CR-69175

REV LTR

THE BOEING COMPANY

AIRPLANE DIVISION
P.O. BOX 707
RENTON, WASHINGTON 98055

D6-10741, PART II

TITLE: A METHOD OF OPTIMIZING CAMBER SURFACES FOR WING-BODY COMBINATIONS AT SUPERSONIC SPEEDS — DIGITAL COMPUTER PROGRAM DESCRIPTION

44 Nasa

Prepared under NASA Contract

NAS2-2282

PREPARED BY J. E. Brown 8-30-65
A. S. Kawaguchi 8/30/65
Eugene Lo Rowe 8/30/65

SUPERVISED BY [illegible] 8-30-65

APPROVED BY Robert L. Palmer 8/30/65

ECEDING PAGE BLANK NOT FILMED.

CONTENTS

1.	INTRODUCTION	1
2.	DESCRIPTION	3
2.1	Geometry Definition Section Description	4
2.2	Geometry Transformation Section Description	5
2.3	Geometry Paneling Section Description	6
2.4	Aerodynamics Section Description	8
3.	USE OF THE PROGRAM	11
3.1	Machine Components	11
3.2	Program Timing and Output Estimates	11
3.3	Input Data Format	13
3.4	Sample Input and Output Data	34
4.	PROGRAM RECORDS	179
4.1	Flow Charts	179
4.2	Subroutine Descriptions	187
5.	APPENDIXES	415
5.1	Appendix A — Dynamic Storage Allocation Package	415
5.2	Appendix B — Matrix Inversion Package	437
5.3	Appendix C — Tape Formats	442
	Format 1 — Geometry definition tape format	442
	Format 2 — Geometry transformation tape format	446
	Format 3 — Transformation data tape format	448
	Format 4 — Geometry paneling tape format	449
5.4	Appendix D — Subroutine Listings	453
6.	REFERENCES	739

PRECEDING PAGE BLANK NOT FILMED.

1. INTRODUCTION

This part of the document describes the digital computer program developed to solve the problem of optimizing the wing camber surfaces for wing-body combinations at supersonic speeds. In addition, the program will solve the direct problem of calculating the pressures, forces, and moments on a given configuration, and the indirect problem of determining the wing shape for a given lift distribution. The theoretical considerations underlying the method are presented in references 1 and 2.

The complete program consists of four sections: Geometry Definition, Geometry Transformation, Geometry Paneling, and Aerodynamics. The first three sections present, in a form acceptable to the Aerodynamics section, a geometric description of the configuration under study. A wing alone, body alone, or wing-body combination may be defined. Basically, the geometry sections redefine the body as an equivalent body of circular cross section. The wing, and that part of the body in the region aft of the wing leading-edge intersection, are subdivided into a large number of panels. The wing may be defined to have thickness or may consist of a camber surface only. The Aerodynamics section then solves the problem by first computing the velocity components induced by the various singularities that represent the configuration. The body thickness and camber effects are represented by line sources and doublets along the body axis, and wing thickness effects by a distribution of sources located in the wing reference plane. The effects of camber and incidence of the wing, together with any residual interference effects, are simulated by a surface distribution of vortices on the wing and body. Once the velocity components have been computed, the program determines the strengths of the singularities required to satisfy the given boundary conditions. The program selects the boundary conditions appropriate to the particular direct, indirect, or optimization problem being considered, and solves the problem.

This Part of the document is a guide for program maintenance. An outline of the program structure is followed by a detailed description of each section of the program. The section on program use contains the input format; timing

and output estimates, and the required machine components. Input and output formats of a sample case are also included. The section for program records contains flow charts, overlay diagrams, and descriptions of each subroutine. A program listing and other appendixes complete this Part of the document.

2. DESCRIPTION

The digital computer program described in this section has been developed to solve the problem of optimization of wing camber surfaces for wing-body combinations at supersonic speeds. Direct and other indirect aerodynamic problems can also be solved.

The program is coded in FORTRAN IV and MAP languages for the IBM 7090/7094 (32K) digital computer under the Systems Monitor, IBSYS Version 12. It is compatible with the NASA-Ames direct-coupled IBM 7040/7094 computer system. Because the program exceeds the capacity of a single core load, the Loader Overlay feature is used, which allows the complete program to be subdivided into several smaller segments, or links. The links are processed in a specified order to solve a particular problem.

The complete program consists of four sections: Geometry Definition, Geometry Transformation, Geometry Paneling, and Aerodynamics (see section 4.1). The first two sections provide a suitable geometric description of the configuration and the third section subdivides the configuration into panels. The Aerodynamics section performs all aerodynamic calculations and solves the problem under consideration.

The computer is initialized by the subroutine OPCAMI and program execution is controlled by the subroutine OPCAM, a control program located in link 0 under the program Overlay structure. Those control cards within the data deck that determine which program sections are used to process the case, are read by the subroutine OPCAM and lower-level subroutines in each of the following four program sections:

1. GEOMD in link 5 (Geometry Definition section)
2. TFLAT in link 11 (Geometry Transformation section)
3. PANEL in link 12 (Geometry Paneling section)
4. AERO in link 19 (Aerodynamics section)

Multiple cases, each involving a different wing-body configuration, can be run. When a non-systems error condition occurs during processing within a section of the program, an error message appears, execution of the present case is terminated, a partial data printout is given, and the following case is processed. Execution time averages 7 to 8 minutes for a typical body-alone or wing-alone case (100 panels) and 18 to 20 minutes for a wing-body combination of 200 panels.

2.1 Geometry Definition Section

This part of the program defines geometrically a wing, a body, or a wing-body combination. The definition, in the form of body meridian lines and wing percent chord lines, is written on a tape that is read by the Geometry Transformation section of the program

Meridian lines may be visualized as stringers running lengthwise along the surface of the body. Each meridian line consists of a series of straight-line segments connecting points on the body surface; adjacent points are connected by straight lines. Constant percent chord lines, much like meridian lines, run along the wing surface in a spanwise direction.

A body surface is specified by a set of defining sections normal to the X-axis. Points on defining sections are given in polar coordinate form (ρ , θ) or in rectangular coordinates (Y, Z). The number of points in each section must be the same as the number of meridian lines. Since the X-coordinate of each defining section is known, the X, Y, Z-coordinates of each point in a section can be computed. The first meridian line is constructed by joining the first point in each section; succeeding meridian lines are similarly constructed. As an option, the meridian lines may be "enriched"; additional meridian line points are inserted by three-dimensional interpolation from a space curve through the original points.

Both upper and lower wing surfaces may be defined. Only the construction of upper wing percent chord lines will be discussed, because later sections of the program are presently limited to this case. Although more complicated, the procedure is similar to that used to define the body. Percent chord lines (enriched if desired) on the surface of the wing are computed from a given set of control chords. These control chords must be parallel to the X-Y plane but are not necessarily parallel to the X-Z plane. In order to locate each control chord in space, sufficient data are given to find the intersections of the chord with the leading and trailing edges of the given planform. The distance between these intersections is called (in this discussion only) the "oblique chord length," because control chords need not be streamwise.

The projection of wing planform in the X-Y plane is specified by a set of points on the leading and trailing edges; the number of points in each set need not be the same. The planform may extend past the control chord region, but the program will reject any chord which does not intersect both the leading and

trailing edges. The planform may be open or closed at the wing tip. A planform is closed if the leading edge and trailing edge intersect at the wing tip; the generated percent chord lines will then converge to that point. If the leading and trailing edges do not intersect the planform is open.

A control chord is given as a set of points in a two-dimensional coordinate system. These points are scaled (except for zero-length chords) so that the difference in abscissa of the first and last points is equal to the oblique chord length. This scaled chord is then placed on the planform. Wing percent chord lines are constructed between scaled control chords.

The wing-body intersection option calculates the intersection of each wing percent chord line with the body surface. The wing percent lines are then truncated at the body surface.

All variable-length arrays are stored in a single buffer of 10,000 cells. As such there are no individual limits on the number of body stations, number of meridian or percent lines, number of chords or number of points defining the leading and trailing edges of the wing. (However, the data are scanned and excessively large arrays are rejected.) Maximum sizes for data arrays are usually set by later program segments.

2.2 Geometry Transformation Section

This section of the program reads the body and wing definitions from a tape written by the previous section. It then transforms the body and wing to a new coordinate system, orients the wing planform parallel to the x-y plane in this transformed system, and finds the intersections of wing percent chord lines with the body surface. Results are written on tapes for use by later program sections.

Three types of configurations may be handled: body alone, wing alone, or wing-body combination. If a body surface is involved only half of it is given; it is assumed to be symmetrical about the X-Z plane. Then a new x, y, z coordinate system is set up with the origin at the centroid of the forward end of the body and the x-axis passing through the aft end centroid; the z-axis lies in the X-Z plane. In the new coordinate system the body ends (unless of zero radius) are not necessarily parallel to the y-z plane; this is corrected by a

linear adjustment of all x-coordinates of the body points. Also, the new body length (distance between body end centroids) may differ from the original length. The original set of body defining stations is converted to the new coordinate system by a transformation of the form $x = \alpha X + \beta$ where α and β are constants. The body radius and centroid are computed at each of these stations. In a body alone case, the radius and centroid are then found by interpolation at a given number of equally spaced stations.

In a wing-body combination case, the wing is also transformed to the x, y, z coordinate system. Then the intersections of the wing leading and trailing edges with the body surface are computed; let z_A be the average of the z-coordinates of these two points. The wing is effectively flattened by changing each z-coordinate of the wing to z_A. The body is then intersected by the wing percent chord lines of the flattened wing.

The calculation of body radius and centroid is now resumed. A set of equally spaced stations is chosen between the forward end of the body and the station at which the wing leading edge intersects the body; a similar set is selected from there on aft. The calculation of the interval between stations in each region is subject to the constraints that the total number of stations must agree with a given number and that the two intervals must be approximately the same in each region.

In a wing alone case, the transformation is omitted. However, the wing is flattened; z_A is taken as the average of the z-coordinates of the inboard point on the leading and trailing edge percent chord lines. Bodies defined by a maximum of 16 meridan lines of 90 points each and wings of 16 percent chord lines of 90 points each can be processed.

2.3 Geometry Paneling Section

This program section panels the body and/or wing and provides additional data required by the following section (Aerodynamics). Two cases can be given as input to the Geometry Paneling section under the present program Overlay setup: the wing-body case and the wing alone case. The body alone case does not use the paneling section.

For a wing-body case, the body definition that is written on a scratch tape by the previous section (Geometry Transformation) is read into storage by an

input subroutine, INPUTB. The paneling is accomplished by combining this definition with data for the body cutting planes, read from input cards. A body cutting plane is oriented normal to the X-axis. The body panel areas are computed, and the panel centroid and control point coordinates are calculated. Streamwise chords are constructed through the panel control points, and the chord lengths are calculated. Finally the panel θ -inclination and α-incidence angles are computed. The paneling description and the additional geometric data are written on an output tape for printout purposes and on a scratch tape for use in the Aerodynamics section.

The wing paneling that is required for either the wing-body or the wing alone case is obtained similarly. The wing definition written by the Geometry Transformation section is read into storage arrays by an input subroutine, INPUTW, and used with card input data to provide a wing paneling scheme. Again the wing panel areas, chord lengths, and centroid and control point coordinates are calculated. The inclination and incidence angles are defined to be zero for wing panels since the wing is assumed to be a flat plate with no dihedral at zero incidence. As an option, camber and thickness ordinates and slopes are computed from airfoil ordinate tables. The wing paneling data are written on output and scratch tapes.

Certain program restrictions are present. Though a body that closes to a point can be defined in the definition section, such a body can not be paneled. For this case the leading row of panels would consist of triangular, not quadrilateral panels. This difficulty is avoided by excluding body cutting planes that intersect the body at the nose.

A second restriction requires a flat wing for all program cases. Under the current Overlay scheme, however, the Geometry Transformation section always provides such a definition. Hence this restriction results in no difficulty to the user.

When errors occur, an error message appears, execution of the present case is terminated, a partial data printout is given, and the next case is processed.

2.4 Aerodynamics Section

The aerodynamics problem is solved in three stages: (1) geometry input and calculation of the velocity components, and (2) a preprocessing stage, which is preparation for (3) the calculation of the aerodynamic cases. Control of the flow through the main links of this section of the program is provided by the subroutine AERO and is dependent on the configuration (wing alone, body alone, or wing-body combination) used.

The wing is defined by a number of panels. For an isolated body, the definition is given as an equivalent body of circular cross-section. In order to compute the interference effects on the body, for wing-body combinations, the body surface aft of the wing leading-edge intersection is also represented by panels. The geometric definition of the configuration is read from tapes generated by the geometry sections; this information also includes the areas, centroids, and control points of each panel.

Velocity components due to the various singularities representing the configuration are computed and stored on tape. For a wing defined to have thickness, the program computes velocity components induced by sources located in the reference plane of the wing. The effects of body thickness, camber, and incidence are simulated by placing line sources and doublets along the axis of the body. The effects of camber and incidence of the wing, and any residual interference effects, are represented by a surface distribution of vorticity on the wing and body panels. An option is provided to print out velocity components due to the individual singularities.

The preprocessing stage consists of the following computations. The velocity components are reordered and rewritten on tape. Several matrix operations are performed on the matrix of the aerodynamic influence coefficients (the normal components of velocity due to the vortex distributions), and the results of these are also written on tape. In particular, if the matrix is represented by the following sub-matrices:

$$\begin{bmatrix} [A_{BB}] & [A_{BW}] \\ [A_{WB}] & [A_{WW}] \end{bmatrix}$$

where

$[A_{BB}]$ is the influence on the body due to the body
$[A_{WB}]$ is the influence on the wing due to the body
$[A_{BW}]$ is the influence on the body due to the wing
$[A_{WW}]$ is the influence on the wing due to the wing

Then the following matrices are formed:

$$[D] = [A_{WB}] \cdot [A_{BB}]^{-1}$$

$$[E] = [A_{BB}]^{-1} \cdot [A_{BW}]$$

$$[A_R] = [A_{WW}] - \left[[A_{WB}] \cdot [A_{BB}]^{-1} \cdot [A_{BW}]\right]$$

A_R is referred to as the "reduced" matrix of the aerodynamic influence coefficients.

For optimization of the wing camber surface shape, the Lagrange multiplier technique is applied, and the coefficients of the resulting system of equations are computed (see equation (126), Part I):

$$\begin{bmatrix} -2A_1 a_{11} & -(A_1 a_{12} + A_2 a_{21}) \cdots & -A_1 & -x_1 A_1 \\ -(A_2 a_{21} + A_1 a_{12}) & & -A_2 & \cdot \\ \vdots & & \vdots & \vdots \\ & & -A_N & -x_N A_N \\ -A_1 & \cdot\ -A_N & 0 & 0 \\ -x_1 A_1 & \cdot\ -x_1 A_N & 0 & 0 \end{bmatrix} \cdot \begin{Bmatrix} p_{W_1} \\ p_{W_2} \\ \vdots \\ p_{W_N} \\ \lambda_1 \\ \lambda_2 \end{Bmatrix} = \begin{Bmatrix} A_1(\bar{n}_1 + \bar{n}_{B_1}) \\ A_2(\bar{n}_2 + \bar{n}_{B_2}) \\ \vdots \\ A_N(\bar{n}_N + \bar{n}_{B_N}) \\ \bar{L} \\ \bar{M} \end{Bmatrix}$$

This operation results in the formation of two matrices. The first matrix includes wing lift and pitching moment constraints, and the second only the wing lift constraint. For the latter case, the row and column of the matrix corresponding to λ_2 is omitted. Both matrices are inverted and stored on tape.

The aerodynamic cases can now be computed. Additional input cards are used to request the following cases:

1) To compute the wing camber and twist, given the pressure distribution of the wing
2) To compute the pressure distribution, given the geometry of the configuration
3) To optimize the wing shape for minimum drag, given the wing lift or wing lift and pitching moment constraints

In each case a drag polar may be computed by inputing a series of incremental angles of attack. If a wing with thickness is specified in the input geometry, the program provides an option to compute the pressures due to thickness and to add them to the pressures due to camber. Another option selects either the general nonlinear or linear equation for computing pressure coefficients. For a given configuration and Mach number, any number of aerodynamic cases may be computed.

Upon completion, the Aerodynamics section returns to the main link for processing of another configuration.

3. USE OF THE PROGRAM

3.1 Machine Components

The program is coded in FORTRAN IV and MAP for the IBM 7090/7094 (32K) digital computer under the control of the Systems Monitor, IBSYS Version 12. As the program is organized to utilize the Overlay feature of the Loader, IBLDR, one of the system units must be used as the input-output tape on which the links are written. This unit is specified on the $ORIGIN control card according to the procedure outlined in reference 3.

In addition to the input and output tapes, the program uses seven tape units for scratch purposes. The choice of tape units to be used will depend on the particular computer installation, and tapes must be changed as required. Subroutine OPCAMI (see section 4.2) initializes all the tape units by assigning a logical number to each. To make the necessary changes, it will be sufficient to change only the logical designations in this subroutine.

3.2 Program Timing and Output Estimates

The computer time and number of lines of printout for a single configuration can be estimated from the following equations based on experience on the IBM 7094/M2.

$$\text{Time (minutes)} = 2.2 + .3G + (3.8 \cdot 10^{-4} \cdot P^2) \cdot A + .6C,$$

where G indicates type of paneling:

= 0., no paneling

= 1., wing paneling only

= 2., wing and body paneling

P is number of panels. If no paneling is required, use P = 10.

