDKVL),-SIN(XDKVL))
  600 CONTINUE
C
C        CALL LEAST SQUARES SURFACE FITTING ROUTINE
C
      IDIM = 2
      CN   = 1.0
      IDEG = NDEG
      IF(NDEG.NE.0) GO TO 675
      DO 650 I=1,10
      IDEG = 10 -I+1
      XM = IDEG + 1
```

```
      XM2 = XM/2.0
      NC = XM * XM2 + XM2 + 1.0E-04
      NC = (3*NC) / 2
      IF(NC.LE.INO) GO TO 675
  650 CONTINUE
  675 CONTINUE
      CALL FITTER(IDEG,IND,X,Y,AVPS,A,CN,IDIM)
C
C           EVALUATE THE POLYNOMIAL EQUATION FOR DELPHI
C
      MDEG = IDEG + 1
      IC = ICS
      DO 900 IX=IBEG,ILIM
      I = IX
      IF(NS.EQ.2) I= IX +IOVLAP
      XP(1) = 1.
      DO 700 IP=2,MDEG
  700 XP(IP) =XP(IP-1) * FLOAT(I)
      JI = JS(I)
      JT = JOC(I) + JI - 1
      DO 900 J=JI,JT
      IC = IC + 1
      IB = IS(J+NCH)
      IT = IB + NOC(J+NCH) - 1
      IF(IX.LT.IB) GO TO 900
      IF(IX.GT.IT) GO TO 900
      YP(1) = 1.0
      DO 800 JP=2,MDEG
  800 YP(JP) = YP(JP-1) * FLOAT(J)
      VP = A(1)
      IA = 1
      DO 850 L2=2,MDEG
      DO 850 L3=1,L2
      L4 = L2 - L3 + 1
      IA = IA + 1
      VP = VP + XP(L4)*YP(L3)*A(IA)
  850 CONTINUE
      DELPHI(IC) = VP
  900 CONTINUE
C
C           CALCULATE THE TRAILING EDGE VELOCITY POTENTIALS (TVP ARRAY)
C
      IF(NS.EQ.2) GO TO 910
      NTST = 1
      NTVPS = MYBSW
      JJ = 0
      GO TO 920
  910 CONTINUE
      NTST = NTVPS + 1
      NTVPS = MYBSW + MYBST
  920 CONTINUE
      DO 930 J=NTST,NTVPS
  930 TVP(J) = (0.,0.)
C
      NBEG = NTST + NSUBD2
      NEND = NTVPS -NSUBD2
      JC = 0
```

```
      DO 960 J=NBEG,NEND,NSUBDV                                         SMOOTH  00268
C         J   = LOCATION IN THE TVP ARRAY (SUBDIVIDED VALUES)           BCSSHA  00001
C         JJ  = LOCATION IN ARRAYS TEXLOC, NOC, IS, ETC.                BCSSHA  00002
C         JC  = UNSUBDIVIDED CHORD NUMBER                               BCSSHA  00003
C         IROW = ROW NUMBER OF LAST BOX ON THE CHORD                    BCSSHA  00004
C         I    = ROW LOCATION OF IROW IN ARRAY IPNTRM                   BCSSHA  00005
      JJ = JJ + 1                                                       SMOOTH  00269
      JC = JC + 1                                                       SMOOTH  00270
      I  = TEXLOC(JJ)                                                   SMOOTH  00271
      IROW = I                                                          BCSSHA  00005
      XINCR = TEXLOC(JJ) - I                                            SMOOTH  00273
      IF (.NOT. COPLAN .AND. J .GT. MYBW) I = I + IOVLAP                SMOOTH  00274
      INDB  = IPNTRM(1,I) + JC - IPNTRM(2,I)                            BCSSHA  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                         BCSSHA  00008
C         3 BOXES OR MORE.  DO LINEAR EXTRAPOLATION.                    SMOOTH  00279
  935 CONTINUE                                                          BCSSHA  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                     BCSSHA  00010
      IB = IS(JJ-2)                                                     SMOOTH  00289
      IX = IB + NOC(JJ-2) + 1                                           SMOOTH  00290
      IM1 = IROW-1                                                      BCSSHA  00011
      IF (IM1 .GE. IB .AND. IM1 .LE. IX) GO TO 948                      BCSSHA  00012
C                                                                       BCSSHA  00013
C         MACH RAY CANNOT BE USED.  TEST FOR 2 BOXES ON CHORD JJ        BCSSHA  00014
  945 CONTINUE                                                          BCSSHA  00015
      IF (NOC(JJ) .LT. 2) GO TO 7010                                    BCSSHA  00016
      GO TO 935                                                         BCSSHA  00017
C                                                                       SMOOTH  00293
C         MACH RAY CAN BE USED                                          SMOOTH  00294
  948 CONTINUE                                                          BCSSHA  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
      TVP(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 TVP ARRAY ON THE NIVPSC FILE            SMOOTH  00306
      CALL RDINIT                                                       SMOOTH  00307
      ITYPE = 5HMIXED                                                   SMOOTH  00308
      N = IPNTRM(1,NENTRS) - 1                                          SMOOTH  00309
      NXARRY = 6HDELPHI                                                 SMOOTH  00310
```

```
      CALL WRTEMX(NIVPSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
     1             DELPHI,ITYPE,2,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 6040
C
      N = NTVPS
      MXARRY = 6HTVP
      CALL WRTEMX(NIVPSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
     1             TVP,    ITYPE,2,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 6040
C
C
C           PRINT THE SMOOTHED VEL. POT. ARRAY
      IF(.NOT.PRVP) GO TO 1500
      TITL(1) = 8H    WING
      TITL(2) =10HVELOCITY P
      TITL(3) =10HOTENTIALS
      IF(COPLAN) TITL(1) = 10HWING/TAIL
      CALL PRINTR(TITL, NM, DELPHI, 1, 1, MXB, MYBW, IPNTRM)
      IF(NSURF.EQ.1.OR.COPLAN) GO TO 1500
      TITL(1) = 8H    TAIL
      CALL PRINTR(TITL,    NM, DELPHI, 1,  IFBT, MXST, MYBT,
     1             IPNTRM(1,IOVLAP+1) )
 1500 CONTINUE
      IF (.NOT. CHECKPR) GO TO 2000
      WRITE (NT6,9400) (TVP(I),I=1,NTVPS)
 9400 FORMAT(// (1X,8E16.8) )
C
 2000 CONTINUE
C
      END FILE NIVPSC
      REWIND NIVPSC
C           CHANGE FILE NAMES
C
      IAICSC =  IVPSC
      IVPSC = NIVPSC
C
      RETURN
 6010 CONTINUE
      WRITE (NT6,9010) IGEOSC,IRR
      WRITE (NT6,9011) MXARRY
      GO TO 6100
 6020 CONTINUE
      WRITE (NT6,9010) MODESC,IRR
      WRITE (NT6,9011) MXARRY
      GO TO 6100
 6040 CONTINUE
      WRITE (NT6,9010) IVPSC,IRR
      WRITE (NT6,9041) NM
 6100 CONTINUE
      WRITE (NT6,9101) ID(1),ID(2)
      WRITE (NT6,9102) PARM,IPARM
      WRITE (NT6,9103) NFS,NMS
      WRITE (NT6,9104) ITYPE,M,N
      WRITE (NT6,9900)
      GO TO 8000
C
 7010 CONTINUE
```

```
      WRITE (NT6,8020)                                                  SMOOTH  00368
      WRITE (NT6,2021) IROW,JC                                          BCSSMA  00019
C                                                                       SMOOTH  00370
 8000 CONTINUE                                                          SMOOTH  00371
      CALL FLUSH(1)                                                     SMOOTH  00372
C                                                                       SMOOTH  00373
 9020 FORMAT(73H0*** 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(53H0*** 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
 9050 FORMAT(56H0*** 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,*MATRIX ID = *, A10, I10)                               SMOOTH  00385
 9102 FORMAT(5X,*PARAMETERS *,10E11.3, / 10X,*(INTEGER)*, I7, 9I11 )    SMOOTH  00386
 9103 FORMAT(5X,*FILE SPACING = *,I3,* MATRIX SPACING = *,I3 )          SMOOTH  00387
 9104 FORMAT(5X,*MATRIX TYPE =*,A10,*, DIMENSIONED (*I4,2H X,I4,1H) )   SMOOTH  00388
 9900 FORMAT(*0     ERROR OCCURRED IN SMOOTHING SECTION. *)             FTN0I   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,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 /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUTP,          FILES  00002
     1                ICUFSP,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,4HNUED /                              FTNXX1 00070
      DATA   BLANK / 1H  /                                               FTNXX1 00071
      DATA   XINIT / -1.0 /                                              FTNXX1 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 = IXB,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) TITLE,TITL, XMACH, 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,I)                                                   PRNTVP 00051
      IDX = IPNTRM(1,I)                                                  PRNTVP 00052
      JE =IPNTRM(1,I+1) - IDX + JS -1                                    PRNTVP 00053
```

```
      IF(JE.EQ.0) GO TO 1400
      DO 1000 J=JS,JE
      S(J) =  REAL(ARRAY(1,IDX) )
      D(J) = AIMAG(ARRAY(1,IDX) )
      IDX = IDX +1
 1000 CONTINUE
      DO 1200 J=N,M
      IF(S(J)) 1300,1100,1300
 1100 CONTINUE
      IF(D(J)) 1300,1200,1300
 1200 CONTINUE
      GO TO 1400
 1300 WRITE (NT6,9013) I,(S(J),D(J),J=N,M)
      LINE = LINE + 1
 1400 CONTINUE
      M = M+4
      N = N+4
      IF(N.GT.MYB ) GO TO 1500
      IF(M.GT.MYB ) M=MYB
      IF(LINE.GT.45) GO TO 100
      WRITE(NT6,6006) (BLANK,J,J=N,M)
      WRITE(NT6,6007) (BLANK,  J=N,M)
      LINE = LINE+3
      GO TO 200
 1500 CONTINUE
      RETURN
 9001 FORMAT(1H1,20X,8A10/ 46X,*SMOOTHED *,3A10/ 46X,7H( MACH F5.3,5X,
     1   12HRED. FREQ. =,F8.5, * )* / 52X,*MODE   SHAPE*, I3)
 9005 FORMAT(44X,42(1H-),20X,A10,A4)
 6006 FORMAT(4HGRCW, A1,14X,5HCHORD,I3, 3(A1,22X,5HCHORD,I3) )
 6007 FORMAT(3X, 4(A1,9X,4HREAL,8X,9HIMAGINARY) )
 9013 FORMAT(I4,8E16.8)
      END
```