A indicates aerodynamic calculations:

= 0., no aerodynamic calculations

= 1., aerodynamic calculations requested

C is number of aerodynamic cases. Each of the following is considered a case:

Wing optimization case;

Direct aerodynamic case;

Indirect aerodynamic case;

Each angle of a polar series.

$$\text{Output (lines)} = 100 + \left[500 + (10 \cdot P^{1/2}) \cdot V \cdot C\right] \cdot T$$

where T indicates type of case:

= 1., wing-alone case

= 2., body-alone case

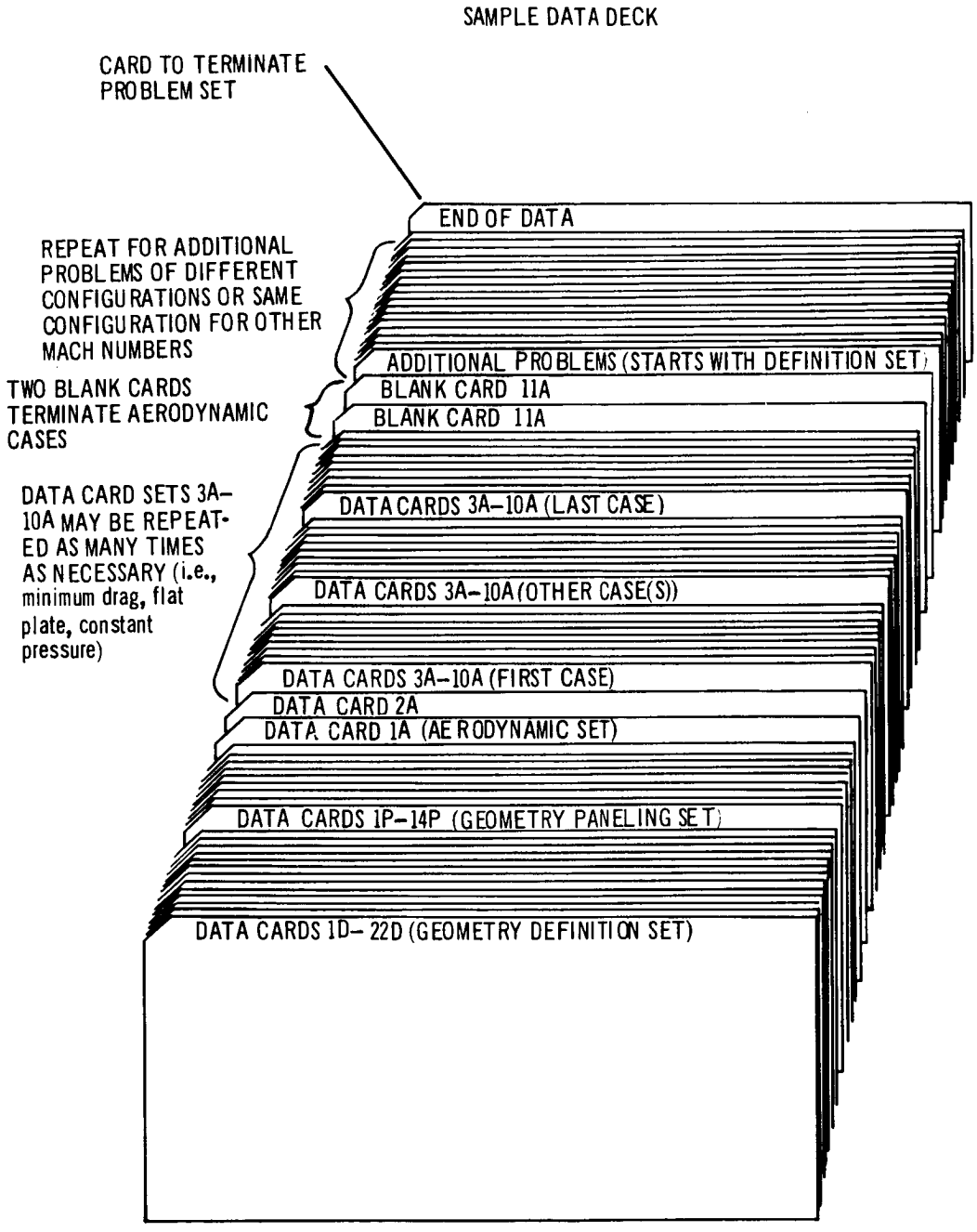
= 3., wing-body case

V indicates velocity component printout:

= 1., no velocity component printout:

= 2., velocity components requested

3.3 Input Data Format



GEOMETRY DEFINITION CARD SET

All geometry definition data, except title cards and literal statements, are punched in six-field, ten-digit format. A decimal point is required in each data field.

For a body-alone problem definition, cards 10D through 19D are omitted.

For a wing-alone problem definition, cards 2D through 6D, 19D, and 20D are omitted.

	Column	Code	Explanation
Card 1D	1-6	DEFINE	Columns 1-6 contain the word DEFINE.
Card 2D	1-4	BODY	Columns 1-4 contain the word BODY. Card 2D is used only when a body or wing-body combination is defined.
Card 3D	1-72	TITLE	Any desired title.
Card 4D	1-10	BNS	Number of defining body stations. $2. \leq$ BNS $\leq 50.$.
	11-20	BTHETA	Number of points on each defining body station, i.e., the number of ρ, θ or Y, Z pairs per station. $2. \leq$ BTHETA $\leq 16.$.
	21-30	AXIS(1)	Y-coordinate of body definition axis.
	31-40	AXIS(2)	Z-coordinate of body definition axis.
	41-50	CHDB	Dimensional tolerance to be used in generating additional body meridian line points between given stations. If CHDB $\leq 0.$ or if BNS $< 4.$, no additional points will be generated. If $0. <$ CHDB < 0.001, then a value of 0.001 will be used.

	Column	Code	Explanation
Card(s) 5D (3 maximum)	1-10	θ_1	Array of angles (θ), in degrees at each defining station. There must be exactly BTHETA angles $\leq$ 16, six per card.
	⋮		
	51-60	θ_6	
	etc.		
Card(s) 6D* (50 maximum)	1-10	STA	X-coordinate of body station.
	11-20	YZ(1)	Δ Y-increment added to body definition axis to establish a local origin from which all ρ, θ for this station are measured.
	21-30	YZ(2)	Δ Z-increment added to body definition axis to establish a local origin from which all ρ, θ for this station are measured.
	31-40	SCODE	= 0. this cross section is identical to previous section.
Note — if options 1, 4, 5, or 6 are designated, the added information card(s) 7D, 8D, or 9D must be inserted behind that station card 6D and before the next station card 6D.			= 1. this cross section is specified by BTHETA values of ρ (on cards 7D). The θ-array of card(s) 5D will be used.
			= 2. this cross section is a circle. (Radius given in columns 41-50.)
			= 3. this cross section is an ellipse. (Horizontal semi-axis is given in columns 41-50, the vertical in columns 51-60.)
			= 4. this cross section is circular (radius given in columns 41-50) with an angle array (on card(s) 8D) different from the θ-array on card(s) 5D. This option allows local deviations in the meridian lines.
*One card is needed for each defining station.			= 5. this cross section is specified by a set of ρ (on card(s) 7D) and by a nonstandard set of θ (on card(s) 8D).

	Column	Code	Explanation
			= 6. this cross section is given by a set of Y, Z pairs (on cards 9D).
	41-50	RAD(1)	Radius of section if SCODE = 2. or 4.. Horizontal semi-axis if SCODE = 3.. Not used otherwise.
	51-60	RAD(2)	Vertical semi-axis, if SCODE = 3.. Not used otherwise.
Card(s) 7D (3 maximum per station)	1-10	ρ_1	A set of body radii ρ if SCODE = 1. or 5.. There must be BTHETA $\leq$ 16 values of ρ.
	⋮	⋮	
	51-60	ρ_6	
	etc.		
Card(s) 8D (3 maximum per station)	1-10	θ_1	A set of θ if SCODE = 4. or 5.. There must be BTHETA $\leq$ 16 values of θ.
	⋮	⋮	
	51-60	θ_6	
	etc.		
Card 9D (6 maximum per station)	1-10	Y_1	Array of Y, Z coordinate pairs if SCODE = 6..
	11-20	Z_1	
	21-30	Y_2	
	31-40	Z_2	
	41-50	Y_3	
	51-60	Z_3	
	etc.		
Card 10D	1-4	WING	Columns 1-4 contain the word WING. This card is used whenever a wing is defined. For the case of a body alone, omit cards 10D through 19D. After reading a WING card, the program expects wing definition data.
Card 11D	1-72	TITLE	Any desired title.
Card 12D	1-10	PNLE	Number of corner or break points defining the planform leading edge.

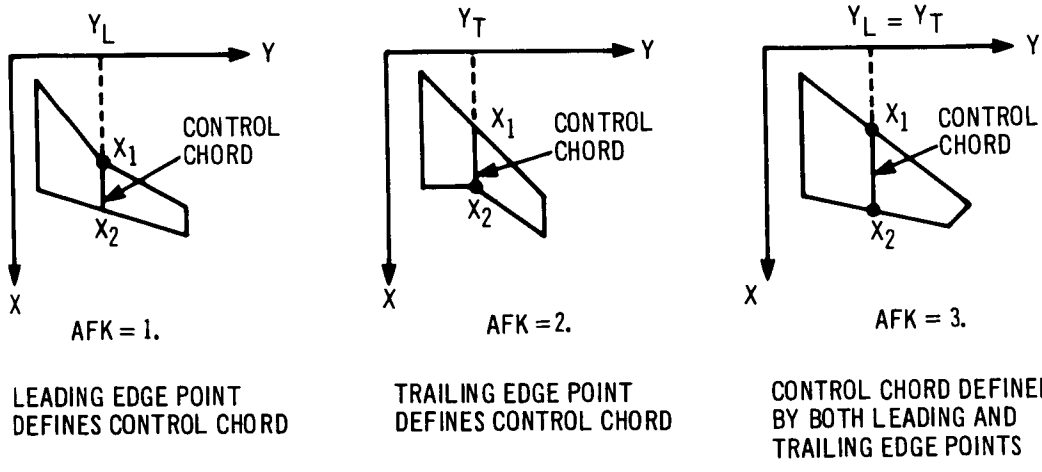
	Column	Code	Explanation
	11-20	PNTE	Number of corner or break points defining the planform trailing edge.
	21-30	AFN	Number of planform control chords. AFN $\geq 2.$, including the wing-tip control chord.
	31-40	PLN	Number of constant percent chord lines used to form spanwise panel edges. Wing leading and trailing edges are counted in this number.
	41-50	WUL	= 1.
	51-60	CHD	Must be left blank.
Card 13D	1-10	PCODE	= 1.
	11-20	ACODE	= 1.
	21-30	EPS	Must be left blank.
Card(s) 14D	1-10	X_1	Array of points defining the planform leading edge, arranged in order from inboard to outboard. There must be PNLE point pairs; three coordinates per card.
	11-20	Y_1	
	21-30	X_2	
	31-40	Y_2	For wing-body combinations, X_1 and Y_1 must lie inside the body so that an intersection can be calculated.
	41-50	X_3	
	51-60	Y_3	
	etc.		
Card(s) 15D	1-10	X_1	Array of points defining the planform trailing edge, arranged in order from inboard to outboard. There must be PNTE point pairs; three coordinates per card.
	11-20	Y_1	
	21-30	X_2	
	31-40	Y_2	For wing-body combinations, X_1 and Y_1 must lie inside the body so that an intersection can be calculated.
	41-50	X_3	

Column	Code	Explanation
51-60	Y_3	
etc.		

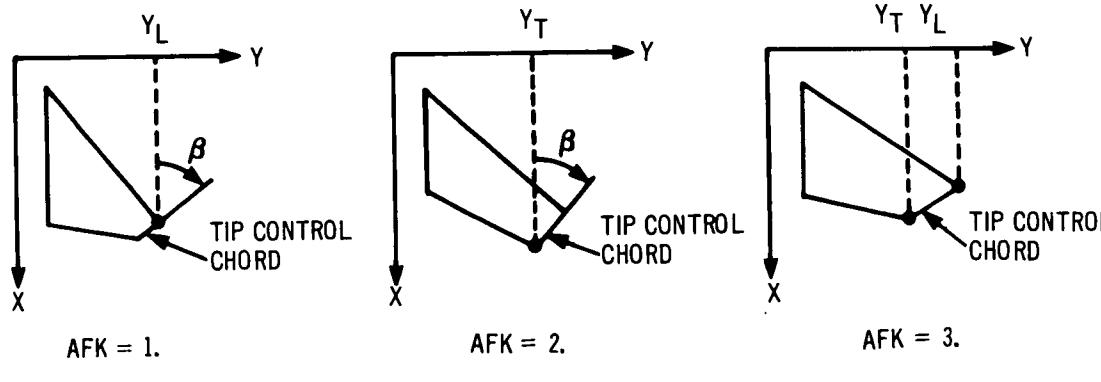
Cards 16D

Cards 16D and 17D always occur in pairs (unless AFNU = 0. on card 16D) to define the wing control chord. There must be AFN ≥ 2. pairs of 16D and 17D cards.

1-10	AFK	Code to indicate how the control chord is oriented on the planform. See sketches below.



INTERNAL WING CONTROL CHORD DEFINITION, $\beta \equiv 0$



WING TIP CONTROL CHORD DEFINITION, $\beta \leq 0$

	Column	Code	Explanation
			Two of the three quantities Y_L, Y_T, or β must be given. AFK indicates the appropriate pair.
	11-20	BETA	The angle β, zero for all wing control chords, except the wing tip (positive as shown above). BETA is ignored if AFK = 3..
	21-30	Y_L	Y-coordinate of the leading edge. Y_L is ignored if AFK = 2..
	31-40	Y_T	Y-coordinate of the trailing edge. Y_T is ignored if AFK = 1..
	41-50	AFNU	= 2. the height and true chord length are specified on the following card 17D. = 0. the previous 17D card values are used. Card 17D should not follow if AFNU = 0..
Card(s) 17D	1-10	X_O	= 0.
	11-20	Z_O	Z-coordinate at the leading edge of control chord.
	21-30	X_C	The control chord true length. If $Z_O = Z_C = 0$, X_C may be given an arbitrary length, which is then scaled by the program to make X_C equal to the true chord length.
	31-40	Z_C	Z-coordinate of control chord at the trailing edge.
Card(s) 18D	1-10 ⋮ 51-60	P_1 ⋮ etc. P_6	Array of constant percent chord values corresponding to the panel spanwise edges. The leading-edge value P_1 = 0.. There are PLN values required with the last value (for the trailing edge) = 100.

	Column	Code	Explanation
Card 19D	1-3	WBX	Columns 1-3 contain the letters WBX. This card indicates that a wing-body intersection is desired. For wing only or body alone cases, this card is omitted.
Card 20D	1-10		= 1. linear interpolation used on body station perimeters to compute additional points between meridian lines in the wing intersection region. = 2. biquadratic interpolation used on body station perimeters to compute additional points between meridian lines in the wing intersection region.
	11-20		Dimensional intersection tolerance. Specifies the accuracy desired in locating wing-body intersection points. A value of 0.001 is suggested.
Card 21D	1-5	TDUMP	Columns 1-5 contain the letters TDUMP. This card is included if a dump of geometry definition and geometry transformation tapes is desired. See Appendix C for a detailed description of these tapes.
Card 22D	1-6	DEFEND	Columns 1-6 contain the word DEFEND. This card ends the definition set and must not be omitted.

GEOMETRY PANELING CARD SET

All paneling data, except title cards and literal statements, are punched in six-field, ten-digit format. A decimal point is required in each data field.

For body-alone case, cards 3P-14P are omitted.

For wing-alone case, cards 4P-7P are omitted.

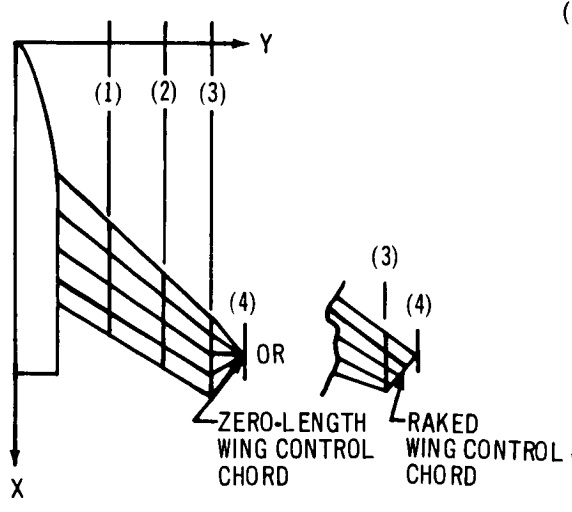
	Column	Code	Explanation
Card 1P	1-5	PANEL	Columns 1-5 contain the word PANEL. This is the first card in the paneling link and must always follow the DEFEND card.
Card 2P	1-10		The number of source control stations at which the radius for an equivalent body of circular cross section and the actual body station centroid height are computed. A maximum of 50 stations may be requested. The radius at each control station is used to determine the source strength necessary to simulate the body thickness. In wing-alone problem card 2P is blank.
	11-20		Dimensional tolerance applied to the additional points generated between meridian lines on the perimeter of body defining stations. This controls the area and centroid location calculations. A value of 0.001 is suggested.
	21-30		This field contains an interpolation code. The program first determines an equivalent radius, R, at each body defining section, X, and then establishes an R vs. X array. Interpolation for additional radii at other stations is performed on this array. The same technique is used to determine centroid locations.

	Column	Code	Explanation
			= 1. linear interpolation for equivalent radii and centroid locations of the source control stations that are between body defining stations.
			= 2. biquadratic interpolation for equivalent radii and centroid locations at the source control stations that are between body defining stations.
	31-40		= 1. linear interpolation between meridian line points on the body definition sections.
			= 2. if biquadratic interpolation is desired.
	41-50		A dimensional tolerance value, E, such that if any equivalent radius length or centroid height, (z centroid), is less than E, its value will be set equal to zero. A value of 0.001 is suggested.
Card 3P	1-10	XPER	Fraction of local streamwise panel chord at which panel control point is located. 0. < XPER < 1..
			NOTE: XPER = .95 for all cases discussed in this report.
	11-20	YPER	Fraction of local panel width at which panel control point is located. 0. < YPER < 1..
			NOTE: YPER = 0. is a code used to locate the panel control point on the chord through the panel centroid. YPER = 0., for all cases discussed in this report.
Card 4P	1-10	BODY PANEL	Columns 1-10 contain the words BODY PANEL.

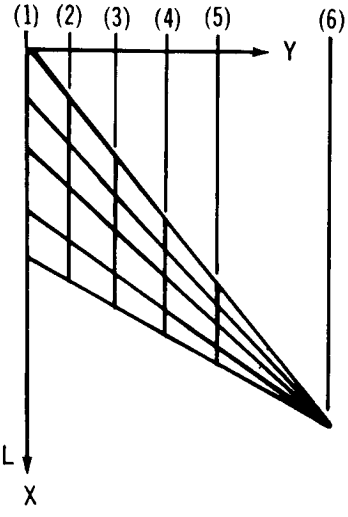
	Column	Code	Explanation
Card 5P	1-10	PLNB*	Number of transverse body panel edges aft of wing trailing edge-body intersection ≤ 21. If PLNB = 0., omit card 6P.
	11-20	PLNW*	Number of transverse body panel edges within the wing body intersection region ≤ 16.
	21-30	TOLB	Slope tolerance on body secondary panel part leading edges. Panel parts with slopes $\leq \left\lvert \frac{\Delta Y}{\Delta X} \right\rvert$ = TOLB (in the local) panel coordinate system) are eliminated. TOLB = 0.02 is suggested.
Card(s) 6P	1-10	$XCEPTB_1$	x-values of transverse body panel edges aft of the wing trailing edge - body intersection. There are PLNB values. Omit this card(s) if PLNB = 0.
	⋮	⋮	
	51-60	$XCEPTB_6$	
	etc.		
Card(s) 7P	1-10	$CODEBW_1$	Each field contains an integer identifying those spanwise wing panel edges which continue around the body to form transverse body panel edges at the body intersection. The table must always start with the integer 1 and terminate with the wing trailing-edge number. There are PLNW values.
	⋮	⋮	
	51-60	$CODEBW_6$	
	etc.		
Card 8P	1-10	WING PANEL	Columns 1-10 contain the words WING PANEL.
Card 9P	1-10	PLANE	Number of buttock lines which locate the streamwise wing panel edges specified by cards 10P and 11P. Wing-alone problem: PLANE is the number of buttock lines locating the streamwise panel edges including both the wing tip and centerline.

*(PLNB + PLNW) $\leq$ 21

Column	Code	Explanation
		Wing-body problem: PLANE is the number of buttock lines locating the streamwise panel edges, but does not include the inboard edge located by the program at the wing-body intersection. PLANE $\geq$ 2.. See sketches below.



PLANE = 4.



PLANE = 6.

11-20	OPTF	= 1. upper and lower airfoil ordinates are read in (cards 12P and 13P) at each wing buttock line passing through the panel centroids. If the wing is untwisted and has the same airfoil section from root to tip, only one airfoil table is necessary. The program will scale this table to fit the appropriate chord.
		= 0. no tables are read in and the wing is a flat plate at zero incidence.
21-30	SNUM	Number of given airfoil ordinate tables.
		= 0., OPTF = 0.

	Column	Code	Explanation
			= 1., same airfoil section from wing root to tip. = (PLANE - 1), wing alone case airfoils specified. = PLANE, wing-body case airfoils specified.
	31-40	TOLW	Slope tolerance on wing secondary panel part leading edges. Panel parts with slopes $\leq \left\lvert \frac{\Delta Y}{\Delta X} \right\rvert$ = TOLW are eliminated. TOLW = 0.01 is suggested.
Card(s) 10P	1-10 ⋮ 51-60 etc.	$YCEPT_1$ ⋮ $YCEPT_6$	Wing buttock line values at which streamwise panel edges are specified. There are (PLANE -1) values. The tip edge is specified on card 11P. NOTE: This card controls the outboard panel edge and in no way influences the spanwise edges which are established by the geometry definition. The outboard panel edge is usually made coincident with the definition wing tip, but it may be used to truncate the defined wing tip and the spanwise panel edges anywhere between the two outboard wing buttock lines specified by card 9P. If truncation is specified, the wing span and area are reduced.
Card 11P	1-10	CPNT	Code indicating how the most outboard panel edge or wing tip is specified. = 0. X and Y coordinates of the wing tip leading and trailing edge are given. Use VALUE(1) through (4).

Column	Code	Explanation
		= 1. X and Y coordinates of the leading edge and the slope ($\Delta X/\Delta Y$) of the wing tip are given. Use VALUE(1), (2) and (5).
		= 2. X and Y coordinates of the trailing edge and the slope ($\Delta X/\Delta Y$) of the wing tip are given. Use VALUE(3), (4) and (5).
11-20	VALUE(1)	X-coordinate of wing tip leading edge if CPNT = 0. or 1..
21-30	VALUE(2)	Y-coordinate of wing tip leading edge if CPNT = 0. or 1..
31-40	VALUE(3)	X-coordinate of wing tip trailing edge if CPNT = 0. or 2..
41-50	VALUE(4)	Y-coordinate of wing tip trailing edge if CPNT = 0. or 2..
51-60	VALUE(5)	wing tip slope, $\frac{\Delta X}{\Delta Y}$, if CPNT = 1. or 2..

Cards 12P and 13P give the SNUM sets of airfoil coordinates. These card sets (12P and 13P) are always used in pairs to define each airfoil at a given panel centroid buttock line. The card sets are omitted if OPTF = 0..

Cards 12P

First Card	1-10	XNUM(1)	Number of points (X, Z coordinate pairs) in upper surface airfoil ordinate table. 4. ≤ XNUM(1) ≤ 25..
Second Cards	1-10	$XFOIL_1$	Upper surface airfoil ordinate table. Local X and Z coordinates are given from leading edge to trailing edge. If the wing has no twist, an unscaled set of ordinates may be given and the program will scale the airfoil to the local chord.
	11-20	$ZFOIL_1$	
	⋮	⋮	
	41-50	$XFOIL_3$	
	51-60	$ZFOIL_3$	
	etc.		

	Column	Code	Explanation
Cards 13P			
First Card	1-10	XNUM(2)	Number of points (x, z coordinate pairs) in lower surface airfoil ordinate table. $4. \leq$ XNUM(2) $\leq 25.$.
Second Cards	1-10	$XFOIL_1$	Lower surface airfoil ordinate table.
	11-20	$ZFOIL_1$	
	⋮	⋮	
	41-50	$XFOIL_3$	
	51-60	$ZFOIL_3$	
	etc.		
Card 14P	1-6	PANEND	Columns 1-6 contain the word PANEND. This card ends the paneling set and must be used whenever any paneling is performed. It is not needed for a body-alone problem.

AERODYNAMIC CARD SET

All aerodynamic data, except title cards and literal statements, are punched in seven-field, ten-digit format. A decimal point is required in each data field. Data cards 1A and 2A are input only once for a given configuration and Mach number. The remaining aerodynamic data cards may be repeated as necessary to solve the selected aerodynamic cases.

	Column	Code	Explanation
Card 1A	1-11	AERODYNAMIC	Columns 1-11 contain the word AERODYNAMIC.
Card 2A	1-10	XMACH	Mach number.
	11-20	SYM	= 0. the aerodynamic problem solved is unsymmetric about the vertical X-Z plane (image panels not included).
			= 1. the aerodynamic problem solved is symmetric about the vertical X-Z plane (image panels included).
Card 3A	1-72	TITLE	Any desired title.
Card 4A	1-10	CASE	= 1. calculates wing twist and camber for a given ΔC_p distribution on wing where $\Delta C_p = C_{p\ lower} - C_{p\ upper}$
			= 2. calculates pressure distribution over the configuration. Wing and body camber can be changed within this option.
			= 3. optimizes wing twist and camber for minimum drag.
			NOTE: For body-alone problems, only case = 2. option is available.

	Column	Code	Explanation
	11-20	CPCALC	= 0. C_p calculations use linear equation; $C_p = -2u$.
			= 1. C_p calculations use nonlinear equation; $C_p = -2u + \beta u^2 - v^2 - w^2$.
	21-30	POLAR	= 0. drag polar not requested.
			= 1. drag polar requested. A series of incremental angles of attack is specified on cards 10A.
	31-40	THICK	= 0. wing thickness pressures are not calculated.
			= 1. wing thickness pressures are calculated.
	41-50	VOUT	= 0. the velocity components are not printed.
			= 1. the velocity components are printed.
Card 5A	1-10	RFAREA	Half-wing reference area. If this field is left blank, the program sums the wing panel areas to obtain the reference area which is the half wing exposed area. For the body-alone problem, a value must be input, or a unit area is used.
	11-20	XP	x-coordinate about which the pitching moments are computed.
	21-30	ZP	z-coordinate about which the pitching moments are computed.

Card(s) 6A For configurations that include a body, two options are available for specifying body camber. The first word on the first card is the key to the type of input the program expects. Omit this card set for wing-alone problems.

Column	Code	Explanation
		Option A
1-5	GIVEN	Columns 1-5 contain the word GIVEN. The program takes the body camber to be that calculated in the geometry definition section. No additional cards are necessary for this option.
7-80	Any additional identifying symbols	
		Option B
1-80	Any identifying symbols	The first card contains any arbitrary identifying symbols (other than GIVEN or CONSTANT as the first word) to describe the body camber and the program expects additional cards immediately following to specify the body camber.
1-10	z_1	z-values or cross-section centroid heights for Option B giving the body camber at the x-locations of the source control stations (see Card 2P). To determine the exact source control stations, it is necessary to have previously run the configuration through the geometry sections of this program.
⋮	⋮	
61-70	z_7	
etc.		

Card(s) 7A

Calculates wing twist and camber for a given wing ΔC_p distribution (CASE = 1., field 1 of card 4A). Two options are available for specifying the ΔC_p distribution. These options are selected by the first word on the first card of this set. Omit this set for body-alone problems or CASE = 2. or 3..

		Option A
1-8	CONSTANT	Columns 1-8 contain the word CONSTANT. This option restricts the wing to have a constant ΔC_p distribution. This constant value is specified on the following card.
9-80	Any additional identifying symbols	
1-10	ΔC_p	ΔC_p for Option A.

Column	Code	Explanation
		Option B
1-80	Any identifying symbols	The first card contains any appropriate identifying symbols (other than GIVEN or CONSTANT as the first word) to select Option B. ΔC_p for each panel is specified on the following card set.
1-10	ΔC_{p_1}	ΔC_p's for Option B. This array must be ordered starting with the inboard panel at the leading edge and running aft to the trailing edge, then proceeding outboard to the tip in the same manner.
⋮	⋮	
61-70	ΔC_{p_7}	
etc.		

Card(s) 8A

Calculates the pressure distribution over the configuration (CASE = 2., field 1 of card 4A). Three options are available for specifying the camber shape of the wing. The options are selected by the first word on the first card of this set. Omit this set for body-alone problems or CASE = 1. or 3..

		Option A
1-8	CONSTANT	Columns 1-8 contain the word CONSTANT. This option restricts the wing camber shape to have a constant slope for each wing panel. This constant value is specified on the following card.
9-80	Any additional identifying symbols	
1-10	$\Delta z/\Delta x$	$\Delta z/\Delta x$ for Option A.
		Option B
1-5	GIVEN	The wing camber shape is specified by the input geometry. The panel slopes are read internally from a tape generated in the paneling section of the program. In this case, no additional cards are necessary.
7-80	Any additional identifying symbols	
		Option C
1-80	Any identifying symbols	Any appropriate identifying symbols (other than GIVEN or CONSTANT as the first word) on the first card of this set are used to select this option. The wing camber shape is specified by a slope for each panel. Additional cards must be input which contain the slope values.

Column	Code	Explanation
1-10	$\Delta z_1/\Delta x_1$	Wing panel slopes for Option C. The array must be ordered starting with the inboard panel at the leading edge and running aft to the trailing edge, then proceeding outboard to the tip in the same manner.
⋮	⋮	
61-70	$\Delta z_7/\Delta x_7$	
etc.		

Card 9A

Optimizes wing twist and camber for minimum drag (CASE = 3., field 1 of card 4A). Two options are available. The first option optimizes the wing for a given wing lift constraint and the second option optimizes the wing for both the wing lift and center of pressure constraints. Only one data card is required. Omit this card for a body-alone problem or CASE = 1. or 2..

1-10	CONSNT	= 0. the wing is optimized for minimum drag with a wing lift constraint.
		= 1. the wing is optimized for minimum drag with both wing lift and x-coordinate of the center of pressure constraints.
11-20	CLBAR	Wing lift coefficient constraint.
21-30	XCPBAR	The x-coordinate of the wing center of pressure constraint. If the center of pressure is not constrained, omit this field.

Card(s) 10A

When the drag polar option is selected (POLAR = 1., field 3 of card 4A), the values of incremental angles of attack added to the immediately preceding case of the defined configuration are given here. These values in degrees are specified in columns 1-10, one value per card for as many cards as necessary. The angle of attack series will be terminated by a blank card. Omit these cards if the polar option is not selected (POLAR = 0.).

NOTE: Additional aerodynamic cases may be requested by returning to card 3A or the aerodynamic cases for this configuration and Mach number may be terminated by proceeding to cards 11A.

	Column	Code	Explanation
Cards 11A			Two blank cards must be placed behind the last data card of each problem, to terminate the selected aerodynamic cases for a given configuration and Mach number. Additional problems (new configurations or Mach numbers) may be stacked consecutively each starting with card 1D.
Terminal Card			Finally, a data card with the words END OF DATA punched in columns 1-11 will terminate the run.

SEVEN FIELD, TEN DIGIT CRD FORMAT

1 10	11 20	21 30	31 40	41 50	51 60	61 69	70	71	72	DENT 80
DEFINE										1D
BODY										2D
TR-805/EQUIVALENT/BODY										3D
24.	8.									4D
0.	25.	50.	75.	102.19	130.					5D
155.	180.									5D
0.	0.	0.	2.	0.						6D
1.5	0.	0.	2.	.270						6D
3.	0.	0.	2.	.4464						6D
4.5	0.	0.	2.	.5943						6D
6.	0.	0.	2.	.7234						6D
7.5	0.	0.	2.	.837						6D
9.	0.	0.	2.	.9364						6D
10.5	0.	0.	2.	1.0223						6D
12.	0.	0.	2.	1.0936						6D
13.5	0.	0.	2.	1.1479						6D
15.	0.	0.	2.	1.184						6D
15.1	0.	0.	0.							6D
17.5	0.	0.	0.							6D
20.	0.	0.	0.							6D
22.5	0.	0.	0.							6D
25.	0.	0.	0.							6D
27.5	0.	0.	0.							6D
29.9	0.	0.	0.							6D
31.	0.	0.	2.	1.174						6D
32.	0.	0.	2.	1.154						6D
33.	0.	0.	2.	1.124						6D
TITLE BOEING WING-BODY MODEL				NAME	DATE				PAGE 1 OF 5	

AD 3778 R1 8-9200

SEVEN FIELD, TEN DIGIT CRD FORMAT

1 10	11 20	21 30	31 40	41 50	51 60	61 69,70	71	72	DENT 80
34.	0.	0.	2.	1.084					6D
35.	0.	0.	2.	1.034					6D
36.	0.	0.	2.	1.					6D
WING									10D
TR-805 WING	(TWP = -.25)								11D
2.	2.	4.	11.	1.	0.				12D
1.	1.	.001							13D
10.67	0.	40.27	10.774						14D
19.88	0.	43.518	8.603						15D
3.	0.	0.	0.	2.					16D
0.	-.25	9.21	-.25						17D
1.	0.	5.							16D
1.	70.	10.774	0.	2.					16D
0.	-.25	3.15	-.25						17D
3.	0.	10.774	8.603	2.					16D
0.	-.25	3.9068	-.25						17D
0.	10.	20.	30.	40.	50.				18D
60.	70.	80.	90.	100.					18D
WBX									19D
2.	.0001								20D
DEFEND									22D
PANEL									1P
50.	0.	1.	1.	.001					2P
.95	0.								3P
BODY PANEL									4P
4.	11.	.02							5P
25.	27.	29.5	32.415						6P

TITLE BOEING WING-BODY MODEL | NAME | DATE | PAGE 2 OF 5

AD 3779 R1 6-9200

SEVEN FIELD, TEN DIGIT CRD FORMAT

1 10	11 20	21 30	31 40	41 50	51 60	61 69 70	71	72 DENT 80
1.	2.	3.	4.	5.	6.			7P
7.	8.	9.	10.	11.				7P
WING PANEL								8P
10.	1.	1.	.05					9P
1.7	2.62	3.54	4.46	5.38	6.3			10P
7.22	8.14	8.603						10P
0.	40.27	10.774	43.518	8.603				11P
15.								12P
0.	0.	.025	.0108	.05	.01556			12P
.1	.02155	.15	.02548	.2	.02832			12P
.3	.03215	.4	.0342	.5	.03385			12P
.6	.03112	.7	.02599	.8	.01915			12P
.9	.01084	.95	.00616	1.	0.			12P
15.								13P
0.	0.	.025	-.00257	.05	-.00325			13P
.1	-.00428	.15	-.0051	.2	-.00581			13P
.3	-.00701	.4	-.00735	.5	-.0065			13P
.6	-.00469	.7	-.00239	.8	-.00068			13P
.9	0.	.95	0.	1.	0.			13P
PANEND								14P
AERODYNAMIC								1A
1.8	1.							2A
TR-805 GIVEN SHAPE								3A
2.	1.	1.	1.	1.				4A
89.375	0.	0.						5A
TR-805 BODY CAMBER								6A
-.2	-.17	-.152	-.136	-.123	-.111	-.10		6A

TITLE BOEING WING-BODY MODEL NAME DATE PAGE 3 OF 5

AD 3779 R1 6-9200

SEVEN FIELD, TEN DIGIT CRD FORMAT

1 10	11 20	21 30	31 40	41 50	51 60	61 69 70	71	72 DENT 80
-.089	-.078	-.0675	-.058	-.048	-.040	-.032		6A
-.025	-.019	-.013	-.008	-.004	-.001	0.		6A
0.	0.	0.	0.	0.	0.	0.		6A
0.	0.	0.	0.	0.	0.	0.		6A
0.	0.	0.	0.	0.	0.	0.		6A
0.	0.	0.	0.	0.	0.	0.		6A
0.								6A
GIVEN WING CAMBER								8A
2.								10A
1.								10A
1.								10A
1.								10A
								BLANK
TR-805 MIN DRAG CL BAR = .1								3A
3.	0.	0.	1.	1.				4A
89.375	0.	0.						5A
TR-805 BODY CAMBER								6A
-.2	-.17	-.152	-.136	-.123	-.111	-.10		6A
-.089	-.078	-.0675	-.058	-.048	-.040	-.032		6A
-.025	-.019	-.013	-.008	-.004	-.001	0.		6A
0.	0.	0.	0.	0.	0.	0.		6A
0.	0.	0.	0.	0.	0.	0.		6A
0.	0.	0.	0.	0.	0.	0.		6A
0.	0.	0.	0.	0.	0.	0.		6A
0.								6A
0.	.1							9A
								11A