```
      SUBROUTINE FITTER(M,N,X,Y,Z,C,CN,IDIM)
      DIMENSION X(100), Y(100), Z(IDIM,100), C(IDIM,66)
      DIMENSION AI(68),A(66,68),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 COEFFIENTS
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.
      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 + 1
      XM2= XM/2.
      NC = XM*XM2 + XM2 + EPS
      IF(NC.LE.N) 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
      NCV = 1
      MDX = M
      VS(1) = X(1)
      DO 60 I=1,N
      DO 50 J=1,NCV
      IF(X(I).EQ.VS(J)) GO TO 55
   50 CONTINUE
      NCV = NCV + 1
      VS(NCV) = X(I)
```

```
      IF(NDV-1.EQ.M) GO TO 65
   55 CONTINUE
   60 CONTINUE
      NDX = NDV -1
   65 CONTINUE
C
      NDV = 1
      NDY = M
      VS(1) = Y(1)
      DO 80 I=1,N
      DO 70 J=1,NDV
      IF(Y(I).EQ.VS(J)) GO TO 75
   70 CONTINUE
      NDV = NDV + 1
      VS(NDV) = Y(I)
      IF(NDV-1.EQ.M) GO TO 85
   75 CONTINUE
   80 CONTINUE
      NDY = NDV - 1
   85 CONTINUE
C
      ITOT =NC +1
      ITOT1 = ITOT
      IF(COMPLX) ITOT = ITOT + 1
C
C         ZERO OUT THE A ARRAY
C
      DO 95 I=1,NC
      C(1,I) = 0.0
      IF(.NOT.COMPLX) GO TO 90
      C(2,I) = 0.0
   90 CONTINUE
      DO 95 J=1,ITOT
   95 A(I,J) = 0.0
C
C         DETERMINE DEVIATION EQUATION AND SQUARE THE EQUATION
C
      AI(1) =1.0
      XP(1) =1.0
      YP(1) =1.0
      MM = M + 1
      DO 200 K=1,N
      DO 10 L=2,MM
      XP(L) = XP(L-1)*X(K)
      YP(L) = YP(L-1)*Y(K)
   10 CONTINUE
C
      I = 1
      DO 40 L=2,MM
      DO 20 LL=1,L
      IL= L - LL +1
      IF(LL-1.GT.NDY) GO TO 30
      IF(IL-1.GT.NDX) GO TO 20
      I = I + 1
      AI(I) = XP(IL)*YP(LL)
   20 CONTINUE
   30 CONTINUE
```

```
   40 CONTINUE
      AI(I+1) = Z(1,K)
      IF(COMPLX) AI(I+2) = Z(2,K)
      IF(K.GT.1) GO TO 45
      NC = I
      ITOT = NC + 1
      ITOT1 = ITOT
      IF(COMPLX) ITOT = ITOT + 1
   45 CONTINUE
C
      DO 1100 I=1,NC
      DO 1100 J=I,ITOT
      ASAV = AI(I)*AI(J)
      A(I,J)=A(I,J)+ASAV
 1100 CONTINUE
  200 CONTINUE
C
C           SQUARE ROOT METHOD
C           INTERMEDIATE MATRIX
      DO 1200 I=1,NC
      IM1 = I-1
      TMP=0.0
      IF(I.EQ.1) GO TO 1150
      DO 1120 L=1,IM1
 1120 TMP= TMP+ A(L,I)**2
 1150 CONTINUE
      T = A(I,I) - TMP
      IF(T.GT.EPS) GO TO 4
      A(I,I) = 0.0
      GO TO 1200
    4 CONTINUE
      A(I,I) = SQRT(T)
      IF(A(I,I).GT.EPS) GO TO 1155
      A(I,ITOT) = 0.0
      GO TO 1200
 1155 CONTINUE
C
      JS = I+1
      DO 1180 J = JS,ITOT
      TMP= 0.0
      IF(I.EQ.1) GO TO 1175
      DO 1160 L=1,IM1
 1160 TMP = TMP + A(L,I)*A(L,J)
 1175 A(I,J) =(A(I,J)-TMP)/A(I,I)
 1180 CONTINUE
 1200 CONTINUE
C
C
C           BACK SUBSTITUTE FOR COEFFICIENTS
      DO 1400 K=1,NC
      I = NC - K + 1
      IP1=I+1
      TMP1  = 0.0
      TMP2  = 0.0
      IF(A(I,I).GT.EPS) GO TO 1325
      C(1,I) = 0.0
      IF(COMPLX) C(2,I) = 0.0
```

```
      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,MM
      DO 1420 LL=1,L
      IL = L -LL +1
      I = I +1
      IF(LL-1.LE.MCY.AND.IL-1.LE.MCX) 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
```

```
C                                                           FITTER  00230
C         THE C ARRAY NOW CONTAINS THE COEFFICIENTS.        FITTER  00231
C                                                           FITTER  00232
      RETURN                                                FITTER  00233
      END                                                   FITTER  00234
```

```
      OVERLAY(GFMBOX,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,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                IOUFSP,MODESC,IVPSC,IGEOSC,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 /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                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
      EQUIVALENCE (PARM,IPARM)                                          TAPEIO  00005
      COMMON /CHECKPR/ DPPCPR,GEOCPR,MODCPR,AICCPR,NWSCPR,SMCPR,GAFCPR  CHECKPR 00002
      LOGICAL  DPPCPR, GEOCPR, MODCPR, AICCPR,NWSCPR,  SMCPR,  GAFCPR   CHECKPR 00003
      EQUIVALENCE (CHECKPR,SMCPR)                                       CHORDF  00019
      LOGICAL CHECKPR                                                   FTNXX1  00073
C                                                                       CHORDF  00020
C             DELPHI(NBOXES),      TVP(NCOLS1 + NCOLS2 * 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              FEXLOC((MYBW+MYBT)*NSUBDV),    TEXLOC(SAME)              CHORDF  00029
      DIMENSION FEXLOC(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, SDELPH, VC, AVPA, AVPB                               CHORDF  00036
      LOGICAL  MXREAD,MXWRIT,RANDIN,RANDOU                              CHORDF  00037
```

```
      K1 = XKVAL(IKVAL)
      MXREAD = .FALSE.
      RANDIN = .FALSE.
      MXWRIT = .FALSE.
      RANDOU = .FALSE.
      MXB = MXBW
      IF(COPLAN) MXB = MXBT
C
C
C        PUT NAME OF SCRATCH FILE FOR SMOOTHED VALUES INTO PLACE
C
      NIVPSC = IAICSC
      REWIND NIVPSC
C
C        GET THE PLANFORM POINTERS FROM THE MODESC FILE
C
      REWIND MODESC
      CALL RDINIT
      ITYPE = 5HMIXED
      MXARRY = 6HIPNTRM
      CALL READMX(MODESC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
     1            LRS,IPNTRM,2,NPNTRS,PARM,IRR)
      IOVLAP = IPARM(3)
      IF(IRR.NE.0) GO TO 6020
C
      CALL RDINIT
      ITYPE = 5HMIXED
      NFS = 1
      MXARRY =6HIS PT.
      CALL READMX(MODESC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,
     1            LRS, IS, M, N, PARM, IRR)
      IF(IRR.NE.0) GO TO 6020
C
C
C        READ THE FEXLOC AND TEXLOC ARRAYS FROM THE GEOMETRY SCRATCH
C        FILE.  THESE ARE NEEDED TO INTERPOLATE VELOCITY POTENTIALS AT
C        BOX EDGES.
C
      REWIND IGEOSC
      CALL RDINIT
      NMS = 2
      IF(NSURF.EQ.1.OR.COPLAN) NMS=1
      ITYPE = 5HMIXED
      MXARRY = 6HFEXLOC
      CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,
     1            LRS,FEXLOC,M,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 6010
C
      CALL RDINIT
      ITYPE = 5HMIXED
      MXARRY =6HTEXLOC
      CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,
     1            LRS,TEXLOC,M,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 6010
C
C        REORDER THE FEXLOC AND TEXLOC ARRAYS SO THAT THERE ARE
C        VALUES FOR UNSUBDIVIDED CHORDS ONLY.
```

```
C
      IF(NSUBDV.EQ.1) GO TO 120
      XSLIDE = NSUBDV - IXBW
      JCOL = NSUBCN
      NCOLS =  MYBW + MYBT
      DO 110 I=1,NCOLS
      TEXLOC(I) = (TEXLOC(JCOL) + XSLIDE)/XSUBDV
      FEXLOC(I) = (FEXLOC(JCOL) + XSLIDE)/XSUBDV
      JCOL = JCOL + NSUBDV
  110 CONTINUE
  120 CONTINUE
C
      IFBT = (IXBT-IXBW)/NSUBDV + 1
C
C         LOOP ON NUMBER OF MODES (ALSO NO. OF V.P.)
      REWIND IVPSC
      DO 2000 NM=1,MMODES
C
C         READ DELPHI ARRAY FROM IVPSC. THE TVP ARRAY MUST BE SKIPPED
C         IF NM IS NOT 1
C
      CALL RDINIT
      ITYPE = 5HMIXED
      NMS = 1
      IF(NM.EQ.1) NMS = 0
      CALL READMX(IVPSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
     1            LRS,DELPHI,M,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 6040
C
C         LOOP ON NUMBER OF CHORDS
C
      NCHRDS = MYBW
      IF(NSURF.EQ.2) NCHRDS = MYBW + MYBT
      DO 1000  J=1,NCHRDS
      NC = 1
      IF(J.GT.MYBW) NC = MYBW + 1
      IST = IS(J)
      NK  = IST + NOC(J) - 1
      JSUM= 0
      ITROW = IST
      IF(.NOT.COPLAN.AND.J.GT.MYBW) ITROW = IST + IOVLAP
      DO 100 I=1,ITROW
  100 JSUM = JSUM + JOC(I)
      JSUM = JSUM - JOC(ITROW) + 1
      IND = 0
      DO 200 I=IST,NK
      IX = I
      IND = IND + 1
      IF(.NOT.COPLAN.AND.J.GT.MYBW) IX = I + IOVLAP
      ISUB = JSUM + J - JS(IX) - NC + 1
      X(IND) = I
      AVPS(IND) = DELPHI(ISUB)
      JSUM = JSUM + JOC(IX)
  200 CONTINUE
C
C         FIND THE DERIVATIVE OF DELPHI, AND SMOOTH THESE
C
```

```
      VC = AVPS(1)
      INDM1 = IND -1
      AVPA = 0.5 * (AVPS(1) + AVPS(2))
      AVPS(1) = (AVPS(2) - AVPS(1))/(X(2) - X(1))
      DO 350 I=2,INDM1
      AVPB = 0.5 * (AVPS(I) + AVPS(I+1))
      AVPS(I) = AVPB - AVPA
      AVPA = AVPB
  350 CONTINUE
      AVPS(IND) = (AVPS(IND) - AVPA)/0.5
      IND = IND + 1
      INDM1 = INDM1 + 1
      AVPS(IND) = AVPS(INDM1)
      X(IND) = X(INDM1) + 0.5
C
      XINC = X(1)
      DO 375 I=1,IND
      X(I) = X(I) - XINC
  375 CONTINUE
      IDEG = NDEG
C
C         CALL FITTING ROUTINE LEAST SQUARES ERROR CURVE
C
      CALL CURVE(IDEG,IND,X,AVPS,A)
C
C         EVALUATE THE CURVE FOR SMOOTH DELPHI VALUES
C
      MDEG = IDEG + 1
      JSUM = 0
      DO 400 I=1,ITROW
  400 JSUM = JSUM + JOC(I)
C
      JSUM = JSUM - JOC(ITROW) + 1
      DO 500 I = IST,NK
      IX = I
      IF(.NOT.COPLAN.AND.J.GT.MYBW) IX = I + IOVLAP
      ISUB = JSUM + J - JS(IX) - NC + 1
      VP = VC
      XV = FLOAT(I) - XINC
      XP = 1.0
      DO 450 L = 1,MDEG
      XP = XP * XV
      XD = L
      XPI = XP/XD
      VP = VP + A(L) * XPI
  450 CONTINUE
      DELPHI(ISUB) = VP
      JSUM = JSUM + JOC(IX)
  500 CONTINUE
C
C         CALCULATE THE TRAILING EDGE VELOCITY POTENTIALS (TVP ARRAY)
C
      L = J
      IF(NSUBDV.NE.1) L = NSUBDV * (J-1) + NSUBCN
      TVP(L) = (0.,0.)
      JJ = J
      JC = J
```

```
      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 + IOVLAP                     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
      IM1 = IROW-1                                                     BCSCFA  00005
      IF (IM1 .GE. IB .AND. IM1 .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 = 5HMIXED                                                  CHORDF  00247
      N = IPNTRM(1,NROWS) - 1                                          CHORDF  00248
      NXARRY = 6HDELPHI                                                CHORDF  00249
      CALL WRTEMX(NIVPSC,NXWRIT,RANDCU,NFS,NAS,LS,NMR,LWS,2,ID,        CHORDF  00250
     1            DELPHI,ITYPE,2,N,PARM,IRR)                           CHORDF  00251
      IF(IRR.NE.0) GO TO 6040                                          CHORDF  00252
C                                                                      CHORDF  00253
      NTVPS = NSUBDV * NCHRDS                                          CHORDF  00254
      N = NTVPS                                                        CHORDF  00255
      NXARRY = 6HTVP                                                   CHORDF  00256
```

```
      CALL WRTEMX(NIVPSC,MXWRIT,RANDOU,NFS,NMS,LS,NMR,LWS,2,ID,
     1            TVP,   ITYPE,2,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 6040
C
C        PRINT THE SMOOTHED VEL. POT. ARRAY
C
      IF(.NOT.PRVP) GO TO 1500
      TITL(1) = 8H   WING
      TITL(2) =10HVELOCITY P
      TITL(3) =10HOTENTIALS
      IF(COPLAN) TITL(1) = 10HWING/TAIL
      CALL PRINTR(TITL,NM,DELPHI,1,1,MXB,MYBW,IPNTRM)
      IF(NSURF.EQ.1.OR.COPLAN) GO TO 1500
      TITL(1) = 8H   TAIL
      CALL PRINTR(TITL,NM,DELPHI,1,IFBT,MXBT,MYBT,IPNTRM(1,IOVLAP+1))
 1500 CONTINUE
      IF(.NOT.CHECKPR) GO TO 2000
      WRITE(NT6,9400) (TVP(I),I=1,NTVPS)
 9400 FORMAT(//(1X,8E16.8))
C
 2000 CONTINUE
C
      END FILE NIVPSC
      REWIND NIVPSC
C        CHANGE FILE NAMES
C
      IAICSC = IVPSC
      IVPSC = NIVPSC
C
      RETURN
 6010 CONTINUE
      WRITE (NT6,9010) IGEOSC,IRR
      WRITE (NT6,9011) MXARRY
      GO TO 6100
 6020 CONTINUE
      WRITE (NT6,9010) MODESC,IRR
      WRITE (NT6,9011) MXARRY
      GO TO 6100
 6040 CONTINUE
      WRITE (NT6,9010) IVPSC,IRR
      WRITE (NT6,9041) NM
 6100 CONTINUE
      WRITE (NT6,9101) ID(1),ID(2)
      WRITE (NT6,9102) PARM,IPARM
      WRITE (NT6,9103) NFS,NMS
      WRITE (NT6,9104) ITYPE,M,N
      WRITE (NT6,9900)
      GO TO 8000
C
 7010 CONTINUE
      WRITE (NT6,8020)
      WRITE (NT6,8021) IROW,JC
C
 8000 CONTINUE
      CALL FLUSH(1)
C
 8020 FORMAT(73H0*** ERROR - NO TIP TRAILING EDGE VELOCITY POTENTIAL CAN
```

```
      1 BE COMPUTED. *** )                                              CHORDF  00314
 8021 FORMAT(5X,13HCOORDINATES (I2,1H,I2,1H) )                          CHORDF  00315
 9010 FORMAT(53HO*** 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(56HO*** 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,NSPAIC,NOUTP,         FILES   00002
     1                IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC          FILES   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,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
      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/ BFCODE,IBFCNT,  BUFF(3280)                        RWBUFF  00002
      DATA   PC    / 10HPAGE CONTI,4HNUED /                             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) TITLE,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
      DO 1000 J=JS,JE
      S(J) =  REAL(ARRAY(1,IDX))
      D(J) = AIMAG(ARRAY(1,IDX))
      IDX = IDX +1
 1000 CONTINUE
      DO 1200 J=N,M
      IF(S(J)) 1300,1100,1300
 1100 CONTINUE
      IF(D(J)) 1300,1200,1300
 1200 CONTINUE
      GO TO 1400
 1300 WRITE (NT6,9013) I,(S(J),D(J),J=N,M)
      LINE = LINE + 1
 1400 CONTINUE
      M = M+4
      N = N+4
      IF(N.GT.MYB) GO TO 1500
      IF(M.GT.MYB) M = MYB
      IF(LINE.GT.45) GO TO 100
      WRITE(NT6,8006) (BLANK,J,J=N,M)
      WRITE(NT6,8007) (BLANK,  J=N,M)
      LINE = LINE+3
      GO TO 200
 1500 CONTINUE
      RETURN
 9001 FORMAT(1H1,20X,8A10/ 46X,*SMOOTHED *,3A10/ 46X,7H( MACH F5.3,5X,
    1  12HRED. FREQ. =,F8.5, * )* / 52X,*MODE   SHAPE*, I3)
 9005 FORMAT(44X,42(1H-),20X,A10,A4)
 8006 FORMAT(4HOROW, A1,14X,5HCHORD,I3, 3(A1,22X,5HCHORD,I3) )
 8007 FORMAT(3X, 4(A1,9X,4HREAL,8X,9HIMAGINARY) )
 9013 FORMAT(I4,8E16.8)
      END
```

```
      SUBROUTINE CURVE(M,N,X,Z,C)
      DIMENSION X(50), Z(2,50), C(2,21)
      DIMENSION AI(23), A(21,23), XP(6)
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)
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
```

```
      A(I,I) = SQRT(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
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
```