TITLE BOEING WING-BODY MODEL | NAME | DATE | PAGE 4 OF 5

AD 3775 R1

6-9200

SEVEN FIELD, TEN DIGIT CRD FORMAT

1 10	11 20	21 30	31 40	41 50	51 60	61	69	70	71	72	DENT 80
											11A
END OF DATA											

TITLE BOEING WING-BODY MODEL | NAME | DATE | PAGE 5 OF 5

AD 3775 R1

6-9200

Sample Data Printout

The computer output for a sample case is given on the following pages:

Data card listing	40
Definition	42
Body definition	43
Wing definition	58
Intersection	68
Transformation	70
Transformed body description	71
Transformed wing description	78
Transformed intersection	80
Body section radius and centroid coordinate vs x-array	81
Paneling	82
Body paneling data	82
Additional body geometry data	85
Wing paneling data	87
Additional wing geometry data	90
Aerodynamics	92
Case 1, wing and body at zero degrees	93
Case 2, wing and body at 2 degrees	107
Case 3, wing and body at 3 degrees	121
Case 4, wing and body at 4 degrees	135
Case 5, wing and body at 5 degrees	149
Case 6, wing optimized for a lift coefficient constraint of 0.1	163

SEP 13, 1965

PROGRAM FOR ANALYSIS AND DESIGN OF SUPERSONIC WING-BODY COMBINATIONS

DATA CARDS ARE LISTED BELOW

```
DEFINE                                                                  1D
BODY                                                                    2D
TR-805 EQUIVALENT BODY                                                  3D
24.       8.                                                            4D
0.        25.       50.       75.       102.19    130.                 5D
155.      180.                                                          5D
0.        0.        0.        2.        0.                            6D
1.5       0.        0.        2.        .270                          6D
3.        0.        0.        2.        .4464                         6D
4.5       0.        0.        2.        .5943                         6D
6.        0.        0.        2.        .7234                         6D
7.5       0.        0.        2.        .837                          6D
9.        0.        0.        2.        .9364                         6D
10.5      0.        0.        2.        1.0223                        6D
12.       0.        0.        2.        1.0936                        6D
13.5      0.        0.        2.        1.1479                        6D
15.       0.        0.        2.        1.184                         6D
15.1      0.        0.        0.                                      6D
17.5      0.        0.        0.                                      6D
20.       0.        0.        0.                                      6D
22.5      0.        0.        0.                                      6D
25.       0.        0.        0.                                      6D
27.5      0.        0.        0.                                      6D
29.9      0.        0.        0.                                      6D
31.       0.        0.        0.        1.174                         6D
32.       0.        0.        0.        1.154                         6D
33.       0.        0.        2.        1.124                         6D
34.       0.        0.        2.        1.084                         6D
35.       0.        0.        2.        1.034                         6D
36.       0.        0.        2.        1.0                           6D
WING                                                                    10D
TR-805 WING (WCP=-.25)                                                  11D
2.        2.        4.        11.       1.                            12D
1.        1.                                                            13D
10.67     0.        40.27     10.774                                  14D
19.88     0.        43.518    8.603                                   15D
3.        0.        0.        0.        2.                            16D
0.        -.25      9.21      -.25                                    17D
1.        0.        5.                                                16D
1.        70.       10.774    0.        2.                            16D
0.        -.25      3.15      -.25                                    17D
3.        0.        10.774    8.603     2.                            16D
0.        -.25      3.9068    -.25                                    17D
0.        10.       20.       30.       40.       50.                 18D
60.       70.       80.       90.       100.                          18D
WBX                                                                     19D
2.        .0001                                                         20D
DEFEND                                                                  22D
PANEL                                                                   1P
50.       0.        1.        1.        .001                          2P
.95       0.                                                            3P
BODY PANEL                                                              4P
4.        11.       .02 .                                               5P
```

```
25.        27.        29.5       32.415                                  6P
1.         2.         3.         4.         5.         6.              7P
7.         8.         9.         10.        11.                        7P
WING PANEL                                                              8P
10.        1.         1.         .05                                    9P
1.7        2.62       3.54       4.46       5.38       6.3             10P
7.22       8.14       8.603                                             10P
0.         40.27      10.774     43.518     8.603                      11P
15.                                                                     12P
0.         0.         .025       .0108      .05        .01556          12P
.1         .02155     .15        .02548     .2         .02832          12P
.3         .03215     .4         .0342      .5         .03385          12P
.6         .03112     .7         .02599     .8         .01915          12P
.9         .01084     .95        .00616     1.0        0.              12P
15.                                                                     13P
0.         0.         .025       -.00257    .05        -.00325         13P
.1         -.00428    .15        -.0051     .2         -.00581         13P
.3         -.00701    .4         -.00735    .5         -.0065          13P
.6         -.00469    .7         -.00239    .8         -.00068         13P
.9         0.         .95        0.         1.         0.              13P
PANEND                                                                  14P
AERODYNAMIC                                                             1A
1.8        1.                                                           2A
TR-805     GIVEN SHAPE                                                  3A
2.         1.         1.         1.         1.                         4A
89.375     0.         0.                                                5A
TR-805     BODY CAMBER                                                  6A
-.2        -.17       -.152      -.136      -.123      -.111      -.10  6A
-.089      -.078      -.C675     -.058      -.048      -.040      -.032 6A
-.025      -.019      -.C13      -.008      -.004      -.001      0.    6A
0.         0.         0.         0.         0.         0.         0.    6A
0.         0.         0.         0.         0.         0.         0.    6A
0.         0.         0.         0.         0.         0.         0.    6A
0.         0.         0.         0.         0.         0.         0.    6A
0.                                                                      6A
GIVEN WING CAMBER                                                       8A
2.                                                                      10A
1.                                                                      10A
1.                                                                      10A
1.                                                                      10A
0.                                                                      10A
TR-805     MIN DRAG CL BAR = .1                                         3A
3.         0.         0.         1.         1.                         4A
89.375     0.         0.                                                5A
TR-805     BODY CAMBER                                                  6A
-.2        -.17       -.152      -.136      -.123      -.111      -.10  6A
-.089      -.078      -.C675     -.058      -.048      -.040      -.032 6A
-.025      -.019      -.C13      -.008      -.004      -.001      0.    6A
0.         0.         0.         0.         0.         0.         0.    6A
0.         0.         0.         0.         0.         0.         0.    6A
0.         0.         0.         0.         0.         0.         0.    6A
0.         0.         0.         0.         0.         0.         0.    6A
0.                                                                      6A
0.         .1                                                           6A
                                                                        9A
                                                                        11A
END OF DATA                                                             11A

     A TOTAL OF  110  DATA CARDS WERE READ
```

2D

BODY

BODY DEFINITION SEP 13, 1965

TR-805/EQUIVALENT/BODY

THERE ARE 24 STATIONS AND 8 MERIDIAN LINES

THE MAIN BODY AXIS (PARALLEL TO THE X AXIS) IS LOCATED AT
Y = -0. Z = -0.

CHORD-HEIGHT TOLERANCE = -0.

BODY STATION X = 0. SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	0.	0.	0.	0.
2	0.	25.0000	0.	0.
3	0.	50.0000	0.	0.
4	0.	75.0000	0.	0.
5	0.	102.1900	0.	-0.
6	0.	130.0000	0.	-0.
7	0.	155.0000	0.	-0.
8	0.	180.0000	0.	-0.

BODY STATION X = 1.5000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	0.2700	0.	0.	0.2700
2	0.2700	25.0000	0.1141	0.2447
3	0.2700	50.0000	0.2068	0.1736
4	0.2700	75.0000	0.2608	0.0699
5	0.2700	102.1900	0.2639	-0.0570
6	0.2700	130.0000	0.2068	-0.1736
7	0.2700	155.0000	0.1141	-0.2447
8	0.2700	180.0000	0.	-0.2700

BODY STATION X = 3.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	0.4464	0.	0.	0.4464
2	0.4464	25.0000	0.1887	0.4046
3	0.4464	50.0000	0.3420	0.2869
4	0.4464	75.0000	0.4312	0.1155
5	0.4464	102.1900	0.4363	-0.0943
6	0.4464	130.0000	0.3420	-0.2869
7	0.4464	155.0000	0.1887	-0.4046

BODY STATION X = 3.0000 (CONTINUED) SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
8	0.4464	180.0000	0.	-0.4464

BODY STATION X = 4.5000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	0.5943	0.	0.	0.5943
2	0.5943	25.0000	0.2512	0.5386
3	0.5943	50.0000	0.4553	0.3820
4	0.5943	75.0000	0.5740	0.1538
5	0.5943	102.1900	0.5809	-0.1255
6	0.5943	130.0000	0.4553	-0.3820
7	0.5943	155.0000	0.2512	-0.5386
8	0.5943	180.0000	0.	-0.5943

BODY STATION X = 6.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	0.7234	0.	0.	0.7234
2	0.7234	25.0000	0.3057	0.6556
3	0.7234	50.0000	0.5542	0.4650
4	0.7234	75.0000	0.6988	0.1872
5	0.7234	102.1900	0.7071	-0.1527
6	0.7234	130.0000	0.5542	-0.4650
7	0.7234	155.0000	0.3057	-0.6556
8	0.7234	180.0000	0.	-0.7234

BODY STATION X = 7.5000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	0.8370	0.	0.	0.8370
2	0.8370	25.0000	0.3537	0.7586
3	0.8370	50.0000	0.6412	0.5380
4	0.8370	75.0000	0.8085	0.2166
5	0.8370	102.1900	0.8181	-0.1767
6	0.8370	130.0000	0.6412	-0.5380
7	0.8370	155.0000	0.3537	-0.7586
8	0.8370	180.0000	0.	-0.8370

TR-805 EQUIVALENT BODY

SEP 13, 1965

BODY STATION X = 9.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	0.9364	0.	0.	0.9364
2	0.9364	25.0000	0.3957	0.8487
3	0.9364	50.0000	0.7173	0.6019
4	0.9364	75.0000	0.9045	0.2424
5	0.9364	102.1900	0.9153	-0.1977
6	0.9364	130.0000	0.7173	-0.6019
7	0.9364	155.0000	0.3957	-0.8487
8	0.9364	180.0000	0.	-0.9364

BODY STATION X = 10.5000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.0223	0.	0.	1.0223
2	1.0223	25.0000	0.4320	0.9265
3	1.0223	50.0000	0.7831	0.6571
4	1.0223	75.0000	0.9875	0.2646
5	1.0223	102.1900	0.9992	-0.2159
6	1.0223	130.0000	0.7831	-0.6571
7	1.0223	155.0000	0.4320	-0.9265
8	1.0223	180.0000	0.	-1.0223

BODY STATION X = 12.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.0936	0.	0.	1.0936
2	1.0936	25.0000	0.4622	0.9911
3	1.0936	50.0000	0.8377	0.7030
4	1.0936	75.0000	1.0563	0.2830
5	1.0936	102.1900	1.0689	-0.2309
6	1.0936	130.0000	0.8377	-0.7030
7	1.0936	155.0000	0.4622	-0.9911
8	1.0936	180.0000	0.	-1.0936

BODY STATION X = 13.5000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.1479	0.	0.	1.1479
2	1.1479	25.0000	0.4851	1.0404
3	1.1479	50.0000	0.8793	0.7379
4	1.1479	75.0000	1.1088	0.2971
5	1.1479	102.1900	1.1220	-0.2424
6	1.1479	130.0000	0.8793	-0.7379
7	1.1479	155.0000	0.4851	-1.0404
8	1.1479	180.0000	0.	-1.1479

BODY STATION X = 15.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.1840	0.	0.	1.1840
2	1.1840	25.0000	0.5004	1.0731
3	1.1840	50.0000	0.9070	0.7611
4	1.1840	75.0000	1.1437	0.3064
5	1.1840	102.1900	1.1573	-0.2500
6	1.1840	130.0000	0.9070	-0.7611
7	1.1840	155.0000	0.5004	-1.0731
8	1.1840	180.0000	0.	-1.1840

BODY STATION X = 15.1000 SCODE = 0.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.1840	0.	0.	1.1840
2	1.1840	25.0000	0.5004	1.0731
3	1.1840	50.0000	0.9070	0.7611
4	1.1840	75.0000	1.1437	0.3064
5	1.1840	102.1900	1.1573	-0.2500
6	1.1840	130.0000	0.9070	-0.7611
7	1.1840	155.0000	0.5004	-1.0731
8	1.1840	180.0000	0.	-1.1840

BODY STATION X = 17.5000 SCODE = 0.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.1840	0.	0.	1.1840
2	1.1840	25.0000	0.5004	1.0731
3	1.1840	50.0000	0.9070	0.7611
4	1.1840	75.0000	1.1437	0.3064
5	1.1840	102.1900	1.1573	-0.2500
6	1.1840	130.0000	0.9070	-0.7611
7	1.1840	155.0000	0.5004	-1.0731
8	1.1840	180.0000	0.	-1.1840

BODY STATION X = 20.0000 SCODE = 0.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.1840	0.	0.	1.1840
2	1.1840	25.0000	0.5004	1.0731
3	1.1840	50.0000	0.9070	0.7611
4	1.1840	75.0000	1.1437	0.3064
5	1.1840	102.1900	1.1573	-0.2500
6	1.1840	130.0000	0.9070	-0.7611
7	1.1840	155.0000	0.5004	-1.0731
8	1.1840	180.0000	0.	-1.1840

BODY STATION X = 22.5000 SCODE = 0.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.1840	0.	0.	1.1840
2	1.1840	25.0000	0.5004	1.0731
3	1.1840	50.0000	0.9070	0.7611
4	1.1840	75.0000	1.1437	0.3064
5	1.1840	102.1900	1.1573	-0.2500
6	1.1840	130.0000	0.9070	-0.7611
7	1.1840	155.0000	0.5004	-1.0731
8	1.1840	180.0000	0.	-1.1840

TR-805 EQUIVALENT BODY

SEP 13, 1965

BODY STATION X = 25.0000 SCODE = 0.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.1840	0.	0.	1.1840
2	1.1840	25.0000	0.5004	1.0731
3	1.1840	50.0000	0.9070	0.7611
4	1.1840	75.0000	1.1437	0.3064
5	1.1840	102.1900	1.1573	-0.2500
6	1.1840	130.0000	0.9070	-0.7611
7	1.1840	155.0000	0.5004	-1.0731
8	1.1840	180.0000	0.	-1.1840

BODY STATION X = 27.5000 SCODE = 0.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.1840	0.	0.	1.1840
2	1.1840	25.0000	0.5004	1.0731
3	1.1840	50.0000	0.9070	0.7611
4	1.1840	75.0000	1.1437	0.3064
5	1.1840	102.1900	1.1573	-0.2500
6	1.1840	130.0000	0.9070	-0.7611
7	1.1840	155.0000	0.5004	-1.0731
8	1.1840	180.0000	0.	-1.1840

BODY STATION X = 29.9000 SCODE = 0.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.1840	0.	0.	1.1840
2	1.1840	25.0000	0.5004	1.0731
3	1.1840	50.0000	0.9070	0.7611
4	1.1840	75.0000	1.1437	0.3064
5	1.1840	102.1900	1.1573	-0.2500
6	1.1840	130.0000	0.9070	-0.7611
7	1.1840	155.0000	0.5004	-1.0731
8	1.1840	180.0000	0.	-1.1840

BODY STATION X = 31.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.1740	0.	0.	1.1740
2	1.1740	25.0000	0.4962	1.0640
3	1.1740	50.0000	0.8993	0.7546
4	1.1740	75.0000	1.1340	0.3039
5	1.1740	102.1900	1.1475	-0.2479
6	1.1740	130.0000	0.8993	-0.7546
7	1.1740	155.0000	0.4962	-1.0640
8	1.1740	180.0000	0.	-1.1740

BODY STATION X = 32.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.1540	0.	0.	1.1540
2	1.1540	25.0000	0.4877	1.0459
3	1.1540	50.0000	0.8840	0.7418
4	1.1540	75.0000	1.1147	0.2987
5	1.1540	102.1900	1.1280	-0.2437
6	1.1540	130.0000	0.8840	-0.7418
7	1.1540	155.0000	0.4877	-1.0459
8	1.1540	180.0000	0.	-1.1540

BODY STATION X = 33.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.1240	0.	0.	1.1240
2	1.1240	25.0000	0.4750	1.0187
3	1.1240	50.0000	0.8610	0.7225
4	1.1240	75.0000	1.0857	0.2909
5	1.1240	102.1900	1.0987	-0.2373
6	1.1240	130.0000	0.8610	-0.7225
7	1.1240	155.0000	0.4750	-1.0187
8	1.1240	180.0000	0.	-1.1240

SEP 13, 1965

BODY STATION X = 34.0000
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)
SCODE = 2.

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.0840	0.	0.	1.0840
2	1.0840	25.0000	0.4581	0.9824
3	1.0840	50.0000	0.8304	0.6968
4	1.0840	75.0000	1.0471	0.2806
5	1.0840	102.1900	1.0596	-0.2289
6	1.0840	130.0000	0.8304	-0.6968
7	1.0840	155.0000	0.4581	-0.9824
8	1.0840	180.0000	0.	-1.0840

BODY STATION X = 35.0000
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)
SCODE = 2.

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.0340	0.	0.	1.0340
2	1.0340	25.0000	0.4370	0.9371
3	1.0340	50.0000	0.7921	0.6646
4	1.0340	75.0000	0.9988	0.2676
5	1.0340	102.1900	1.0107	-0.2183
6	1.0340	130.0000	0.7921	-0.6646
7	1.0340	155.0000	0.4370	-0.9371
8	1.0340	180.0000	0.	-1.0340

BODY STATION X = 36.0000
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)
SCODE = 2.