```
      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,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                IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC          FILES   00003
      EQUIVALENCE (IWTFSC,ITSLSC)                                       FORCES  00011
      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 / KERN / ERR,MXSKRN,IPKERN,NPLKRN,NSPATK,NROWEA            KERN    00002
      COMMON /KVAL  / IKVAL,XKVAL(20), XKS(20)                          KVAL    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 / MODES/ SYM,SYMT,MTYPEW,MTYPET                            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 /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,N,N,      TAPEIO  00002
     1                PARM(10),IRR                                      TAPEIO  00003
      DIMENSION IPARM(10)                                               TAPEIO  00004
      EQUIVALENCE (PARM,IPARM)                                          TAPEIO  00005
      COMMON /CHECKPR/ DPPCPR,GEOCPR,MODCPR,AICCPR,NWSCPR,SMCPR,GAFCPR  CHECKPR 00002
      LOGICAL  DPPCPR, GEOCPR, MODCPR, AICCPR,NWSCPR,  SMCPR, GAFCPR    CHECKPR 00003
      EQUIVALENCE (CHECKPR,GAFCPR)                                      FORCES  00022
      LOGICAL CHECKPR                                                   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            FEXLOC((MYBW+MYBT)*NSUBDV),TEXLOC(SAME)                    FORCES  00031
      DIMENSION FEXLOC(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(NCOLS), IBXCD(NCOLS), IBXCDA(NCOLS)                 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),TVF(NCOLS1+NCOLS2*NSUBDV)                    FORCES  00045
      COMPLEX   DELPHI(1000),  TVF(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), GPFAFC(SAME)                          FORCES  00053
      DIMENSION GFAFC(400),           GPFAFC(400)                          FORCES  00054
C                                                                          FORCES  00055
      DIMENSION  AFC(2)                                                    FORCES  00056
      EQUIVALENCE (AFCSTR,AFC)                                             FORCES  00057
C                                                                          BCSFRB  00008
      COMPLEX  SECMOM(100), GAF                                            BCSFRB  00009
C                                                                          BCSFRB  00010
C              VPTE(NCOLS)                                                 FORCES  00058
      COMPLEX   VPTE(50),VPLE, TEMP1, TEMP2, TEMP3, BL                     FORCES  00059
      LOGICAL MXREAD,RANDIN,MXWRIT,RANDOU                                  FORCES  00060
      LOGICAL BLNEED                                                       FORCES  00061
      COMPLEX XINDEF                                                       FORCES  00062
      DIMENSION XINDFF(2)                                                  FTNXX1  00079
      EQUIVALENCE (XINDEF,XINDFF)                                          FTNXX1  00080
      COMMON /RWBUFF/ BFCODE,IBFCNT,  BUFF(3280)                           RWBUFF  00002
      MXREAD = .FALSE.                                                     FORCES  00064
      RANDIN = .FALSE.                                                     FORCES  00065
      MXWRIT = .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. PRCM               BCSFRB  00011
      TWOBET = (2.0*B1BETA)/B1                                             FORCES  00072
      TWOBB = TWOBET/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*S3                                                        FORCES  00077
      BS3BET = -B1BETA/S3                                                  FORCES  00078
      BKS4BT = 0.0                                                         BCSFRA  00004
      IF (XKVL .EQ. 0.) GO TO 5                                            BCSFRA  00005
      BKS4BT = -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
      REWIND MODESC                                                        FORCES  00087
      NMSPCE = 0                                                           FORCES  00088
```

```
C
C        READ THE PROINTERS FROM THE MODESC FILE.
C
      CALL RDINIT
      ITYPE = 5HMIXED
      MXARRY = 6HIPNTRM
      CALL READMX(MODESC,MXREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
     1            LRS,IPNTRM,M,N,FARM,IRR)
      IOVLAP = IFARM(3)
      NPNTRS = N
      MXB = NPNTRS -1
      MYB = MAX0(MYBW,MYBT)
      NBOXES = MYB * MXB
      IF(IRR.NE.0) GO TO 6020
C
C
      REWIND IGEOSC
C
C        READ BOX CODES INTO STORAGE FROM GEOMETRY SCRATCH FILE
C
      CALL RDINIT
      ITYPE = 6HMIXED
      MXARRY = 6HIBOXW
      CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,150,NID,ID,ITYPE,
     1          LRS,IBOXW,M,N,FARM,IRR)
      IF(IRR.NE.0) GO TO 6010
C
      NPLS = 1
      IF(NSURF.EQ.1.OR.COPLAN) GO TO 10
      NPLS = 2
C
      CALL RDINIT
      ITYPE = 6HMIXED
      MXARRY = 6HIBOXT
      CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR, 90,NID,ID,ITYPE,
     1          LRS,IBOXT,M,N,FARM,IRR)
      IF(IRR.NE.0) GO TO 6010
C
   10 CONTINUE
C
C        READ THE TEXLOC AND FEXLOC ARRAYS FROM THE GEOMETRY SCRATCH
C        FILE.  THESE ARE NEEDED TO INTERPOLATE VELOCITY POTENTIALS AT
C        BOX EDGES.
C
      CALL RDINIT
      ITYPE = 5HMIXED
      MXARRY = 6HFEXLOC
      CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,
     1            LRS,FEXLOC,M,N,FARM,IRR)
      IF(IRR.NE.0) GO TO 6010
C
      CALL RDINIT
      ITYPE = 5HMIXED
      MXARRY = 6HTEXLOC
      CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,
     1            LRS,TEXLOC,M,N,FARM,IRR)
      IF(IRR.NE.0) GO TO 6010
```

```
C
C         READ AREAS AND POINTERS FOR AREAS FROM GEOMETRY SCRATCH FILE.
C         THESE ARE USED IN CALCULATION OF BOX LIFTS AND GEN. FORCES.
C
      CALL RDINIT
      ITYPE = 5HMIXED
      MXARRY = 6HALPHAS
      CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,
     1            LRS,ALPHA,M,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 6010
C
      CALL RDINIT
      ITYPE = 5HMIXED
      MXARRY = 6HIJALPH
      CALL READMX(IGEOSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,
     1            LRS,IJALPH,M,N,PARM,IRR)
      NALPH = N
      NALPHW = IPARM(3)
      NALPHT = N - NALPHW
      IF(IRR.NE.0) GO TO 6010
C
C
C         REORDER THE FEXLOC AND TEXLOC ARRAYS SO THAT THERE ARE
C         VALUES FOR UNSUBDIVIDED CHORDS ONLY.
C
      IF(NSUBDV.EQ.1) GO TO 120
      XSLIDE = NSUBDV-IXBW
      JCOL = NSUBCN
      NCOLS = MYBW + MYBT
      DO 110 I=1,NCOLS
      TEXLOC(I) = (TEXLOC(JCOL)+XSLIDE)/XSUBDV
      FEXLOC(I) = (FEXLOC(JCOL)+XSLIDE)/XSUBDV
      JCOL = JCOL + NSUBDV
  110 CONTINUE
  120 CONTINUE
C
C         LOOP ON THICKNESS SLOPE FUNCTIONS (IF NONE WERE REQUESTED,
C         ONE DUMMY SET OF ONES WILL HAVE BEEN GENERATED.)
C
      REWIND ITSLSC
      DO 750 ITSLOP=1,NTSLOP
C
C         READ THICKNESS SLOPE FUNCTIONS
C
      CALL RDINIT
      ITYPE = 4HREAL
      MXARRY = 6HTSLFN
      CALL READMX(ITSLSC,MXREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,
     1            LRS,TSLFN,M,N,PARM,IRR)
      IF(IRR.NE.0) GO TO 6040
C
C         ZERO OUT THE AIR FORCES ARRAY
C
      DO 150 J=1,MAX
  150 GENAF(J) = (0.,0.)
C         LOOP ON NUMBER OF MODE SHAPES
      DO 650 NM=1,NMODES
```

```
C                                                                        FORCES  00203
C          GET MODE SHAPE NM FROM MODESC SCRATCH FILE                    FORCES  00204
      CALL RDINIT                                                        FORCES  00205
      IF(NM.EQ.1) NMS = NMSPCE                                           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 BXLIFT(I) = (0.,0.)
  240 CONTINUE
C
C
C*********************************************************************
C
C         THE FOLLOWING BLOCK OF CODE COMPUTES
C           BXLIFT  - BOX LIFTS
C           SLIFT   - SECTION (CHORD) LIFTS
C           TLIFT   - TOTAL LIFT
C           GENAF   - GENERALIZED AIRFORCES
C
C*********************************************************************
C
      AFROW(JVP) = (0.,0.)
C              AFROW  = CURRENT ROW OF GENERALIZED AIRFORCES
C
C         SET UP INITIAL CONDITIONS FOR DOUBLE LOOP OVER THE ENTIRE
C         BOX PATTERN
C              IBXCDF = FORWARD
C              IBXCD  = CENTER    ROWS OF BOX CODES, EXPANDED
C              IBXCDA = AFT      /
C              VPLE   = VELOCITY POTENTIAL AT BOX LEADING EDGE
C              VPTE   = ARRAY OF BOX TRAILING EDGE VELOCITY POTENTIALS
C
      DO 565 NP=1,NPLS
      IF (NP.EQ.2) GO TO 245
      ISROWA = 1
      NBXA = IPNTRM(1,2)
      CALL DCODER(IBOXW,150,ISROWA,1,ISROWA,NBXA,.F.,IBXCDA)
      NBX = NBXA
      GO TO 250
  245 CONTINUE
      ISROWA = (IXBT-IXBW)/NSUBDV + 1
      IXBUT = ISROWA
      IDEX = ISROWA + IOVLAP
      NBXA = IPNTRM(1,IDEX+1) - IPNTRM(1,IDEX)
      ISUBT = 2-IXBST
      CALL DCODER(IBOXT(ISUBT,1),LBXCDT,ISROWA,1,ISROWA,NBXA,.F.,IBXCDA)
      NBX = NBXA
  250 CONTINUE
      DO 270 JCOL = 1,NBXA
      IBXCD(JCOL) = IBXCDA(JCOL)
      IF (IBXCDA(JCOL) .EQ. 1) GO TO 260
      VPTE(JCOL) = XINDEF
      DEFLTE(JCOL) = XINDEF
      GO TO 270
  260 CONTINUE
      VPTE(JCOL) = (0.,0.)
      IDC = JCOL
      DEFLTE(JCOL) = DEFSL(1,IDC) + DEFSL(2,IDC)*B1*(FEXLOC(JCOL)-1.0)
  270 CONTINUE
C
C
C          LOOP ON ROWS OF THE BOX PATTERN
      IF(NP.EQ.2) GO TO 275
      IRS = 1
```