MERIDIAN NO.	RHO	THETA	Y	Z
1	1.0000	0.	0.	1.0000
2	1.0000	25.0000	0.4226	0.9063
3	1.0000	50.0000	0.7660	0.6428
4	1.0000	75.0000	0.9659	0.2588
5	1.0000	102.1900	0.9775	-0.2112
6	1.0000	130.0000	0.7660	-0.6428
7	1.0000	155.0000	0.4226	-0.9063
8	1.0000	180.0000	0.	-1.0000

TR-805/EQUIVALENT/BODY

SEP 13, 1965

BODY MERIDIAN LINE 1

	X	Y	Z		RHO	THETA
1	0.	0.	0.	*	0.	0.
2	1.5000	0.	0.2700	*	0.2700	0.
3	3.0000	0.	0.4464	*	0.4464	0.
4	4.5000	0.	0.5943	*	0.5943	0.
5	6.0000	0.	0.7234	*	0.7234	0.
6	7.5000	0.	0.8370	*	0.8370	0.
7	9.0000	0.	0.9364	*	0.9364	0.
8	10.5000	0.	1.0223	*	1.0223	0.
9	12.0000	0.	1.0936	*	1.0936	0.
10	13.5000	0.	1.1479	*	1.1479	0.
11	15.0000	0.	1.1840	*	1.1840	0.
12	15.1000	0.	1.1840	*	1.1840	0.
13	17.5000	0.	1.1840	*	1.1840	0.
14	20.0000	0.	1.1840	*	1.1840	0.
15	22.5000	0.	1.1840	*	1.1840	0.
16	25.0000	0.	1.1840	*	1.1840	0.
17	27.5000	0.	1.1840	*	1.1840	0.
18	29.9000	0.	1.1840	*	1.1840	0.
19	31.0000	0.	1.1740	*	1.1740	0.
20	32.0000	0.	1.1540	*	1.1540	0.
21	33.0000	0.	1.1240	*	1.1240	0.
22	34.0000	0.	1.0840	*	1.0840	0.
23	35.0000	0.	1.0340	*	1.0340	0.
24	36.0000	0.	1.0000	*	1.0000	0.

BODY MERIDIAN LINE 2

	X	Y	Z		RHO	THETA
1	0.	0.	0.	*	0.	0.
2	1.5000	0.1141	0.2447	*	0.2700	25.0000
3	3.0000	0.1887	0.4046	*	0.4464	25.0000
4	4.5000	0.2512	0.5386	*	0.5943	25.0000
5	6.0000	0.3057	0.6556	*	0.7234	25.0000
6	7.5000	0.3537	0.7586	*	0.8370	25.0000
7	9.0000	0.3957	0.8487	*	0.9364	25.0000
8	10.5000	0.4320	0.9265	*	1.0223	25.0000
9	12.0000	0.4622	0.9911	*	1.0936	25.0000
10	13.5000	0.4851	1.0404	*	1.1479	25.0000
11	15.0000	0.5004	1.0731	*	1.1840	25.0000
12	15.1000	0.5004	1.0731	*	1.1840	25.0000
13	17.5000	0.5004	1.0731	*	1.1840	25.0000
14	20.0000	0.5004	1.0731	*	1.1840	25.0000

BODY MERIDIAN LINE 2 (CONTINUED)

	X	Y	Z		RHO	THETA
15	22.5000	0.5004	1.0731	*	1.1840	25.0000
16	25.0000	0.5004	1.0731	*	1.1840	25.0000
17	27.5000	0.5004	1.0731	*	1.1840	25.0000
18	29.5000	0.5004	1.0731	*	1.1840	25.0000
19	31.0000	0.4962	1.0640	*	1.1740	25.0000
20	32.0000	0.4877	1.0459	*	1.1540	25.0000
21	33.0000	0.4750	1.0187	*	1.1240	25.0000
22	34.0000	0.4581	0.9824	*	1.0840	25.0000
23	35.0000	0.4370	0.9371	*	1.0340	25.0000
24	36.0000	0.4226	0.9063	*	1.0000	25.0000

BODY MERIDIAN LINE 3

	X	Y	Z		RHO	THETA
1	0.	0.	0.	*	0.	0.
2	1.5000	0.2068	0.1736	*	0.2700	50.0000
3	3.0000	0.3420	0.2869	*	0.4464	50.0000
4	4.5000	0.4553	0.3820	*	0.5943	50.0000
5	6.0000	0.5542	0.4650	*	0.7234	50.0000
6	7.5000	0.6412	0.5380	*	0.8370	50.0000
7	9.0000	0.7173	0.6019	*	0.9364	50.0000
8	10.5000	0.7831	0.6571	*	1.0223	50.0000
9	12.0000	0.8377	0.7030	*	1.0936	50.0000
10	13.5000	0.8793	0.7379	*	1.1479	50.0000
11	15.0000	0.9070	0.7611	*	1.1840	50.0000
12	15.1000	0.9070	0.7611	*	1.1840	50.0000
13	17.5000	0.9070	0.7611	*	1.1840	50.0000
14	20.0000	0.9070	0.7611	*	1.1840	50.0000
15	22.5000	0.9070	0.7611	*	1.1840	50.0000
16	25.0000	0.9070	0.7611	*	1.1840	50.0000
17	27.5000	0.9070	0.7611	*	1.1840	50.0000
18	29.5000	0.9070	0.7611	*	1.1840	50.0000
19	31.0000	0.8993	0.7546	*	1.1740	50.0000
20	32.0000	0.8840	0.7418	*	1.1540	50.0000
21	33.0000	0.8610	0.7225	*	1.1240	50.0000
22	34.0000	0.8304	0.6968	*	1.0840	50.0000
23	35.0000	0.7921	0.6646	*	1.0340	50.0000
24	36.0000	0.7660	0.6425	*	1.0000	50.0000

BODY MERICIAN LINE 4

	X	Y	Z		RHO	THETA
1	0.	0.	0.	*	0.	0.
2	1.5000	0.2608	0.0699	*	0.2700	75.0000
3	3.0000	0.4312	0.1155	*	0.4464	75.0000
4	4.5000	0.5740	0.1538	*	0.5943	75.0000
5	6.0000	0.6988	0.1872	*	0.7234	75.0000
6	7.5000	0.8085	0.2166	*	0.8370	75.0000
7	9.0000	0.9045	0.2424	*	0.9364	75.0000
8	10.5000	0.9875	0.2646	*	1.0223	75.0000
9	12.0000	1.0563	0.2830	*	1.0936	75.0000
10	13.5000	1.1088	0.2971	*	1.1479	75.0000
11	15.0000	1.1437	0.3064	*	1.1840	75.0000
12	15.1000	1.1437	0.3064	*	1.1840	75.0000
13	17.5000	1.1437	0.3064	*	1.1840	75.0000
14	20.0000	1.1437	0.3064	*	1.1840	75.0000
15	22.5000	1.1437	0.3064	*	1.1840	75.0000
16	25.0000	1.1437	0.3064	*	1.1840	75.0000
17	27.5000	1.1437	0.3064	*	1.1840	75.0000
18	29.9000	1.1437	0.3064	*	1.1840	75.0000
19	31.0000	1.1340	0.3039	*	1.1740	75.0000
20	32.0000	1.1147	0.2987	*	1.1540	75.0000
21	33.0000	1.0857	0.2909	*	1.1240	75.0000
22	34.0000	1.0471	0.2806	*	1.0840	75.0000
23	35.0000	0.9988	0.2676	*	1.0340	75.0000
24	36.0000	0.9659	0.2588	*	1.0000	75.0000

BODY MERICIAN LINE 5

	X	Y	Z		RHO	THETA
1	0.	0.	-0.	*	0.	0.
2	1.5000	0.2639	-0.0570	*	0.2700	102.1900
3	3.0000	0.4363	-0.0943	*	0.4464	102.1900
4	4.5000	0.5809	-0.1255	*	0.5943	102.1900
5	6.0000	0.7071	-0.1527	*	0.7234	102.1900
6	7.5000	0.8181	-0.1767	*	0.8370	102.1900
7	9.0000	0.9153	-0.1977	*	0.9364	102.1900
8	10.5000	0.9992	-0.2159	*	1.0223	102.1900
9	12.0000	1.0689	-0.2309	*	1.0936	102.1900
10	13.5000	1.1220	-0.2424	*	1.1479	102.1900
11	15.0000	1.1573	-0.2500	*	1.1840	102.1900
12	15.1000	1.1573	-0.2500	*	1.1840	102.1900
13	17.5000	1.1573	-0.2500	*	1.1840	102.1900
14	20.0000	1.1573	-0.2500	*	1.1840	102.1900

BODY MERIDIAN LINE 5 (CONTINUED)

	X	Y	Z		RHO	THETA
15	22.5000	1.1573	-0.2500	*	1.1840	102.1900
16	25.0000	1.1573	-0.2500	*	1.1840	102.1900
17	27.5000	1.1573	-0.2500	*	1.1840	102.1900
18	29.9000	1.1573	-0.2500	*	1.1840	102.1900
19	31.0000	1.1475	-0.2479	*	1.1740	102.1900
20	32.0000	1.1280	-0.2437	*	1.1540	102.1900
21	33.0000	1.0987	-0.2373	*	1.1240	102.1900
22	34.0000	1.0596	-0.2289	*	1.0840	102.1900
23	35.0000	1.0107	-0.2183	*	1.0340	102.1900
24	36.0000	0.9775	-0.2112	*	1.0000	102.1900

BODY MERIDIAN LINE 6

	X	Y	Z		RHO	THETA
1	0.	0.	-0.	*	0.	0.
2	1.5000	0.2068	-0.1736	*	0.2700	130.0000
3	3.0000	0.3420	-0.2869	*	0.4464	130.0000
4	4.5000	0.4553	-0.3820	*	0.5943	130.0000
5	6.0000	0.5542	-0.4650	*	0.7234	130.0000
6	7.5000	0.6412	-0.5380	*	0.8370	130.0000
7	9.0000	0.7173	-0.6019	*	0.9364	130.0000
8	10.5000	0.7831	-0.6571	*	1.0223	130.0000
9	12.0000	0.8377	-0.7030	*	1.0936	130.0000
10	13.5000	0.8793	-0.7379	*	1.1479	130.0000
11	15.0000	0.9070	-0.7611	*	1.1840	130.0000
12	15.1000	0.9070	-0.7611	*	1.1840	130.0000
13	17.5000	0.9070	-0.7611	*	1.1840	130.0000
14	20.0000	0.9070	-0.7611	*	1.1840	130.0000
15	22.5000	0.9070	-0.7611	*	1.1840	130.0000
16	25.0000	0.9070	-0.7611	*	1.1840	130.0000
17	27.5000	0.9070	-0.7611	*	1.1840	130.0000
18	29.9000	0.9070	-0.7611	*	1.1840	130.0000
19	31.0000	0.8993	-0.7546	*	1.1740	130.0000
20	32.0000	0.8840	-0.7418	*	1.1540	130.0000
21	33.0000	0.8610	-0.7225	*	1.1240	130.0000
22	34.0000	0.8304	-0.6968	*	1.0840	130.0000
23	35.0000	0.7921	-0.6646	*	1.0340	130.0000
24	36.0000	0.7660	-0.6428	*	1.0000	130.0000

BODY MERIDIAN LINE 7

	X	Y	Z		RHO	THETA
1	0.	0.	-0.	*	0.	0.
2	1.5000	0.1141	-0.2447	*	0.2700	155.0000
3	3.0000	0.1887	-0.4046	*	0.4464	155.0000
4	4.5000	0.2512	-0.5386	*	0.5943	155.0000
5	6.0000	0.3057	-0.6556	*	0.7234	155.0000
6	7.5000	0.3537	-0.7586	*	0.8370	155.0000
7	9.0000	0.3957	-0.8487	*	0.9364	155.0000
8	10.5000	0.4320	-0.9265	*	1.0223	155.0000
9	12.0000	0.4622	-0.9911	*	1.0936	155.0000
10	13.5000	0.4851	-1.0404	*	1.1479	155.0000
11	15.0000	0.5004	-1.0731	*	1.1840	155.0000
12	15.1000	0.5004	-1.0731	*	1.1840	155.0000
13	17.5000	0.5004	-1.0731	*	1.1840	155.0000
14	20.0000	0.5004	-1.0731	*	1.1840	155.0000
15	22.5000	0.5004	-1.0731	*	1.1840	155.0000
16	25.0000	0.5004	-1.0731	*	1.1840	155.0000
17	27.5000	0.5004	-1.0731	*	1.1840	155.0000
18	29.9000	0.5004	-1.0731	*	1.1840	155.0000
19	31.0000	0.4962	-1.0640	*	1.1740	155.0000
20	32.0000	0.4877	-1.0459	*	1.1540	155.0000
21	33.0000	0.4750	-1.0187	*	1.1240	155.0000
22	34.0000	0.4581	-0.9824	*	1.0840	155.0000
23	35.0000	0.4370	-0.9371	*	1.0340	155.0000
24	36.0000	0.4226	-0.9063	*	1.0000	155.0000

BODY MERIDIAN LINE 8

	X	Y	Z		RHO	THETA
1	0.	0.	-0.	*	0.	0.
2	1.5000	0.	-0.2700	*	0.2700	180.0000
3	3.0000	0.	-0.4464	*	0.4464	180.0000
4	4.5000	0.	-0.5943	*	0.5943	180.0000
5	6.0000	0.	-0.7234	*	0.7234	180.0000
6	7.5000	0.	-0.8370	*	0.8370	180.0000
7	9.0000	0.	-0.9364	*	0.9364	180.0000
8	10.5000	0.	-1.0223	*	1.0223	180.0000
9	12.0000	0.	-1.0936	*	1.0936	180.0000
10	13.5000	0.	-1.1479	*	1.1479	180.0000
11	15.0000	0.	-1.1840	*	1.1840	180.0000
12	15.1000	0.	-1.1840	*	1.18	180.0000
13	17.5000	0.	-1.1840	*	1.184	180.0000
14	20.0000	0.	-1.1840	*	1.1840	180.0000

BODY MERIDIAN LINE 5 (CONTINUED)

	X	Y	Z		RHO	THETA
15	22.5000	1.1573	-0.2500	*	1.1840	102.1900
16	25.0000	1.1573	-0.2500	*	1.1840	102.1900
17	27.5000	1.1573	-0.2500	*	1.1840	102.1900
18	29.9000	1.1573	-0.2500	*	1.1840	102.1900
19	31.0000	1.1475	-0.2479	*	1.1740	102.1900
20	32.0000	1.1280	-0.2437	*	1.1540	102.1900
21	33.0000	1.0987	-0.2373	*	1.1240	102.1900
22	34.0000	1.0596	-0.2289	*	1.0840	102.1900
23	35.0000	1.0107	-0.2183	*	1.0340	102.1900
24	36.0000	0.9775	-0.2112	*	1.0000	102.1900

BODY MERIDIAN LINE 6

	X	Y	Z		RHO	THETA
1	0.	0.	-0.	*	0.	0.
2	1.5000	0.2068	-0.1736	*	0.2700	130.0000
3	3.0000	0.3420	-0.2869	*	0.4464	130.0000
4	4.5000	0.4553	-0.3820	*	0.5943	130.0000
5	6.0000	0.5542	-0.4650	*	0.7234	130.0000
6	7.5000	0.6412	-0.5380	*	0.8370	130.0000
7	9.0000	0.7173	-0.6019	*	0.9364	130.0000
8	10.5000	0.7831	-0.6571	*	1.0223	130.0000
9	12.0000	0.8377	-0.7030	*	1.0936	130.0000
10	13.5000	0.8793	-0.7379	*	1.1479	130.0000
11	15.0000	0.9070	-0.7611	*	1.1840	130.0000
12	15.1000	0.9070	-0.7611	*	1.1840	130.0000
13	17.5000	0.9070	-0.7611	*	1.1840	130.0000
14	20.0000	0.9070	-0.7611	*	1.1840	130.0000
15	22.5000	0.9070	-0.7611	*	1.1840	130.0000
16	25.0000	0.9070	-0.7611	*	1.1840	130.0000
17	27.5000	0.9070	-0.7611	*	1.1840	130.0000
18	29.9000	0.9070	-0.7611	*	1.1840	130.0000
19	31.0000	0.8993	-0.7546	*	1.1740	130.0000
20	32.0000	0.8840	-0.7418	*	1.1540	130.0000
21	33.0000	0.8610	-0.7225	*	1.1240	130.0000
22	34.0000	0.8304	-0.6968	*	1.0840	130.0000
23	35.0000	0.7921	-0.6646	*	1.0340	130.0000
24	36.0000	0.7660	-0.6428	*	1.0000	130.0000

BODY MERICIAN LINE 7

	X	Y	Z		RHO	THETA
1	0.	0.	-0.	*	0.	0.
2	1.5000	0.1141	-0.2447	*	0.2700	155.0000
3	3.0000	0.1887	-0.4046	*	0.4464	155.0000
4	4.5000	0.2512	-0.5386	*	0.5943	155.0000
5	6.0000	0.3057	-0.6556	*	0.7234	155.0000
6	7.5000	0.3537	-0.7586	*	0.8370	155.0000
7	9.0000	0.3957	-0.8487	*	0.9364	155.0000
8	10.5000	0.4320	-0.9265	*	1.0223	155.0000
9	12.0000	0.4622	-0.9911	*	1.0936	155.0000
10	13.5000	0.4851	-1.0404	*	1.1479	155.0000
11	15.0000	0.5004	-1.0731	*	1.1840	155.0000
12	15.1000	0.5004	-1.0731	*	1.1840	155.0000
13	17.5000	0.5004	-1.0731	*	1.1840	155.0000
14	20.0000	0.5004	-1.0731	*	1.1840	155.0000
15	22.5000	0.5004	-1.0731	*	1.1840	155.0000
16	25.0000	0.5004	-1.0731	*	1.1840	155.0000
17	27.5000	0.5004	-1.0731	*	1.1840	155.0000
18	29.9000	0.5004	-1.0731	*	1.1840	155.0000
19	31.0000	0.4962	-1.0640	*	1.1740	155.0000
20	32.0000	0.4877	-1.0459	*	1.1540	155.0000
21	33.0000	0.4750	-1.0187	*	1.1240	155.0000
22	34.0000	0.4581	-0.9824	*	1.0840	155.0000
23	35.0000	0.4370	-0.9371	*	1.0340	155.0000
24	36.0000	0.4226	-0.9063	*	1.0000	155.0000