```
      NXBT = MXBW
      IF(COPLAN) NXBT = MXBT
      NROWS = NXBT
      GO TO 280
  275 CONTINUE
      IRS  = (IXBT-IXBW)/NSUBDV + 1 + IOVLAP
      NXBT = MXBT + IOVLAP
  280 CONTINUE
      DO 560 IROW=IRS,NXBT
C
C           TRANSFER BOX CODES TO CORRECT ARRAYS FOR NEW ROW
      NBXF = NBX
      NBX  = NBXA
      IF (NBXF .EQ. 0) GO TO 315
      DO 310 JCOL = 1,NBXF
  310 IBXCDF(JCOL) = IBXCD(JCOL)
  315 CONTINUE
      IF (NBX .EQ. 0) GO TO 325
      DO 320 JCOL = 1,NBX
  320 IBXCD(JCOL)   = IBXCDA(JCOL)
  325 CONTINUE
      IF (IROW .EQ. NXBT) GO TO 335
      ISROWA = ISROWA + 1
      IF(NP.EQ.2) GO TO 330
      NBXA =IPNTRM(1,ISROWA+1)-IPNTRM(1,ISROWA)+IPNTRM(2,ISROWA)-1
      CALL DCODER(IBOXW,150,ISROWA,1,ISROWA,NBXA,.F.,IBXCDA)
      GO TO 340
  330 CONTINUE
      IDEX = ISROWA + IOVLAP
      NBXA = IPNTRM(1,IDEX+1) - IPNTRM(1,IDEX) + IPNTRM(2,IDEX) - 1
      ISUBT = 2-IXBST
      CALL DCODER(IBOXT(ISUBT,1),LBXCDT,ISROWA,1,ISROWA,NBXA,.F.,IBXCDA)
      GO TO 340
  335 CONTINUE
      NBXA = 0
  340 CONTINUE
C
      ITROW = IROW
      IF(NP.EQ.2) ITROW = IROW -IOVLAP
C
C           LOOP ON CHORDS OF THE BOX PATTERN
      IF (NBX .EQ. 0) GO TO 560
      DO 550 JCOL = 1,NBX
C
      IF (IBXCD(JCOL) .NE. 1) GO TO 550
      IDC = LOCSDW(IROW,JCOL, IPNTRM,LPNTRM,1,LPNTRM)
      IF (IDC .EQ. 0) GO TO 970
C
C           GET THE SUBSCRIPT TO USE IN THE EDGE ARRAYS, JJ
      JJ = JCOL
      IF (NP .EQ. 2) GO TO 350
      IF ( .NOT. COPLAN) GO TO 355
      IF (JJ .GT. MYBT) GO TO 355
      IF (YEXLOC(JJ) .GE. FLOAT(ITROW) ) GO TO 355
  350 JJ = JJ + MYBW
  355 CONTINUE
C
```

```
C        DETERMINE BOX LEADING EDGE VALUES
      IF (FEXLOC(JJ) .GT. FLOAT(ITROW-1) ) GO TO 410
C        BOX LEADING EDGE IS INTERNAL TO THE PLANFORM
      VPLE = VPTE(JCOL)
      DEFLLE = DEFLTE(JCOL)
      GO TO 440
C        BOX IS ON PLANFORM LEADING EDGE. IS IT INFLUENCED BY THE WAKE-
  410 CONTINUE
      IF ( JJ .EQ. JCOL .OR. .NOT. COPLAN) GO TO 420
C
C        LEADING EDGE OF SECOND PLANFORM.  VELOCITY POTENTIAL
C        COMPUTED FROM WAKE EQUATION.
      XDKVL = (FEXLOC(JJ) - TEXLOC(JCOL)) * XKVL
      VPLE  =  TVP(JCOL) * CMPLX(COS(XDKVL),-SIN(XDKVL))
      GO TO 425
C
C        LEADING EDGE OF SECOND PLANFORM OF SPATIAL ANALYSIS
C        OR LEADING EDGE OF FIRST PLANFORM (WING)
  420 CONTINUE
      VPLE = (0.,0.)
  425 CONTINUE
C        TEST FOR SINGLE BOX
      IF (TEXLOC(JJ) .LT. FLOAT(ITROW+1) ) GO TO 430
C        BOX IS A SIMPLE LEADING EDGE BOX
      IDA = LOCSDW(IROW+1,JCOL, IPNTRM,LPNTRM,1,LPNTRM)
      IF (IDA .EQ. 0) GO TO 970
      SLOPE = B1*DEFSL(2,IDC)
      XDIF = FLOAT(ITROW) - FEXLOC(JJ)
      DEFLLE = DEFSL(1,IDC) - SLOPE * XDIF
      GO TO 450
C
C        SINGLE BOX.  GET LEADING AND TRAILING VALUES
  430 CONTINUE
      SLOPE = B1* DEFSL(2,IDC)
      DEFFLE = DEFSL(1,IDC) - SLOPE * (FLOAT(ITROW) - FEXLOC(JJ))
      VPTE(JCOL) = TVP(JJ)
      DEFLTE(JCOL) = DEFSL(1,IDC) + SLOPE * (TEXLOC(JJ) - FLOAT(ITROW))
      GO TO 500
C
C        DETERMINE BOX TRAILING EDGE VALUES
  440 CONTINUE
      IF (TEXLOC(JJ) .LT. FLOAT(ITROW+1) ) GO TO 460
C
C        BOX TRAILING EDGE IS INTERNAL TO THE PLANFORM
      IDA = LOCSDW(IROW+1,JCOL, IPNTRM,LPNTRM, 1, LPNTRM)
  450 CONTINUE
      VPTE(JCOL) = .5 * (DELPHI(IDC) + DELPHI(IDA))
      DEFLTE(JCOL) = 0.5*(DEFSL(1,IDC) + DEFSL(1,IDA))
      GO TO 500
C
C        BOX IS ON SURFACE TRAILING EDGE
  460 CONTINUE
      VPTE(JCOL) = TVP(JJ)
      SLOPE = B1*DEFSL(2,IDC)
      DEFLTE(JCOL) = DEFSL(1,IDC) + SLOPE * (TEXLOC(JJ) - FLOAT(ITROW))
C
C        BOX LEADING AND TRAILING EDGE VALUES ARE COMPUTED.  GET
```

```
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 = JCOL*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 * (XKVL*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*(VPTE(JCOL)-VPLE))*TWOBET* TSLFN(IDC)      FORCES  00458
      DELCP(IDC) = BXLIFT(IDC)/(ALPH*B1)                                     FORCES  00459
  530 CONTINUE                                                               FORCES  00460
C                                                                            FORCES  00461
      TEMP2 = DEFLTE(JCOL)*VPTE(JCOL) - DEFLLE* VPLE                         FORCES  00462
      TEMP3 = (B1*ALPH*DEFSL(2,IDC)) * DELPHI(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 PLANFORMS                                  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) = XKVL                                                         FORCES  00478
      PARM(2) = B1                                                           FORCES  00479
```

```
      PARM(3) = XMACH
C
C         IF THIS IS THE FIRST WEIGHTING FUNCTION, IT MAY BE NECESSARY
C         TO WRITE AND/OR PRINT BOX LIFTS, ETC
      IF (.NOT. BLNEED) GO TO 600
      IF(NM.NE.1) GO TO 600
C
      IF (.NOT. PRBL) GO TO 570
C         PRINT BOX LIFTS
      TITL(1) = 8H    WING
      TITL(2) = 10HBOX LIFTS
      TITL(3) = 2H
      IF(COPLAN) TITL(1) = 10HWING/TAIL
      CALL PRNTBL(TITL,JVP,BXLIFT,1,NROWS,MYBW,IPNTRM)
      IF(NSURF.EQ.1 .OR. COPLAN) GO TO 570
      TITL(1) = 8H    TAIL
      CALL PRNTBL(TITL,JVP,BXLIFT,IXBUT,MXBT,MYBT,IPNTRM(1,IOVLAP+1) )
  570 CONTINUE
C
      IF (.NOT.PRDCP) GO TO 572
C
C         PRINT PRESSURE DIFFERENTIAL
      TITL(1) =  8H    WING
      TITL(2) = 10HPRESS.  DI
      TITL(3) = 10HFFERENCE
      IF(COPLAN) TITL(1) = 10HWING/TAIL
      CALL PRNTBL(TITL,JVP,DELCP,1,NROWS, MYBW,IPNTRM)
      IF (NSURF.EQ.1 .OR. COPLAN) GO TO 572
      TITL(1) = 8H    TAIL
      CALL PRNTBL(TITL,JVP,DELCP,IXBUT,MXBT,MYBT,IPNTRM(1,IOVLAP+1) )
  572 CONTINUE
C
      IF(.NOT. (WTBL.OR.PRSL)) GO TO 600
C
C*********************************************************************
C         EXPAND BOX LIFTS FOR WRITING ON TAPE. WRTETP FORMAT ONLY
      IF (MXWRIT) WRITE (NT6,9999)
C         INITIALIZE COUNTERS FOR PASSING OVER ARRAY BACKWARDS
C               IJKL  = CURRENT LOCATION IN INPUT (COMPRESSED) ARRAY
C               IJ    = CURRENT LOCATION IN OUTPUT (EXPANDED) ARRAY
C               IJFST = FIRST LOCATION FOR CURRENT ROW IN INPUT ARRAY
C               IJPRV = FIRST LOCATION FOR PREVIOUS ROW IN INPUT ARRAY
C
      IROW  = MXB
      IJFST = IPNTRM(1,MXB)
      LOCFST = IPNTRM(2,MXB)
      IJPRV = IPNTRM(1,MXB+1)
      IJKL  = IJPRV
      IJ    = NBOXES - MYB + IJPRV-IJFST + LOCFST
      DO 575 I = 1,NBOXES
      RWBF(I) = (0.,0.)
  575 CONTINUE
C
C          LOOP BACK HERE ON ROWS,  AND ON CHORDS WITHIN A ROW
  580 CONTINUE
      IJ = IJ - 1
      IJKL = IJKL - 1
```