BODY MERICIAN LINE 8

	X	Y	Z		RHO	THETA
1	0.	0.	-0.	*	0.	0.
2	1.5000	0.	-0.2700	*	0.2700	180.0000
3	3.0000	0.	-0.4464	*	0.4464	180.0000
4	4.5000	0.	-0.5943	*	0.5943	180.0000
5	6.0000	0.	-0.7234	*	0.7234	180.0000
6	7.5000	0.	-0.8370	*	0.8370	180.0000
7	9.0000	0.	-0.9364	*	0.9364	180.0000
8	10.5000	0.	-1.0223	*	1.0223	180.0000
9	12.0000	0.	-1.0936	*	1.0936	180.0000
10	13.5000	0.	-1.1479	*	1.1479	180.0000
11	15.0000	0.	-1.1840	*	1.1840	180.0000
12	15.1000	0.	-1.1840	*	1.18	180.0000
13	17.5000	0.	-1.1840	*	1.184	180.0000
14	20.0000	0.	-1.1840	*	1.1840	180.0000

BODY MERIDIAN LINE 8 (CONTINUED)

	X	Y	Z		RHO	THETA
15	22.5000	0.	-1.1840	*	1.1840	180.0000
16	25.0000	0.	-1.1840	*	1.1840	180.0000
17	27.5000	0.	-1.1840	*	1.1840	180.0000
18	29.9000	0.	-1.1840	*	1.1840	180.0000
19	31.0000	0.	-1.1740	*	1.1740	180.0000
20	32.0000	0.	-1.1540	*	1.1540	180.0000
21	33.0000	0.	-1.1240	*	1.1240	180.0000
22	34.0000	0.	-1.0840	*	1.0840	180.0000
23	35.0000	0.	-1.0340	*	1.0340	180.0000
24	36.0000	0.	-1.0000	*	1.0000	180.0000

```
WING DEFINITION                                                    SEP 13, 1965

TR-805 WING (WCP=-.25)


THE PLANFORM HAS  2 POINTS ON THE LEADING EDGE,
                  2 POINTS ON THE TRAILING EDGE.
  4 AIRFOILS ARE GIVEN.  11 PERCENT LINES ARE TO BE GENERATED.

PCODE = 1., ACODE = 1., EPS =  0.001000, CHD = 0.
UPPER WING SURFACE IS DEFINED


PLANFORM LEADING  EDGE POINTS (GIVEN) ARE (X,Y) =
   10.6700    0.            40.2700    10.7740


PLANFORM TRAILING EDGE POINTS (GIVEN) ARE (X,Y) =
   19.8800    0.            43.5180     8.6030
```

```
WING DEFINITION                                                    SEP 13, 1965

TR-805 WING (WCP=-.25)


AIRFOIL NO.  1
AFK =  3.,    AFANG =   0.    ,    YL =    0.    ,    YT =    0.
AFNU = 2.,     AFNL =-0.
LEADING AND TRAILING EDGE POINTS (X,Y) ARE
            10.6700    0.                    19.8800    0.
CHORD LENGTH =    9.2100

THE GIVEN   2 UPPER AIRFOIL POINTS ARE
    XWING                  YWING
     0.                   -0.2500
     9.2100               -0.2500

AIRFOIL POINTS AFTER SCALING TO FIT ON PLANFORM ARE
    XWING (SCALED)          Z
     0.                   -0.2500
     9.2100               -0.2500
```

```
WING DEFINITION                                          SEP 13, 1965

TR-805 WING (WCP=-.25)


AIRFOIL NO.  2
AFK =  1.,     AFANG =   0.     ,     YL =    5.0000,     YT =   -0.
AFNU =-0.,     AFNL =-C.
LEADING AND TRAILING EDGE POINTS (X,Y) ARE
            24.4068    5.0000              33.6182    5.0000
CHORD LENGTH =     9.2115

THE GIVEN UPPER AIRFOIL IS THE SAME AS PRECEDING ONE

AIRFOIL POINTS AFTER SCALING TO FIT ON PLANFORM ARE
    XWING (SCALED)          Z
     0.                  -0.2500
     9.2115              -0.2500
```

```
WING DEFINITION                                          SEP 13, 1965

TR-805 WING (WCP=-.25)


AIRFOIL NO.  3
AFK =  1.,    AFANG =  70.0000,    YL =   10.7740,    YT =    0.
AFNU = 2.,     AFNL =-C.
LEADING AND TRAILING EDGE POINTS (X,Y) ARE
            40.2700   10.7740              41.3477    7.8131
CHORD LENGTH =    3.15C9

THE GIVEN   2 UPPER AIRFOIL POINTS ARE
    XWING                  YWING
     0.                    -0.2500
     3.1500                -0.2500

AIRFOIL POINTS AFTER SCALING TO FIT ON PLANFORM ARE
    XWING (SCALED)           Z
     0.                    -0.2501
     3.1509                -0.2501
```

```
WING DEFINITION                                          SEP 13, 1965

TR-805 WING (WCP=-.25)


AIRFOIL NO.  4
AFK =  3.,     AFANG =   0.     ,     YL =   10.7740,     YT =     8.6030
AFNU = 2.,     AFNL =-C.
LEADING AND TRAILING EDGE POINTS (X,Y) ARE
            40.2700   10.7740              43.5180    8.6030
CHORD LENGTH =     3.9068

THE GIVEN   2 UPPER AIRFOIL POINTS ARE
    XWING                  YWING
     0.                   -0.2500
     3.9068               -0.2500

AIRFOIL POINTS AFTER SCALING TO FIT ON PLANFORM ARE
    XWING (SCALED)          Z
     0.                   -0.2500
     3.9068               -0.2500
```

```
WING DEFINITION                                          SEP 13, 1965

TR-805 WING (WCP=-.25)


PLANFORM LEADING  EDGE POINTS (USED)  ARE (X,Y) =
   10.6700     0.            24.4068     5.0000        40.2700    10.7740
   40.2700    10.7740


PLANFORM TRAILING EDGE POINTS (USED)  ARE (X,Y) =
   19.8800     0.            33.6182     5.0000        41.3477     7.8131
   43.5180     8.6030


THE FOLLCWING PERCENT CHORD LINES ARE TO BE CONSTRUCTED
    0.        10.0000   20.0000   30.0000   40.0000   50.0000
   60.0000    7C.0000   8C.0000   90.00C0  100.0000
```

TR-805 WING (WCP=-.25)
POINTS ON UPPER WING SURFACE PERCENT CHORD LINES SEP 13, 1965

WING PERCENT CHORD LINE 1, C. PERCENT

AF NO.	X	Y	Z
1	10.6700	0.	-0.2500
2	24.4068	5.0000	-0.2500
3	40.2700	10.7740	-0.2501
4	40.2700	10.7740	-0.2500

WING PERCENT CHORD LINE 2, 10.0000 PERCENT

AF NO.	X	Y	Z
1	11.5910	0.	-0.2500
2	25.3279	5.0000	-0.2500
3	40.3778	10.4779	-0.2501
4	40.5948	10.5569	-0.2500

WING PERCENT CHORD LINE 3, 20.0000 PERCENT

AF NO.	X	Y	Z
1	12.5120	0.	-0.2500
2	26.2491	5.0000	-0.2500
3	40.4855	10.1818	-0.2501
4	40.9196	10.3398	-0.2500

WING PERCENT CHORD LINE 4, 30.0000 PERCENT

AF NO.	X	Y	Z
1	13.4330	0.	-0.2500
2	27.1702	5.0000	-0.2500
3	40.5933	9.8857	-0.2501
4	41.2444	10.1227	-0.2500

WING PERCENT CHORD LINE 5, 40.0000 PERCENT

AF NO.	X	Y	Z
1	14.3540	0.	-0.2500
2	28.0914	5.0000	-0.2500
3	40.7011	9.5896	-0.2501
4	41.5692	9.9056	-0.2500

TR-805 WING (WCP=-.25)
POINTS ON UPPER WING SURFACE PERCENT CHORD LINES SEP 13, 1965

WING PERCENT CHORD LINE 6, 50.0000 PERCENT

AF NO.	X	Y	Z
1	15.2750	0.	-0.2500
2	29.0125	5.0000	-0.2500
3	40.8088	9.2936	-0.2501
4	41.8940	9.6885	-0.2500

WING PERCENT CHORD LINE 7, 60.0000 PERCENT

AF NO.	X	Y	Z
1	16.1960	0.	-0.2500
2	29.9336	5.0000	-0.2500
3	40.9166	8.9975	-0.2501
4	42.2188	9.4714	-0.2500

WING PERCENT CHORD LINE 8, 70.0000 PERCENT

AF NO.	X	Y	Z
1	17.1170	0.	-0.2500
2	30.8548	5.0000	-0.2500
3	41.0244	8.7014	-0.2501
4	42.5436	9.2543	-0.2500

WING PERCENT CHORD LINE 9, 80.0000 PERCENT

AF NO.	X	Y	Z
1	18.0380	0.	-0.2500
2	31.7759	5.0000	-0.2500
3	41.1321	8.4053	-0.2501
4	42.8684	9.0372	-0.2500

WING PERCENT CHORD LINE 10, 90.0000 PERCENT

AF NO.	X	Y	Z
1	18.9590	0.	-0.2500
2	32.6971	5.0000	-0.2500
3	41.2399	8.1092	-0.2501
4	43.1932	8.8201	-0.2500

TR-805 WING (WCP=-.25)
POINTS ON UPPER WING SURFACE PERCENT CHORD LINES SEP 13, 1965

WING PERCENT CHORD LINE 11, 100.0000 PERCENT

AF NO.	X	Y	Z
1	19.8800	0.	-0.2500
2	33.6182	5.0000	-0.2500
3	41.3477	7.8131	-0.2501
4	43.5180	8.6030	-0.2500

```
FIND THE INTERSECTION OF EACH WING PERCENT LINE WITH THE BODY SURFACE.

          CODE =  2

          TOLERANCE = 0.0001

LET P BE A POINT WHICH LIES ON THE BODY SURFACE
AND Q A POINT ON A LINE SEGMENT CONNECTING TWO ADJACENT
WING PERCENT LINE POINTS, SUCH THAT P AND Q HAVE THE SAME X-COORDINATE.

THE PROGRAM ITERATES UNTIL P AND Q HAVE THE SAME
(OR NEARLY THE SAME) Y AND Z COORDINATES.
THEN P IS THE INTERSECTION POINT.

IF P AND Q DIFFER BY MORE THAN THE GIVEN TOLERANCE,
A WARNING MESSAGE IS WRITTEN.

THE SHAPE OF THE BODY SURFACE BETWEEN MERIDIAN LINES (IN ANY SECTION
PARALLEL TO THE YZ PLANE) IS DETERMINED BY THE GIVEN CODE.
    IF CODE = 1, LINEAR INTERPOLATION IS USED BETWEEN MERIDIAN LINES.
    IF CODE = 2, BIQUADRATIC INTERPOLATION IS USED.
```

TR-805 EQUIVALENT BODY
TR-805 WING (WCP=-.25)

SEP 13, 1965

INTERSECTIONS OF UPPER WING PERCENT CHORD LINES WITH BODY SURFACE
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (-0. , -0.)

NO.	PERCENT	X	Y	Z	RHO	THETA
1	0.	13.7701	1.1284	-0.2500	1.1558	102.4926
2	10.0000	14.7546	1.1516	-0.2500	1.1784	102.2492
3	20.0000	15.6916	1.1573	-0.2500	1.1840	102.1901
4	30.0000	16.6126	1.1573	-0.2500	1.1840	102.1901
5	40.0000	17.5337	1.1573	-0.2500	1.1840	102.1901
6	50.0000	18.4547	1.1573	-0.2500	1.1840	102.1901
7	60.0000	19.3757	1.1573	-0.2500	1.1840	102.1901
8	70.0000	20.2968	1.1573	-0.2500	1.1840	102.1901
9	80.0000	21.2178	1.1573	-0.2500	1.1840	102.1901
10	90.0000	22.1388	1.1573	-0.2500	1.1840	102.1901
11	100.0000	23.0599	1.1573	-0.2500	1.1840	102.1901

SEP 13, 1965

PROGRAM ENTERED LINK TO TRANSFORM BODY AND/OR WING BEFORE PANELLING

THE ORIGIN OF THE NEW BODY COORDINATE SYSTEM IS AT
(X,Y,Z) = (0. , 0. , 0.)

A UNIT VECTOR ALONG THE X-PRIME AXIS HAS COMPONENTS
(X,Y,Z) = (1.0000, 0. , 0.)

NO.	ORIGINAL BODY STATION	CIRCULARIZED BODY STATION	BODY RADIUS	Z-BODY CENTROID
1	0.	0.	0.	0.
2	1.5000	1.5000	0.2700	0.
3	3.0000	3.0000	0.4464	0.
4	4.5000	4.5000	0.5943	0.
5	6.0000	6.0000	0.7234	0.
6	7.5000	7.5000	0.8370	0.
7	9.0000	9.0000	0.9364	0.
8	10.5000	10.5000	1.0223	0.
9	12.0000	12.0000	1.0936	0.
10	13.5000	13.5000	1.1479	0.
11	15.0000	15.0000	1.1840	0.
12	15.1000	15.1000	1.1840	0.
13	17.5000	17.5000	1.1840	0.
14	20.0000	20.0000	1.1840	0.
15	22.5000	22.5000	1.1840	0.
16	25.0000	25.0000	1.1840	0.
17	27.5000	27.5000	1.1840	0.
18	29.9000	29.9000	1.1840	0.
19	31.0000	31.0000	1.1740	0.
20	32.0000	32.0000	1.1540	0.
21	33.0000	33.0000	1.1240	0.
22	34.0000	34.0000	1.0840	0.
23	35.0000	35.0000	1.0340	0.
24	36.0000	36.0000	1.0000	0.

TRANSFORMED BODY (PRIMED-AXIS SYSTEM)

SEP 13, 1965

BODY MERIDIAN LINE 1

	X	Y	Z	RHO	THETA
1	0.	0.	0.	0.	0.
2	1.5000	0.	0.2700	0.2700	0.
3	3.0000	0.	0.4464	0.4464	0.
4	4.5000	0.	0.5943	0.5943	0.
5	6.0000	0.	0.7234	0.7234	0.
6	7.5000	0.	0.8370	0.8370	0.
7	9.0000	0.	0.9364	0.9364	0.
8	10.5000	0.	1.0223	1.0223	0.
9	12.0000	0.	1.0936	1.0936	0.
10	13.5000	0.	1.1479	1.1479	0.
11	15.0000	0.	1.1840	1.1840	0.
12	15.1000	0.	1.1840	1.1840	0.
13	17.5000	0.	1.1840	1.1840	0.
14	20.0000	0.	1.1840	1.1840	0.
15	22.5000	0.	1.1840	1.1840	0.
16	25.0000	0.	1.1840	1.1840	0.
17	27.5000	0.	1.1840	1.1840	0.
18	29.9000	0.	1.1840	1.1840	0.
19	31.0000	0.	1.1740	1.1740	0.
20	32.0000	0.	1.1540	1.1540	0.
21	33.0000	0.	1.1240	1.1240	0.
22	34.0000	0.	1.0840	1.0840	0.
23	35.0000	0.	1.0340	1.0340	0.
24	36.0000	0.	1.0000	1.0000	0.

BODY MERIDIAN LINE 2

	X	Y	Z	RHO	THETA
1	0.	0.	0.	0.	0.
2	1.5000	0.1141	0.2447	0.2700	25.0000
3	3.0000	0.1887	0.4046	0.4464	25.0000
4	4.5000	0.2512	0.5386	0.5943	25.0000
5	6.0000	0.3057	0.6556	0.7234	25.0000
6	7.5000	0.3537	0.7586	0.8370	25.0000
7	9.0000	0.3957	0.8487	0.9364	25.0000
8	10.5000	0.4320	0.9265	1.0223	25.0000
9	12.0000	0.4622	0.9911	1.0936	25.0000
10	13.5000	0.4851	1.0404	1.1479	25.0000
11	15.0000	0.5004	1.0731	1.1840	25.0000
12	15.1000	0.5004	1.0731	1.1840	25.0000
13	17.5000	0.5004	1.0731	1.1840	25.0000
14	20.0000	0.5004	1.0731	1.1840	25.0000

TRANSFORMED BODY (PRIMED-AXIS SYSTEM)

SEP 13, 1965

BODY MERIDIAN LINE 2 (CONTINUED)

	X	Y	Z	RHO	THETA
15	22.5000	0.5004	1.0731	1.1840	25.0000
16	25.0000	0.5004	1.0731	1.1840	25.0000
17	27.5000	0.5004	1.0731	1.1840	25.0000
18	29.9000	0.5004	1.0731	1.1840	25.0000
19	31.0000	0.4962	1.0640	1.1740	25.0000
20	32.0000	0.4877	1.0459	1.1540	25.0000
21	33.0000	0.4750	1.0187	1.1240	25.0000
22	34.0000	0.4581	0.9824	1.0840	25.0000
23	35.0000	0.4370	0.9371	1.0340	25.0000
24	36.0000	0.4226	0.9063	1.0000	25.0000

BODY MERIDIAN LINE 3

	X	Y	Z	RHO	THETA
1	0.	0.	0.	0.	0.
2	1.5000	0.2068	0.1736	0.2700	50.0000
3	3.0000	0.3420	0.2869	0.4464	50.0000
4	4.5000	0.4553	0.3820	0.5943	50.0000
5	6.0000	0.5542	0.4650	0.7234	50.0000
6	7.5000	0.6412	0.5380	0.8370	50.0000
7	9.0000	0.7173	0.6019	0.9364	50.0000
8	10.5000	0.7831	0.6571	1.0223	50.0000
9	12.0000	0.8377	0.7030	1.0936	50.0000
10	13.5000	0.8793	0.7379	1.1479	50.0000
11	15.0000	0.9070	0.7611	1.1840	50.0000
12	15.1000	0.9070	0.7611	1.1840	50.0000
13	17.5000	0.9070	0.7611	1.1840	50.0000
14	20.0000	0.9070	0.7611	1.1840	50.0000
15	22.5000	0.9070	0.7611	1.1840	50.0000
16	25.0000	0.9070	0.7611	1.1840	50.0000
17	27.5000	0.9070	0.7611	1.1840	50.0000
18	29.9000	0.9070	0.7611	1.1840	50.0000
19	31.0000	0.8993	0.7546	1.1740	50.0000
20	32.0000	0.8840	0.7418	1.1540	50.0000
21	33.0000	0.8610	0.7225	1.1240	50.0000
22	34.0000	0.8304	0.6968	1.0840	50.0000
23	35.0000	0.7921	0.6646	1.0340	50.0000
24	36.0000	0.7660	0.6428	1.0000	50.0000

TRANSFORMED BODY (PRIMED-AXIS SYSTEM)

SEP 13, 1965

BODY MERICIAN LINE 4

	X	Y	Z	RHO	THETA
1	0.	0.	0.	0.	0.
2	1.5000	0.2608	0.0699	0.2700	75.0000
3	3.0000	0.4312	0.1155	0.4464	75.0000
4	4.5000	0.5740	0.1538	0.5943	75.0000
5	6.0000	0.6988	0.1872	0.7234	75.0000
6	7.5000	0.8085	0.2166	0.8370	75.0000
7	9.0000	0.9045	0.2424	0.9364	75.0000
8	10.5000	0.9875	0.2646	1.0223	75.0000
9	12.0000	1.0563	0.2830	1.0936	75.0000
10	13.5000	1.1086	0.2971	1.1479	75.0000
11	15.0000	1.1437	0.3064	1.1840	75.0000
12	15.1000	1.1437	0.3064	1.1840	75.0000
13	17.5000	1.1437	0.3064	1.1840	75.0000
14	20.0000	1.1437	0.3064	1.1840	75.0000
15	22.5000	1.1437	0.3064	1.1840	75.0000
16	25.0000	1.1437	0.3064	1.1840	75.0000
17	27.5000	1.1437	0.3064	1.1840	75.0000
18	29.9000	1.1437	0.3064	1.1840	75.0000
19	31.0000	1.1340	0.3039	1.1740	75.0000
20	32.0000	1.1147	0.2987	1.1540	75.0000
21	33.0000	1.0857	0.2909	1.1240	75.0000
22	34.0000	1.0471	0.2806	1.0840	75.0000
23	35.0000	0.9988	0.2676	1.0340	75.0000
24	36.0000	0.9659	0.2588	1.0000	75.0000

BODY MERICIAN LINE 5

	X	Y	Z	RHO	THETA
1	-0.	-0.	-0.	0.	0.
2	1.5000	0.2639	-0.0570	0.2700	102.1900
3	3.0000	0.4363	-0.0943	0.4464	102.1900
4	4.5000	0.5809	-0.1255	0.5943	102.1900
5	6.0000	0.7071	-0.1527	0.7234	102.1900
6	7.5000	0.8181	-0.1767	0.8370	102.1900
7	9.0000	0.9153	-0.1977	0.9364	102.1900
8	10.5000	0.9992	-0.2159	1.0223	102.1900
9	12.0000	1.0689	-0.2309	1.0936	102.1900
10	13.5000	1.1220	-0.2424	1.1479	102.1900
11	15.0000	1.1573	-0.2500	1.1840	102.1900
12	15.1000	1.1573	-0.2500	1.1840	102.1900
13	17.5000	1.1573	-0.2500	1.1840	102.1900
14	20.0000	1.1573	-0.2500	1.1840	102.1900

TRANSFORMED BODY (PRIMED-AXIS SYSTEM)

SEP 13, 1965

BODY MERIDIAN LINE 5 (CONTINUED)

	X	Y	Z	RHO	THETA
15	22.5000	1.1573	-0.2500	1.1840	102.1900
16	25.0000	1.1573	-0.2500	1.1840	102.1900
17	27.5000	1.1573	-0.2500	1.1840	102.1900
18	29.9000	1.1573	-0.2500	1.1840	102.1900
19	31.0000	1.1475	-0.2479	1.1740	102.1900
20	32.0000	1.1280	-0.2437	1.1540	102.1900
21	33.0000	1.0987	-0.2373	1.1240	102.1900
22	34.0000	1.0596	-0.2289	1.0840	102.1900
23	35.0000	1.0107	-0.2183	1.0340	102.1900
24	36.0000	0.9775	-0.2112	1.0000	102.1900

BODY MERIDIAN LINE 6

	X	Y	Z	RHO	THETA
1	-0.	-0.	-0.	0.	0.
2	1.5000	0.2068	-0.1736	0.2700	130.0000
3	3.0000	0.3420	-0.2869	0.4464	130.0000
4	4.5000	0.4553	-0.3820	0.5943	130.0000
5	6.0000	0.5542	-0.4650	0.7234	130.0000
6	7.5000	0.6412	-0.5380	0.8370	130.0000
7	9.0000	0.7173	-0.6019	0.9364	130.0000
8	10.5000	0.7831	-0.6571	1.0223	130.0000
9	12.0000	0.8377	-0.7030	1.0936	130.0000
10	13.5000	0.8793	-0.7379	1.1479	130.0000
11	15.0000	0.9070	-0.7611	1.1840	130.0000
12	15.1000	0.9070	-0.7611	1.1840	130.0000
13	17.5000	0.9070	-0.7611	1.1840	130.0000
14	20.0000	0.9070	-0.7611	1.1840	130.0000
15	22.5000	0.9070	-0.7611	1.1840	130.0000
16	25.0000	0.9070	-0.7611	1.1840	130.0000
17	27.5000	0.9070	-0.7611	1.1840	130.0000
18	29.9000	0.9070	-0.7611	1.1840	130.0000
19	31.0000	0.8993	-0.7546	1.1740	130.0000
20	32.0000	0.8840	-0.7418	1.1540	130.0000
21	33.0000	0.8610	-0.7225	1.1240	130.0000
22	34.0000	0.8304	-0.6968	1.0840	130.0000
23	35.0000	0.7921	-0.6646	1.0340	130.0000
24	36.0000	0.7660	-0.6428	1.0000	130.0000

TRANSFORMED BODY (PRIMED-AXIS SYSTEM)

SEP 13, 1965

BODY MERIDIAN LINE 7

	X	Y	Z	RHO	THETA
1	-0.	-0.	-0.	0.	0.
2	1.5000	0.1141	-0.2447	0.2700	155.0000
3	3.0000	0.1867	-0.4046	0.4464	155.0000
4	4.5000	0.2512	-0.5386	0.5943	155.0000
5	6.0000	0.3057	-0.6556	0.7234	155.0000
6	7.5000	0.3537	-0.7586	0.8370	155.0000
7	9.0000	0.3957	-0.8487	0.9364	155.0000
8	10.5000	0.4320	-0.9265	1.0223	155.0000
9	12.0000	0.4622	-0.9911	1.0936	155.0000
10	13.5000	0.4851	-1.0404	1.1479	155.0000
11	15.0000	0.5004	-1.0731	1.1840	155.0000
12	15.1000	0.5004	-1.0731	1.1840	155.0000
13	17.5000	0.5004	-1.0731	1.1840	155.0000
14	20.0000	0.5004	-1.0731	1.1840	155.0000
15	22.5000	0.5004	-1.0731	1.1840	155.0000
16	25.0000	0.5004	-1.0731	1.1840	155.0000
17	27.5000	0.5004	-1.0731	1.1840	155.0000
18	29.9000	0.5004	-1.0731	1.1840	155.0000
19	31.0000	0.4962	-1.0640	1.1740	155.0000
20	32.0000	0.4877	-1.0459	1.1540	155.0000
21	33.0000	0.4750	-1.0187	1.1240	155.0000
22	34.0000	0.4581	-0.9824	1.0840	155.0000
23	35.0000	0.4370	-0.9371	1.0340	155.0000
24	36.0000	0.4226	-0.9063	1.0000	155.0000

BODY MERIDIAN LINE 8

	X	Y	Z	RHO	THETA
1	-0.	-0.	-0.	0.	0.
2	1.5000	-0.	-0.2700	0.2700	180.0000
3	3.0000	-0.	-0.4464	0.4464	180.0000
4	4.5000	-0.	-0.5943	0.5943	180.0000
5	6.0000	-0.	-0.7234	0.7234	180.0000
6	7.5000	-0.	-0.8370	0.8370	180.0000
7	9.0000	-0.	-0.9364	0.9364	180.0000
8	10.5000	-0.	-1.0223	1.0223	180.0000
9	12.0000	-0.	-1.0936	1.0936	180.0000
10	13.5000	-0.	-1.1479	1.1479	180.0000
11	15.0000	-0.	-1.1840	1.1840	180.0000
12	15.1000	-0.	-1.1840	1.1840	180.0000
13	17.5000	-0.	-1.1840	1.1840	180.0000
14	20.0000	-0.	-1.1840	1.1840	180.0000

TRANSFORMED BODY (PRIMED-AXIS SYSTEM)

SEP 13, 1965

BODY MERIDIAN LINE 8 (CONTINUED)

	X	Y	Z	RHO	THETA
15	22.5000	-0.	-1.1640	1.1640	180.0000
16	25.0000	-0.	-1.1840	1.1840	180.0000
17	27.5000	-0.	-1.1640	1.1840	180.0000
18	29.9000	-0.	-1.1640	1.1640	180.0000
19	31.0000	-0.	-1.1740	1.1740	180.0000
20	32.0000	-0.	-1.1540	1.1540	180.0000
21	33.0000	-0.	-1.1240	1.1240	180.0000
22	34.0000	-0.	-1.0840	1.0840	180.0000
23	35.0000	-0.	-1.0340	1.0340	180.0000
24	36.0000	-0.	-1.0000	1.0000	180.0000

POINTS ON FLAT TRANSFORMED WING (PRIMED-AXIS SYSTEM)

SEP 13, 1965

WING PERCENT CHORD LINE 1, 0.

	X	Y	Z
1	10.6700	-0.	-0.2500
2	24.4068	5.0000	-0.2500
3	40.2700	10.7740	-0.2500

WING PERCENT CHORD LINE 2, 10.0000

	X	Y	Z
1	11.5910	-0.	-0.2500
2	25.3279	5.0000	-0.2500
3	40.3776	10.4779	-0.2500
4	40.5948	10.5569	-0.2500

WING PERCENT CHORD LINE 3, 20.0000

	X	Y	Z
1	12.5120	-0.	-0.2500
2	26.2491	5.0000	-0.2500
3	40.4855	10.1818	-0.2500
4	40.9196	10.3398	-0.2500

WING PERCENT CHORD LINE 4, 30.0000

	X	Y	Z
1	13.4330	-0.	-0.2500
2	27.1702	5.0000	-0.2500
3	40.5933	9.8857	-0.2500
4	41.2444	10.1227	-0.2500

WING PERCENT CHORD LINE 5, 40.0000

	X	Y	Z
1	14.3540	-0.	-0.2500
2	28.0914	5.0000	-0.2500
3	40.7011	9.5896	-0.2500
4	41.5692	9.9056	-0.2500

WING PERCENT CHORD LINE 6, 50.0000

	X	Y	Z
1	15.2750	-0.	-0.2500
2	29.0125	5.0000	-0.2500
3	40.8088	9.2936	-0.2500
4	41.8940	9.6885	-0.2500

POINTS ON FLAT TRANSFORMED WING (PRIMED-AXIS SYSTEM)

SEP 13, 1965

WING PERCENT CHORD LINE 7, 60.0000

	X	Y	Z
1	16.1960	-0.	-0.2500
2	29.9336	5.0000	-0.2500
3	40.9166	8.9975	-0.2500
4	42.2188	9.4714	-0.2500

WING PERCENT CHORD LINE 8, 70.0000

	X	Y	Z
1	17.1170	-0.	-0.2500
2	30.8548	5.0000	-0.2500
3	41.0244	8.7014	-0.2500
4	42.5436	9.2543	-0.2500

WING PERCENT CHORD LINE 9, 80.0000

	X	Y	Z
1	18.0380	-0.	-0.2500
2	31.7759	5.0000	-0.2500
3	41.1321	8.4053	-0.2500
4	42.8684	9.0372	-0.2500

WING PERCENT CHORD LINE 10, 90.0000

	X	Y	Z
1	18.9590	-0.	-0.2500
2	32.6971	5.0000	-0.2500
3	41.2399	8.1092	-0.2500
4	43.1932	8.8201	-0.2500

WING PERCENT CHORD LINE 11, 100.0000

	X	Y	Z
1	19.8800	-0.	-0.2500
2	33.6182	5.0000	-0.2500
3	41.3477	7.8131	-0.2500
4	43.5180	8.6030	-0.2500

SEP 13, 1965

INTERSECTIONS OF FLAT WING PERCENT CHORD LINES WITH BODY SURFACE
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (0. , 0.)

(PRIMED-AXIS SYSTEM)

NO.	PERCENT	X	Y	Z	RHO	THETA
1	0.	13.7610	1.1251	-0.2500	1.1525	102.5285
2	10.0000	14.7529	1.1509	-0.2500	1.1777	102.2562
3	20.0000	15.6916	1.1573	-0.2500	1.1840	102.1901
4	30.0000	16.6126	1.1573	-0.2500	1.1840	102.1901
5	40.0000	17.5337	1.1573	-0.2500	1.1840	102.1901
6	50.0000	18.4547	1.1573	-0.2500	1.1840	102.1901
7	60.0000	19.3757	1.1573	-0.2500	1.1840	102.1901
8	70.0000	20.2968	1.1573	-0.2500	1.1840	102.1901
9	80.0000	21.2178	1.1573	-0.2500	1.1840	102.1901
10	90.0000	22.1388	1.1573	-0.2500	1.1840	102.1901
11	100.0000	23.0599	1.1573	-0.2500	1.1840	102.1901

SEP 13, 1965

BODY RADIUS AND Z-COORDINATE OF BODY CENTROID VERSUS X-PRIME

NO.	X-PRIME	RADIUS	Z-PRIME
1	0.	0.	0.
2	0.7243	0.1304	0.
3	1.4485	0.2607	0.
4	2.1728	0.3491	0.
5	2.8970	0.4343	0.
6	3.6213	0.5077	0.
7	4.3456	0.5791	0.
8	5.0698	0.6433	0.
9	5.7941	0.7057	0.
10	6.5184	0.7627	0.
11	7.2426	0.8175	0.
12	7.9669	0.8679	0.
13	8.6911	0.9159	0.
14	9.4154	0.9602	0.
15	10.1397	1.0017	0.
16	10.8639	1.0396	0.
17	11.5882	1.0740	0.
18	12.3124	1.1049	0.
19	13.0367	1.1311	0.
20	13.7610	1.1542	0.
21	14.5023	1.1720	0.
22	15.2436	1.1840	0.
23	15.9849	1.1840	0.
24	16.7262	1.1840	0.
25	17.4675	1.1840	0.
26	18.2088	1.1840	0.
27	18.9501	1.1840	0.
28	19.6914	1.1840	0.
29	20.4327	1.1840	0.
30	21.1740	1.1840	0.
31	21.9153	1.1840	0.
32	22.6566	1.1840	0.
33	23.3979	1.1840	0.
34	24.1392	1.1840	0.
35	24.8805	1.1840	0.
36	25.6218	1.1840	0.
37	26.3631	1.1840	0.
38	27.1044	1.1840	0.
39	27.8457	1.1840	0.
40	28.5870	1.1840	0.
41	29.3283	1.1840	0.
42	30.0696	1.1825	0.
43	30.8109	1.1757	0.
44	31.5522	1.1630	0.
45	32.2935	1.1452	0.
46	33.0343	1.1226	0.
47	33.7761	1.0930	0.
48	34.5174	1.0581	0.
49	35.2587	1.0252	0.
50	36.0000	1.0000	0.