```
      BL = BXLIFT(IJKL)                                                 FORCES  00537
      BXLIFT(IJKL) = (0.,0.)                                            FORCES  00538
      RWBF(IJ)= BL                                                      FORCES  00539
      IF (IJKL .GT. IJFST) GO TO 580                                    FORCES  00540
C          END OF LOOP ON CHORDS WITHIN ONE ROW.  STEP TO NEXT ROW      FORCES  00541
      IJPRV = IJFST                                                     FORCES  00542
      LOCPV = LOCFST                                                    FORCES  00543
C          LOOP BACK HERE ON EMPTY ROWS (COPLANAR CASE)                 FORCES  00544
  582 CONTINUE                                                          FORCES  00545
      IROW = IROW - 1                                                   FORCES  00546
      IJ = IJ - MYB                                                     FORCES  00547
C          DETERMINE WHETHER DONE  -                                    FORCES  00548
      IF (IROW .EQ. 0) GO TO 584                                        FORCES  00549
C          IS THE ROW EMPTY  -                                          FORCES  00550
      IF (IPNTRM(1,IROW) .EQ. IJPRV) GO TO 582                          FORCES  00551
      IJFST  = IPNTRM(1,IROW)                                           FORCES  00552
      LOCFST = IPNTRM(2,IROW)                                           FORCES  00553
      IJ  = IJ - LOCPV  + IJPRV-IJFST + LOCFST                          FORCES  00554
      GO TO 580                                                         FORCES  00555
C            END OF LOOP ON ROWS.                                       FORCES  00556
C*****************************************************************      FORCES  00557
C                                                                       FORCES  00558
  584 CONTINUE                                                          FORCES  00559
C          ARE THE SECTION LIFTS TO BE PRINTED OR WRITTEN -             FORCES  00560
      IF(.NOT. PRSL) GO TO 595                                          FORCES  00561
C          COMPUTE SECTION LIFTS                                        FORCES  00562
      TLIFT = (0.,0.)                                                   FORCES  00563
      TLIFT1 = (0.,0.)                                                  FORCES  00564
      TLIFT2 = (0.,0.)                                                  FORCES  00565
      DO 590 JCOL = 1,MYB                                               FORCES  00566
      BL = (0.,0.)                                                      FORCES  00567
      BL2 = (0.,0.)                                                     FORCES  00568
      IROW = 0                                                          FORCES  00569
      DO 587 IJ = JCOL,NBOXES,MYB                                       FORCES  00570
      IROW = IROW + 1                                                   FORCES  00571
      IF(IROW.GT.TEXLOC(JCOL))  GO TO 586                               FORCES  00572
      BL = BL + RWBF(IJ)                                                FORCES  00573
      GO TO 587                                                         FORCES  00574
  586 BL2 = BL2 + RWBF(IJ)                                              FORCES  00575
  587 CONTINUE                                                          FORCES  00576
      TLIFT1 = TLIFT1 + BL                                              FORCES  00577
      TLIFT2 = TLIFT2 + BL2                                             FORCES  00578
      SLIFT(JCOL+MYBW) = BL2                                            FORCES  00579
      SLIFT(JCOL) = BL                                                  FORCES  00580
  590 CONTINUE                                                          FORCES  00581
C                                                                       FORCES  00582
      IF (.NOT. PRSL ) GO TO 595                                        FORCES  00583
C          PRINT SECTION LIFTS AND TOTAL LIFT                           FORCES  00584
      CALL PRNTSL(JVP,SLIFT,TLIFT1,TLIFT2,MYBW,MYBT )                   FORCES  00585
C                                                                       FORCES  00586
  595 CONTINUE                                                          FORCES  00587
      IF(.NOT.PRCM) GO TO 599                                           BCSFRB  00023
      DO 596 I=1,NTVPS                                                  BCSFRB  00024
      SECHOM(I) = SECHOM(I) * TWOBET                                    BCSFRB  00025
  596 CONTINUE                                                          BCSFRB  00026
      NCM = -JVP                                                        BCSFRB  00027
      CALL PRNTSL(NCM,SECHOM,TLIFT1,TLIFT2,MYBW,MYBT)                   BCSFRB  00028
```

```
C
  599 CONTINUE
      IF (.NOT. WTBL) GO TO 600
C         WRITE BOX LIFTS ONTO THE BINARY OUTPUT FILE
      ITYPE = 7HCOMPLEX
      M = MXB
      K = - M
      N = MYB
      MXARRY = 7HBXLIFTS
      ID(2) = 1000000*IKVAL + JVP
      CALL WRTEMX( NOUTP, MXWRIT,RANDOU, NFS, NMS,LS, NMR,LWB, K, ID,
     1             RWBF,ITYPE, M, N, FARM, IRR )
      IF (IRR .NE. 0) GO TO 922
C
  600 CONTINUE
C         END OF LOOP ON VELOCITY POTENTIALS
C
C         STORE THE ROW OF GENERALIZED AIRFORCES INTO THE FULL MATRIX
      IJ = NM
      DO 620 JVP = 1,NVPS
      GENAF(IJ) = AFROW(JVP)*TWOBET
      GPAFC(IJ) = BS3BET * REAL(GENAF(IJ))
      GPPAFC(IJ) = 0.
      IF(XKVL.NE.0.) GPPAFC(IJ) = BKS4BT * AIMAG(GENAF(IJ))
      IJ = IJ + NMODES
  620 CONTINUE
  650 CONTINUE
C
C
      IF(.NOT.WTGNAF) GO TO 670
      K = NMODES
      M = NMODES
      N = NMODES
      ID(2) = IKVAL
      CALL WRTEMX( NOUTP, MXWRIT,RANDOU, NFS,NMS,LS, NMR,LWB, K, ID,
     1    GENAF,   ITYPE, M,N, FARM, IRR)
      IF (IRR .NE. 0) GO TO 928
C
C         ARE THE FORCES TO BE PRINTED -
  670 CONTINUE
      IF (.NOT. PRGNAF) GO TO 700
C
      CALL PRNTAF(GENAF,PRGNAC,GPAFC,GPPAFC)
  700 CONTINUE
C
      REWIND MODESC
      NMSPCE = 1
  750 CONTINUE
C
      RETURN
C
C         DIAGNOSTICS - ALL CALL FLUSH
C
C                 READING FROM SCRATCH FILE
 8010 CONTINUE
      WRITE (NT6,9100) IGEOSC
      GO TO 950
```

```
 6020 CONTINUE
      WRITE (NT6,9120) MODESC
      GO TO 950
 6030 CONTINUE
      WRITE (NT6,9180) IVPSC
      GO TO 950
C
 6040 CONTINUE
      WRITE (NT6,9140) ITSLSC
      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(MXWRIT) GO TO 990
      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,LWS
      GO TO 964
  962 WRITE (NT6,9620) ID(1),ID(2)
      WRITE (NT6,9622)  FARM,FARM
      WRITE (NT6,9624)  NFS,NMS
  964 WRITE (NT6,9640) ITYPE,M,N
      WRITE (NT6,9650) MXARRY
      GO TO 990
  970 CONTINUE
      WRITE (NT6,9700) IROW, JCOL
      GO TO 990
C
  990 CC  INUE
      WRITE (NT6,9900)
C
      CALL FLUSH(1)
C
```

```
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 XINOFF  /2* 6000000000002003777777B /                           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(1H0,   48H*** MATRIX READ ERROR.  THE  M  DIMENSION SHOULD    FORCES  00715
     1   4H BE ,I2, 4H ***)                                                FORCES  00716
 9500 FORMAT(16H0 *** ERROR CODE ,I5, 28H WHILE READING THE FOLLOWING      FORCES  00717
     1  11H MATRIX *** )                                                   FORCES  00718
 9520 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) = O2,3H, (,O2,1H) )   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 - *,A10, *, 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    16HSNARK FORMAT *** )                                            FORCES  00733
C                                                                          FORCES  00734
      END                                                                  FORCES  00735
```

```
      SUBROUTINE DCODER(IBOX,LBOX, IA,JA,  IL,JL,  SUBD, ICODE)
      DIMENSION  IBOX(LBOX,1), ICODE(1)
C
C         IBOX   - ARRAY OF BOX CODES IN PACKED WORD FORMAT
C         LBOX   - ROW DIMENSION OF BOX CODES ARRAY
C         IA     - I-TH INDEX OF FIRST CODE TO RETRIEVE
C         JA     - J-TH INDEX OF FIRST CODE TO RETRIEVE
C         IL     -  LAST BOX CODE ON THE JA-TH CHORD TO RETRIEVE
C         JL     -  LAST BOX ON THE IA-TH ROW TO RETRIEVE
C         SUBD   - .T., SUBDIVIDED BOX CODES DESIRED, .F. UNSUBDIVIDED.
C         ICODE  - ARRAY INTO WHICH BOX CODE WILL BE STORED.
C
C         COMMENT ON USAGE
C             BOX CODES CAN BE RETRIEVED FOR ONE BOX, A ROW OR PART OF
C         A ROW, OR A COLUMN OR PART OF A COLUMN. A ROW AND COLUMN CAN
C         NOT BE RETRIEVED AT THE SAME TIME. IF ONLY 1 BOX IS DESIRED
C         SET  IL = IA AND  JL = JA.   IF BOTH  IL .NE. IA AND  JL .NE.
C         JA, ONE ROW WILL BE RETURNED,  IL BEING IGNORED.
C
      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
      LOGICAL SUBD
      INTEGER SHIFT
      DATA NBWRD /20/
      MASK = 7
      IB = 1
      IF (SUBD) GO TO 50
      I = NSUBDV * (IA-1) + IXBW
      J = NSUBDV * (JA-1) + NSUBCN
      ISKIP = NSUBDV
      IEND = NSUBDV * (IL-1) + IXBW
      JEND = NSUBDV * (JL-1) + NSUBCN
      GO TO 60
   50 CONTINUE
      I = IA
      J = JA
      ISKIP = 1
      IEND = IL
      JEND = JL
   60 CONTINUE
      IF (JL  .EQ. JA) GO TO 1100
C
C          PROGRAM WILL RETRIEVE NI BOXES FROM ROW I
  100 CONTINUE
      DO 1000 JJ = J,JEND,ISKIP
      JSB =(JJ-1)/NBWRD + 1
      IJWORD = IBOX(I,JSB)
      JB = (NBWRD - MOD(JJ,NBWRD) ) * 3
      IF(JB.EQ.60) JB = 0
C         JB = NUMBER OF BITS TO SHIFT LEFT.
      IJMASK= SHIFT(MASK,JB)
      IJCODE  =   IJWORD.AND.IJMASK
      NJB = -JB
      ICODE(IB) = SHIFT(IJCODE,NJB)
```

```
      IB = IB + 1                                                       DCODAF  00055
 1000 CONTINUE                                                          DCODAF  00056
      GO TO 3000                                                        DCODAF  00057
C                                                                       DCODAF  00058
C          PROGRAM WILL RETRIEVE NJ BOXES FROM CHORD J                  DCODAF  00059
 1100 CONTINUE                                                          DCODAF  00060
      JSB  = (J-1)/NBWRD + 1                                            DCODAF  00061
      JB = (NBWRD - MOD(J,NBWRD) ) * 3                                  DCODAF  00062
      IF(JB.EQ.60) JB = 0                                               DCODAF  00063
      IJMASK = SHIFT(MASK,JB)                                           DCODAF  00064
      NJB = -JB                                                         DCODAF  00065
      DO 2000 II = I,IEND,ISKIP                                         DCODAF  00066
      IJWORD = IBOX(II,JSB)                                             DCODAF  00067
      IJCODE = IJWORD.AND.IJMASK                                        DCODAF  00068
      ICODE(IB) = SHIFT(IJCODE,NJB)                                     DCODAF  00069
      IB = IB + 1                                                       DCODAF  00070
 2000 CONTINUE                                                          DCODAF  00071
C                                                                       DCODAF  00072
 3000 CONTINUE                                                          DCODAF  00073
      RETURN                                                            DCODAF  00074
      END                                                               DCODAF  00075
```

```
      SUBROUTINE PRNTBL(TITL,IMODE,ARRAY,IXB,MXB,MYB,IPNTRM)            PRNTBL 00002
C                                                                       PRNTBL 00003
C      PRINTS BOX LIFTS.  USES /RWBUFF/ FOR INTERMEDIATE SCRATCH        PRNTBL 00004
C         IMODE  - MODE SHAPE NUMBER                                    PRNTBL 00005
C         ARRAY  - ARRAY TO BE PRINTED                                  PRNTBL 00006
C         IPNTRM - POINTER ARRAY FOR ROWS IN -ARRAY-                    PRNTBL 00007
C         NPNTRS - NUMBER OF POINTERS                                   PRNTBL 00008
C                                                                       PRNTBL 00009
      COMPLEX ARRAY(1)                                                  PRNTBL 00010
      DIMENSION  TITL(3)                                                PRNTBL 00011
      DIMENSION IPNTRM(2,50)                                            PRNTBL 00012
C                                                                       PRNTBL 00013
      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 /KVAL  / IKVAL,XKVAL(20), XKS(20)                          KVAL   00002
      COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSPAIC,NOUTP,         FILES  00002
     1               IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC           FILES  00003
C                                                                       PRNTBL 00019
      INTEGER PAGE                                                      PRNTBL 00020
      DIMENSION S(1),D(1)                                               PRNTBL 00021
      EQUIVALENCE (S,BUFF),(D,BUFF(1251))                               PRNTBL 00022
      COMPLEX TLIFT                                                     PRNTBL 00023
      COMPLEX  TLIFT1,TLIFT2                                            PRNTBL 00024
      DIMENSION PC(2), IPNT(2)                                          PRNTBL 00025
      EQUIVALENCE (IPNT,  TLIFT)                                        PRNTBL 00027
      COMMON /RWBUFF/ BFCODE,IBFCNT,  BUFF(3280)                        RWBUFF 00002
      DATA  PC / 10HPAGE CONTI,4HNUED /                                 FTNXI  00086
      DATA BLANK /1H /                                                  PRNTBL 000_3
      DATA  XINIT / -1.0 /                                              PRNTBL 00029
      DATA LINEMX /50 /                                                 PRNTBL 00030
C                                                                       PRNTBL 00031
C                                                                       PRNTBL 00032
      XKVL = XKVAL(IKVAL)                                               PRNTBL 00033
      IF(XKS(IKVAL).NE.XINIT) XKVL = XKS(IKVAL)                         PRNTBL 00034
      PAGE = 0                                                          PRNTBL 00035
      N =1                                                              PRNTBL 00036
      M =4                                                              PRNTBL 00037
      IF(M.GT.MYB) M = MYB                                              PRNTBL 00038
C                                                                       PRNTBL 00039
  100 LINE = 100                                                        PRNTBL 00040
  200 DO 1400 I=IXB,MXB                                                 PRNTBL 00041
      DO 300  J=N,M                                                     PRNTBL 00042
      S(J) = 0.0                                                        PRNTBL 00043
      D(J) = 0.0                                                        PRNTBL 00044
  300 CONTINUE                                                          PRNTBL 00045
      IF(LINE.LE.50) GO TO 900                                          PRNTBL 00046
      PAGE = PAGE + 1                                                   PRNTBL 00047
      LINE = 4                                                          PRNTBL 00048
      WRITE (NT6,9001) TITLE,TITL,XMACH,XKVL,IMODE                      PRNTBL 00049
      IF(PAGE.EQ.1) GO TO 700                                           PRNTBL 00050
      WRITE(NT6,9005) PC                                                PRNTBL 00051
      GO TO 800                                                         PRNTBL 00052
  700 WRITE(NT6,9005)                                                   PRNTBL 00053
```

```
  800 CONTINUE
      WRITE(NT6,6006) (BLANK,J,J=N,M)
      WRITE(NT6,6007) (BLANK,  J=N,M)
C
  900 CONTINUE
      JS = IPNTRM(2,I)
      IF(JS.LE.0) GO TO 1400
      IDX = IPNTRM(1,I)
      JE = IPNTRM(1,I+1) - IDX  + JS-1
      IF(JE.EQ.0) GO TO 1400
      DO 1000 J=JS,JE
      S(J) =  REAL(ARRAY(IDX))
      D(J) = AIMAG(ARRAY(IDX))
      IDX = IDX + 1
 1000 CONTINUE
      DO 1200 J =N,M
      IF(S(J)) 1300,1100,1300
 1100 CONTINUE
      IF(D(J)) 1300,1200,1300
 1200 CONTINUE
      GO TO 1400
 1300 WRITE(NT6,9013) I,(S(J),D(J),J=N,M)
      LINE = LINE + 1
 1400 CONTINUE
C
      M = M+4
      N = N+4
      IF(N.GT.MYB) GO TO 1500
      IF(M.GT.MYB) M = MYB
      IF(LINE.GT.45) GO TO 100
      WRITE(NT6,6006) (BLANK,J,J=N,M)
      WRITE(NT6,6007) (BLANK,  J=N,M)
      LINE = LINE+3
      GO TO 200
 1500 CONTINUE
      RETURN
 9001 FORMAT(1H1,20X,8A10,/50X,3A10,/ 46X,7H( MACH F5.3,5X,10HRED.FREQ.
     1    *=*,F8.5, * )* /52X,*MODE   SHAPE*, I3 )
 9005 FORMAT(44X,42(1H-),20X,A10,A4 )
 6006 FORMAT(4H0ROW,A1,14X,5HCHORD,I3,3(A1,22X,5HCHORD,I3))
 6007 FORMAT(3X, 4(A1,9X,4HREAL,8X,9HIMAGINARY) )
 9013 FORMAT(I4,8E16.8)
      END
```

```
      SUBROUTINE PRNTSL(IMODE,SLIFT,TLIFT1,TLIFT2,MYBW,MYBT)
C
C        PRINTS THE SECTION LIFTS AND TOTAL LIFTS
C
C        IMODE  - MODE SHAPE NUMBER
C        SLIFT  - SECTION LIFT ARRAY
C        TLIFT1 - WING TOTAL LIFT
C        TLIFT2 - TAIL TOTAL LIFT
C
C        IF IMODE IS NEGATIVE THE PROGRAM WILL OUTPUT SECTION MOMENTS
C
C
      COMMON /PROBLM/ XMACH,NMODES,NTSLOP,NKVALS,SMOOTH,NDEG,CRDFIT,
     1                EXAIC,SUBDV,PLYWOOD
      LOGICAL         SMOOTH,CRDFIT,EXAIC,SUBDV,PLYWOOD
      COMMON /CONTRL/ PREVEX,OMACH, TITLE(8), PRVGEOM,PRVMODE,DIHW,DIHT,
     1               DEFAULT
      LOGICAL     PRVGEOM,PRVMODE,DIHW,DIHT,DEFAULT
      COMMON /KVAL  / IKVAL,XKVAL(20), XKS(20)
      COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NSFAIC,NOUTP,
     1                IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC
C
      COMPLEX  SLIFT(1)
      COMPLEX  TLIFT1,TLIFT2,TLIFT
      LOGICAL PRCM
      DATA    BLANK/1H /
      DATA  XINIT / -1.0 /
C
      IF(IMODE.LT.0) GO TO 100
      PRCM = .FALSE.
      GO TO 200
  100 CONTINUE
      PRCM = .TRUE.
      IMODE = -IMODE
  200 CONTINUE
C
      XKVL = XKVAL(IKVAL)
      IF(XKS(IKVAL).NE.XINIT) XKVL = XKS(IKVAL)
      IF(PRCM) GO TO 300
      WRITE (NT6,6010) XMACH,XKVL,IMODE
      GO TO 400
  300 CONTINUE
      WRITE (NT6,9010) XMACH,XKVL,IMODE
  400 CONTINUE
      WRITE (NT6,6008)
      WRITE (NT6,6005)
      WRITE (NT6,6020)
      WRITE (NT6,6007) BLANK,BLANK,BLANK,BLANK
      DO 600 I=1,MYBW,4
      II = I + 3
      IF (II .GT. MYBW) II = MYBW
      WRITE (NT6,6030) I,(SLIFT(IJ),IJ=I,II)
  600 CONTINUE
      IF(PRCM) GO TO 650
      WRITE (NT6,6023) TLIFT1
      IF(MYBT.EQ.0) GO TO 800
C
```

```
      WRITE (NT6,6009)                                                  FRNTBL  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)                                                  FRNTBL  00131
      WRITE (NT6,6007) BLANK,BLANK,BLANK,BLANK                          FRNTBL  00132
      DO 700 I=1,MYBT,4                                                 FRNTBL  00133
      II = I + 3                                                        FRNTBL  00134
      IF (II .GT. MYBT) II = MYBT                                       FRNTBL  00135
      I2 = I + MYBW                                                     FRNTBL  00136
      II2 = II + MYBW                                                   FRNTBL  00137
      WRITE (NT6,6030) I,(SLIFT(IJ),IJ=I2,II2)                          FRNTBL  00138
  700 CONTINUE                                                          FRNTBL  00139
C                                                                       FRNTBL  00140
      IF(FRCM) GO TO 800                                                BCSFRB  00055
      WRITE (NT6,6024) TLIFT2                                           FRNTBL  00141
      TLIFT = TLIFT1 + TLIFT2                                           FRNTBL  00142
      WRITE (NT6,6025) TLIFT                                            FRNTBL  00143
  800 CONTINUE                                                          BCSFRB  00056
      RETURN                                                            FRNTBL  00144
C                                                                       FRNTBL  00145
C                                                                       FRNTBL  00146
 6005 FORMAT(44X,32(1H-),30X,A10,A4)                                    FRNTBL  00147
 6007 FORMAT(3X, 4(A1,9X,4HREAL,8X,9HIMAGINARY) )                       FRNTBL  00148
 6008 FORMAT(58X,*WING* )                                               FRNTBL  00149
 6009 FORMAT(1H0,// 53X,*SECTION LIFTS* / 58X,*TAIL*,/ 44X,32(1H-) / )  FRNTBL  00150
 6010 FORMAT(1H0,52X, 14HSECTION  LIFTS        /44X,*( MACH *,F5.3,5X,  FRNTBL  00151
     1 *RED. FREQ. =* F8.5,* )* / 52X,*MODE   SHAPE*,I3)                FRNTBL  00152
 6020 FORMAT(6HOCHORD)                                                  FRNTBL  00153
 6023 FORMAT(1H0,44X,* TOTAL LIFT  -  WING * / 1H0 40X,2E16.8 )         FRNTBL  00154
 6024 FORMAT(1H0,44X,* TOTAL LIFT  -  TAIL * / 1H0 40X,2E16.8 )         FRNTBL  00155
 6025 FORMAT(1H0,53X,* TOTAL LIFT */ 1H0,40X,2E16.8 )                   FRNTBL  00156
 6030 FORMAT(I4,8E16.8)                                                 FRNTBL  00157
 9009 FORMAT(1H0//   45X,*SECTIONAL MOMENT COEFFICIENTS*/58X,*TAIL*/    BCSFRB  00057
     1      44X, 32(1H-) / )                                            BCSFRB  00058
 9010 FORMAT(1H0////,45X,*SECTIONAL MOMENT COEFFICIENTS*/44X,*(MACH *,  BCSFRB  00059
     1        F5.3,5X,*RED. FREQ. =* F8.5,* )* /52X,* MODE   SHAPE*,I3) BCSFRB  00060
      END                                                               FRNTBL  00158
```

```
      SUBROUTINE PRNTAF(ARRAY,PRGNAC,GPAFC,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         GPAFC  - AGARD GENERALIZED AERODYNAMIC COEFFICIENT            PRNTAF  00008
C         GPPAFC - AGARD GENERALIZED AERODYNAMIC COEFFICIENT            PRNTAF  00009
C                                                                       PRNTAF  00010
      COMPLEX ARRAY(1)                                                  PRNTAF  00011
      DIMENSION  GPAFC(1),GPPAFC(1)                                     PRNTAF  00012
      LOGICAL    PRGNAC                                                 PRNTAF  00013
C                                                                       PRNTAF  00014
      DIMENSION PC(2)                                                   PRNTAF  00015
      COMMON /PRCBLM/ XMACH,NMODES,NTSLOP,NKVALS,SMOOTH,NDEG,CRDFIT,    PRCBLM  00002
     1                EXAIC,SUBDV,PLYWOOD                               PRCBLM  00003
      LOGICAL         SMOOTH,CRDFIT,EXAIC,SUBDV,PLYWOOD                 PRCBLM  00004
      COMMON /KVAL  / IKVAL, XKVAL(20), XKS(20)                         KVAL    00002
      COMMON /FILES / NT5,NT6,INTAPE,IAFSP,NFLAIC,NSFAIC,NOUTF,         FILES   00002
     1                IOUFSP,MODESC,IVPSC,IGEOSC,IWTFSC,IAICSC          FILES   00003
      INTEGER PAGE                                                      PRNTAF  00022
      DATA    PC     / 10HPAGE CONTI,4HNUED /                           FTNXX   00087
      DATA    BLANK / 1H  /                                             FTNXX   00088
      DATA    XINIT / -1.0 /                                            FTNXX   00089
      XKVL = XKVAL(IKVAL)                                               PRNTAF  00023
      IF(XKS(IKVAL).NE.XINIT) XKVL = XKS(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 = 8                                                          PRNTAF  00037
      WRITE (NT6,6001) XMACH,XKVL                                       PRNTAF  00038
      IF (PAGE .EQ. 1) GO TO 150                                        PRNTAF  00039
      WRITE (NT6,6005) PC                                               PRNTAF  00040
      GO TO 180                                                         PRNTAF  00041
  150 WRITE (NT6,6005)                                                  PRNTAF  00042
  180 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
C           PRINT THE GENERALIZED AERODYNAMIC COEFFICIENTS
C           IF DESIRED.
C
      IF(.NOT.PRGNAC) 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,XKVL
      IF(IPR.EQ.2) GO TO 1140
      IF(PAGE.EQ.1) GO TO 1130
      WRITE (NT6,7005) PC
      GO TO 1160
 1130 CONTINUE
      WRITE(NT6,7005)
      GO TO 1160
 1140 CONTINUE
      IF (PAGE.EQ.1) GO TO 1150
      WRITE (NT6,7015) PC
      GO TO 1160
 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
```

```
 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 (NT6,7006)                                               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
 6001 FORMAT(1H1,51X,18HGENERALIZED FORCES /44X,6H(MACH ,F5.3,5X,*RED. F PRNTAF  00130
     1REQ. =* F8.5,1H) )                                             PRNTAF  00131
 6005 FORMAT(35X,50(1H-),20X,A10,A4)                                 PRNTAF  00132
 6006 FORMAT(5H0 WT. / 6H FUNCT, 4(A2,23HVELOCITY POTENTIAL MODE,I3,3X)) PRNTAF  00133
 6007 FORMAT(2X,4(A10,4HREAL,8X, 9HIMAGINARY) )                      PRNTAF  00134
 6010 FORMAT( I4, 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*,/ )   PRNTAF  00138
 7007 FORMAT(I13,7I16)                                               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)
C
C        RETURNS THE LOCATION OF THE WORD IN THE END-AROUND SUBDIVIDED
C        DOWNWASH ARRAY CORRESPONDING TO BOX(IROW,JCOL) OF THE SUB-
C        DIVIDED BOX ARRAY
C
C              IROW   = BOX CHORDWISE LOCATION
C              JCOL   = BOX SPANWISE LOCATION
C              IPNTSD = ARRAY OF POINTERS
C              IPNTIN = NEXT AVAILABLE (UNUSED) CELL IN IPNTSD (END-
C                       AROUND)
C              IPNTOT = FIRST CURRENTLY AVAILABLE CELL IN IPNTSD
C              IPNTLS = LAST CELL OF IPNTSD (LENGTH OF ARRAY)
C        RETURN -
C              LOCSDW = LOCATION OF DESIRED DOWNWASH, IF SUCCESSFUL
C                     = 0, IF LOCPNT LIES OUTSIDE THE DEFINED AREA.
C
      DIMENSION IPNTSD(2,IPNTLS)
C
      LOCPNT = MOD(IROW-1,IPNTLS) + 1
C              LOCPNT = LOCATION OF CELL IN IPNTSD WHICH WAS OR IS TO BE
C                       USED
      IF(IPNTIN - IPNTOT) 100, 300, 200
C          END AROUND HAS OCCURRED
  100 IF (LOCPNT - IPNTIN) 400, 300, 150
C          NOT IN UPPER PART. IS LOCPNT WITHIN BOTTOM PART -
  150 IF (LOCPNT - IPNTOT) 300, 400, 400
C
C          NO END AROUND, NORMAL SEQUENCE
  200 IF (LOCPNT - IPNTIN) 250, 300, 300
C          LESS THAN UPPER LIMIT. IS LOCPNT .GE. LOWER LIMIT -
  250 IF (LOCPNT .GE. IPNTOT) GO TO 400
C
C          ERROR OR INITIAL CONDITION ENCOUNTERED (SHOULD NEVER OCCUR)
  300 LOCSDW = 0
      GO TO 500
C
C          SUCCESSFUL, BOX HAS BEEN DEFINED
  400 IFB = IPNTSD(2,LOCPNT)
      IF(JCOL.LT.IFB) GO TO 300
      LOCSDW = IPNTSD(1,LOCPNT) + JCOL-IFB
C
  500 CONTINUE
      RETURN
      END
```

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						

☆U.S.Government Printing Office: 1972 — 759-084/453

UNCLASSIFIED
Security Classification