BODY PANEL

4P

BCDY PANEL ROUTINE

THERE ARE 15 TRANSVERSE VERTICAL PLANES THAT INTERSECT THE BODY
TO DEFINE PANEL LEADING AND TRAILING EDGES

X-INTERCEPTS

13.76097
14.75289
15.69159
16.61262
17.53366
18.45469
19.37572
20.29676
21.21779
22.13882
23.05986
25.00000
27.00000
29.50000
32.41500

BODY PANEL CORNER POINT COORDINATES
1 AND 2 INDICATE BODY PANEL LEADING-EDGE POINTS, 3 AND 4 INDICATE TRAILING-EDGE POINTS

PANEL NO	PARTS	X 1	Y 1	Z 1	X 2	Y 2	Z 2	X 3	Y 3	Z 3	X 4	Y 4	Z 4
1	1	13.76097	0.	1.15418	13.76097	0.48778	1.04604	14.75289	0.00504	1.17693	14.75289	0.49282	1.06880
2	1	14.75289	0.	1.17805	14.75289	0.49787	1.06768	15.69159	0.00126	1.18372	15.69159	0.49912	1.07335
3	1	15.69159	0.	1.18400	15.69159	0.50038	1.07307	16.61262	0.	1.18400	16.61262	0.50038	1.07307
4	1	16.61262	0.	1.18400	16.61262	0.50038	1.07307	17.53366	0.	1.18400	17.53366	0.50038	1.07307
5	1	17.53366	0.	1.18400	17.53366	0.50038	1.07307	18.45469	0.	1.18400	18.45469	0.50038	1.07307
6	1	18.45469	0.	1.18400	18.45469	0.50038	1.07307	19.37572	0.	1.18400	19.37572	0.50038	1.07307
7	1	19.37572	0.	1.18400	19.37572	0.50038	1.07307	20.29676	0.	1.18400	20.29676	0.50038	1.07307
8	1	20.29676	0.	1.18400	20.29676	0.50038	1.07307	21.21779	0.	1.18400	21.21779	0.50038	1.07307
9	1	21.21779	0.	1.18400	21.21779	0.50038	1.07307	22.13882	0.	1.18400	22.13882	0.50038	1.07307
10	1	22.13882	0.	1.18400	22.13882	0.50038	1.07307	23.05986	0.	1.18400	23.05986	0.50038	1.07307
11	1	23.05986	0.	1.18400	23.05986	0.50038	1.07307	25.00000	0.	1.18400	25.00000	0.50038	1.07307
12	1	25.00000	0.	1.18400	25.00000	0.50038	1.07307	27.00000	0.	1.18400	27.00000	0.50038	1.07307
13	1	27.00000	0.	1.18400	27.00000	0.50038	1.07307	29.50000	0.	1.18400	29.50000	0.50038	1.07307
14	1	29.50000	0.00897	1.18201	29.50000	0.49141	1.07506	32.41500	0.	1.14155	32.41500	0.48244	1.03460
15	1	13.76097	0.48778	1.04604	13.76097	0.88415	0.74189	14.75289	0.50197	1.06453	14.75289	0.89934	0.76038
16	1	14.75289	0.49787	1.06768	14.75289	0.90244	0.75724	15.69159	0.50140	1.07228	15.69159	0.90598	0.76184
17	1	15.69159	0.50038	1.07307	15.69159	0.90700	0.76106	16.61262	0.50038	1.07307	16.61262	0.90700	0.76106
18	1	16.61262	0.50038	1.07307	16.61262	0.90700	0.76106	17.53366	0.50038	1.07307	17.53366	0.90700	0.76106
19	1	17.53366	0.50038	1.07307	17.53366	0.90700	0.76106	18.45469	0.50038	1.07307	18.45469	0.90700	0.76106
20	1	18.45469	0.50038	1.07307	18.45469	0.90700	0.76106	19.37572	0.50038	1.07307	19.37572	0.90700	0.76106
21	1	19.37572	0.50038	1.07307	19.37572	0.90700	0.76106	20.29676	0.50038	1.07307	20.29676	0.90700	0.76106
22	1	20.29676	0.50038	1.07307	20.29676	0.90700	0.76106	21.21779	0.50038	1.07307	21.21779	0.90700	0.76106
23	1	21.21779	0.50038	1.07307	21.21779	0.90700	0.76106	22.13882	0.50038	1.07307	22.13882	0.90700	0.76106
24	1	22.13882	0.50038	1.07307	22.13882	0.90700	0.76106	23.05986	0.50038	1.07307	23.05986	0.90700	0.76106
25	1	23.05986	0.50038	1.07307	23.05986	0.90700	0.76106	25.00000	0.50038	1.07307	25.00000	0.90700	0.76106
26	1	25.00000	0.50038	1.07307	25.00000	0.90700	0.76106	27.00000	0.50038	1.07307	27.00000	0.90700	0.76106
27	1	27.00000	0.50038	1.07307	27.00000	0.90700	0.76106	29.50000	0.50038	1.07307	29.50000	0.90700	0.76106
28	1	29.50000	0.50767	1.06748	29.50000	0.89971	0.76665	32.41500	0.48244	1.03460	32.41500	0.87448	0.73377
29	1	13.76097	0.88415	0.74189	13.76097	1.11485	0.29872	14.75289	0.90483	0.75265	14.75289	1.13553	0.30949
30	1	14.75289	0.90244	0.75724	14.75289	1.13791	0.30490	15.69159	0.90759	0.75992	15.69159	1.14306	0.30758
31	1	15.69159	0.90700	0.76106	15.69159	1.14366	0.30644	16.61262	0.90700	0.76106	16.61262	1.14366	0.30644
32	1	16.61262	0.90700	0.76106	16.61262	1.14366	0.30644	17.53366	0.90700	0.76106	17.53366	1.14366	0.30644
33	1	17.53366	0.90700	0.76106	17.53366	1.14366	0.30644	18.45469	0.90700	0.76106	18.45469	1.14366	0.30644
34	1	18.45469	0.90700	0.76106	18.45469	1.14366	0.30644	19.37572	0.90700	0.76106	19.37572	1.14366	0.30644
35	1	19.37572	0.90700	0.76106	19.37572	1.14366	0.30644	20.29676	0.90700	0.76106	20.29676	1.14366	0.30644
36	1	20.29676	0.90700	0.76106	20.29676	1.14366	0.30644	21.21779	0.90700	0.76106	21.21779	1.14366	0.30644
37	1	21.21779	0.90700	0.76106	21.21779	1.14366	0.30644	22.13882	0.90700	0.76106	22.13882	1.14366	0.30644
38	1	22.13882	0.90700	0.76106	22.13882	1.14366	0.30644	23.05986	0.90700	0.76106	23.05986	1.14366	0.30644
39	1	23.05986	0.90700	0.76106	23.05986	1.14366	0.30644	25.00000	0.90700	0.76106	25.00000	1.14366	0.30644
40	1	25.00000	0.90700	0.76106	25.00000	1.14366	0.30644	27.00000	0.90700	0.76106	27.00000	1.14366	0.30644
41	1	27.00000	0.90700	0.76106	27.00000	1.14366	0.30644	29.50000	0.90700	0.76106	29.50000	1.14366	0.30644
42	1	29.50000	0.91124	0.75291	29.50000	1.13941	0.31459	32.41500	0.87448	0.73377	32.41500	1.10265	0.29545
43	1	13.76097	1.11485	0.29872	13.76097	1.12816	-0.24371	14.75289	1.13805	0.29929	14.75289	1.15135	-0.24314
44	1	14.75289	1.13791	0.30490	14.75289	1.15149	-0.24875	15.69159	1.14369	0.30504	15.69159	1.15727	-0.24861
45	1	15.69159	1.14366	0.30644	15.69159	1.15730	-0.25001	16.61262	1.14366	0.30644	16.61262	1.15730	-0.25001
46	1	16.61262	1.14366	0.30644	16.61262	1.15730	-0.25001	17.53366	1.14366	0.30644	17.53366	1.15730	-0.25001
47	1	17.53366	1.14366	0.30644	17.53366	1.15730	-0.25001	18.45469	1.14366	0.30644	18.45469	1.15730	-0.25001
48	1	18.45469	1.14366	0.30644	18.45469	1.15730	-0.25001	19.37572	1.14366	0.30644	19.37572	1.15730	-0.25001
49	1	19.37572	1.14366	0.30644	19.37572	1.15730	-0.25001	20.29676	1.14366	0.30644	20.29676	1.15730	-0.25001
50	1	20.29676	1.14366	0.30644	20.29676	1.15730	-0.25001	21.21779	1.14366	0.30644	21.21779	1.15730	-0.25001
51	1	21.21779	1.14366	0.30644	21.21779	1.15730	-0.25001	22.13882	1.14366	0.30644	22.13882	1.15730	-0.25001
52	1	22.13882	1.14366	0.30644	22.13882	1.15730	-0.25001	23.05986	1.14366	0.30644	23.05986	1.15730	-0.25001

53	1	23.05986	1.14366	0.30644	23.05986	1.15730	-0.25001	25.00000	1.14366	0.30644	25.00000	1.15730	-0.25001
54	1	25.00000	1.14366	0.30644	25.00000	1.15730	-0.25001	27.00000	1.14366	0.30644	27.00000	1.15730	-0.25001
55	1	27.00000	1.14366	0.30644	27.00000	1.15730	-0.25001	29.50000	1.14366	0.30644	29.50000	1.15730	-0.25001
56	1	29.50000	1.14390	0.29647	29.50000	1.15706	-0.24003	32.41500	1.10265	0.29545	32.41500	1.11581	-0.24104
57	1	13.76097	1.12816	-0.24371	13.76097	0.88415	-0.74199	14.75289	1.14897	-0.25390	14.75289	0.90496	-0.75209
58	1	14.75289	1.15149	-0.24375	14.75289	0.90244	-0.75724	15.69159	1.15668	-0.25129	15.69159	0.90763	-0.75978
59	1	15.69159	1.15730	-0.25001	15.69159	0.90700	-0.76106	16.61262	1.15730	-0.25001	16.61262	0.90700	-0.76106
60	1	16.61262	1.15730	-0.25001	16.61262	0.90700	-0.76106	17.53366	1.15730	-0.25001	17.53366	0.90700	-0.76106
61	1	17.53366	1.15730	-0.25001	17.53366	0.90700	-0.76106	18.45469	1.15730	-0.25001	18.45469	0.90700	-0.76106
62	1	18.45469	1.15730	-0.25001	18.45469	0.90700	-0.76106	19.37572	1.15730	-0.25001	19.37572	0.90700	-0.76106
63	1	19.37572	1.15730	-0.25001	19.37572	0.90700	-0.76106	20.29676	1.15730	-0.25001	20.29676	0.90700	-0.76106
64	1	20.29676	1.15730	-0.25001	20.29676	0.90700	-0.76106	21.21779	1.15730	-0.25001	21.21779	0.90700	-0.76106
65	1	21.21779	1.15730	-0.25001	21.21779	0.90700	-0.76106	22.13882	1.15730	-0.25001	22.13882	0.90700	-0.76106
66	1	22.13882	1.15730	-0.25001	22.13882	0.90700	-0.76106	23.05986	1.15730	-0.25001	23.05986	0.90700	-0.76106
67	1	23.05986	1.15730	-0.25001	23.05986	0.90700	-0.76106	25.00000	1.15730	-0.25001	25.00000	0.90700	-0.76106
68	1	25.00000	1.15730	-0.25001	25.00000	0.90700	-0.76106	27.00000	1.15730	-0.25001	27.00000	0.90700	-0.76106
69	1	27.00000	1.15730	-0.25001	27.00000	0.90700	-0.76106	29.50000	1.15730	-0.25001	29.50000	0.90700	-0.76106
70	1	29.50000	1.15282	-0.25917	29.50000	0.91148	-0.75190	32.41500	1.11581	-0.24104	32.41500	0.87448	-0.73377
71	1	13.76097	0.88415	-0.74189	13.76097	0.48778	-1.04604	14.75289	0.89834	-0.76038	14.75289	0.50197	-1.06453
72	1	14.75289	0.90244	-0.75724	14.75289	0.49787	-1.06768	15.69159	0.90598	-0.76184	15.69159	0.50140	-1.07228
73	1	15.69159	0.90700	-0.76106	15.69159	0.50038	-1.07307	16.61262	0.90700	-0.76106	16.61262	0.50038	-1.07307
74	1	16.61262	0.90700	-0.76106	16.61262	0.50038	-1.07307	17.53366	0.90700	-0.76106	17.53366	0.50038	-1.07307
75	1	17.53366	0.90700	-0.76106	17.53366	0.50038	-1.07307	18.45469	0.90700	-0.76106	18.45469	0.50038	-1.07307
76	1	18.45469	0.90700	-0.76106	18.45469	0.50038	-1.07307	19.37572	0.90700	-0.76106	19.37572	0.50038	-1.07307
77	1	19.37572	0.90700	-0.76106	19.37572	0.50038	-1.07307	20.29676	0.90700	-0.76106	20.29676	0.50038	-1.07307
78	1	20.29676	0.90700	-0.76106	20.29676	0.50038	-1.07307	21.21779	0.90700	-0.76106	21.21779	0.50038	-1.07307
79	1	21.21779	0.90700	-0.76106	21.21779	0.50038	-1.07307	22.13882	0.90700	-0.76106	22.13882	0.50038	-1.07307
80	1	22.13882	0.90700	-0.76106	22.13882	0.50038	-1.07307	23.05986	0.90700	-0.76106	23.05986	0.50038	-1.07307
81	1	23.05986	0.90700	-0.76106	23.05986	0.50038	-1.07307	25.00000	0.90700	-0.76106	25.00000	0.50038	-1.07307
82	1	25.00000	0.90700	-0.76106	25.00000	0.50038	-1.07307	27.00000	0.90700	-0.76106	27.00000	0.50038	-1.07307
83	1	27.00000	0.90700	-0.76106	27.00000	0.50038	-1.07307	29.50000	0.90700	-0.76106	29.50000	0.50038	-1.07307
84	1	29.50000	0.89971	-0.76665	29.50000	0.50767	-1.06748	32.41500	0.87448	-0.73377	32.41500	0.48244	-1.03460
85	1	13.76097	0.48778	-1.04604	13.76097	0.	-1.15418	14.75289	0.49282	-1.06880	14.75289	0.00504	-1.17693
86	1	14.75289	0.49787	-1.06768	14.75289	0.	-1.17805	15.69159	0.49912	-1.07335	15.69159	0.00126	-1.18372
87	1	15.69159	0.50038	-1.07307	15.69159	0.	-1.18400	16.61262	0.50038	-1.07307	16.61262	0.	-1.18400
88	1	16.61262	0.50038	-1.07307	16.61262	0.	-1.18400	17.53366	0.50038	-1.07307	17.53366	0.	-1.18400
89	1	17.53366	0.50038	-1.07307	17.53366	0.	-1.18400	18.45469	0.50038	-1.07307	18.45469	0.	-1.18400
90	1	18.45469	0.50038	-1.07307	18.45469	0.	-1.18400	19.37572	0.50038	-1.07307	19.37572	0.	-1.18400
91	1	19.37572	0.50038	-1.07307	19.37572	0.	-1.18400	20.29676	0.50038	-1.07307	20.29676	0.	-1.18400
92	1	20.29676	0.50038	-1.07307	20.29676	0.	-1.18400	21.21779	0.50038	-1.07307	21.21779	0.	-1.18400
93	1	21.21779	0.50038	-1.07307	21.21779	0.	-1.18400	22.13882	0.50038	-1.07307	22.13882	0.	-1.18400
94	1	22.13882	0.50038	-1.07307	22.13882	0.	-1.18400	23.05986	0.50038	-1.07307	23.05986	0.	-1.18400
95	1	23.05986	0.50038	-1.07307	23.05986	0.	-1.18400	25.00000	0.50038	-1.07307	25.00000	0.	-1.18400
96	1	25.00000	0.50038	-1.07307	25.00000	0.	-1.18400	27.00000	0.50038	-1.07307	27.00000	0.	-1.18400
97	1	27.00000	0.50038	-1.07307	27.00000	0.	-1.18400	29.50000	0.50038	-1.07307	29.50000	0.	-1.18400
98	1	29.50000	0.49141	-1.07506	29.50000	0.00897	-1.18201	32.41500	0.48244	-1.03460	32.41500	0.	-1.14155

BODY PANEL CENTROID AND CONTROL POINT COORDINATES

PANEL	X C	Y C	Z C	X CP	Y CP	Z CP	AREA	THETA-INCLIN	ALPHA-INCID
1	14.25863	0.24642	1.11153	14.70330	0.24868	1.12173	0.50085	-0.21817	0.02349
2	15.22264	0.24956	1.12570	15.64465	0.25013	1.12825	0.47991	-0.21817	0.00619
3	16.15210	0.25019	1.12853	16.56657	0.25019	1.12853	0.47206	-0.21817	0.
4	17.07314	0.25019	1.12853	17.48760	0.25019	1.12853	0.47206	-0.21817	0.
5	17.99417	0.25019	1.12853	18.40864	0.25019	1.12853	0.47206	-0.21817	0.
6	18.91521	0.25019	1.12853	19.32967	0.25019	1.12853	0.47206	-0.21817	0.
7	19.83624	0.25019	1.12853	20.25070	0.25019	1.12853	0.47206	-0.21817	0.
8	20.75727	0.25019	1.12853	21.17174	0.25019	1.12853	0.47206	-0.21817	0.
9	21.67831	0.25019	1.12853	22.09277	0.25019	1.12853	0.47206	-0.21817	0.
10	22.59934	0.25019	1.12853	23.01381	0.25019	1.12853	0.47206	-0.21817	0.
11	24.02993	0.25019	1.12853	24.90299	0.25019	1.12853	0.99438	-0.21817	0.
12	26.00000	0.25019	1.12853	26.90000	0.25019	1.12853	1.02506	-0.21817	0.
13	28.25000	0.25019	1.12853	29.37500	0.25019	1.12853	1.28132	-0.21817	0.
14	30.94863	0.24573	1.10843	32.26925	0.24167	1.09010	1.46739	-0.21817	-0.01422
15	14.25863	0.69308	0.90324	14.70330	0.69944	0.91153	0.50085	-0.65450	0.02349
16	15.22264	0.70192	0.91476	15.64465	0.70351	0.91683	0.47991	-0.65450	0.00619
17	16.15210	0.70369	0.91706	16.56657	0.70369	0.91706	0.47206	-0.65450	0.
18	17.07314	0.70369	0.91706	17.48760	0.70369	0.91706	0.47206	-0.65450	0.
19	17.99417	0.70369	0.91706	18.40864	0.70369	0.91706	0.47206	-0.65450	0.
20	18.91521	0.70369	0.91706	19.32967	0.70369	0.91706	0.47206	-0.65450	0.
21	19.83624	0.70369	0.91706	20.25070	0.70369	0.91706	0.47206	-0.65450	0.
22	20.75727	0.70369	0.91706	21.17174	0.70369	0.91706	0.47206	-0.65450	0.
23	21.67831	0.70369	0.91706	22.09277	0.70369	0.91706	0.47206	-0.65450	0.
24	22.59934	0.70369	0.91706	23.01381	0.70369	0.91706	0.47206	-0.65450	0.
25	24.02993	0.70369	0.91706	24.90299	0.70369	0.91706	0.99438	-0.65450	0.
26	26.00000	0.70369	0.91706	26.90000	0.70369	0.91706	1.02506	-0.65450	0.
27	28.25000	0.70369	0.91706	29.37500	0.70369	0.91706	1.28132	-0.65450	0.
28	30.94863	0.69115	0.90072	32.26925	0.67972	0.88583	1.46739	-0.65450	-0.01422
29	14.25863	1.00398	0.52571	14.70330	1.01914	0.53053	0.50085	-1.09083	0.02349
30	15.22264	1.02275	0.53241	15.64465	1.02507	0.53362	0.47991	-1.09083	0.00619
31	16.15210	1.02533	0.53375	16.56657	1.02533	0.53375	0.47206	-1.09083	0.
32	17.07314	1.02533	0.53375	17.48760	1.02533	0.53375	0.47206	-1.09083	0.
33	17.99417	1.02533	0.53375	18.40864	1.02533	0.53375	0.47206	-1.09083	0.
34	18.91521	1.02533	0.53375	19.32967	1.02533	0.53375	0.47206	-1.09083	0.
35	19.83624	1.02533	0.53375	20.25070	1.02533	0.53375	0.47206	-1.09083	0.
36	20.75727	1.02533	0.53375	21.17174	1.02533	0.53375	0.47206	-1.09083	0.
37	21.67831	1.02533	0.53375	22.09277	1.02533	0.53375	0.47206	-1.09083	0.
38	22.59934	1.02533	0.53375	23.01381	1.02533	0.53375	0.47206	-1.09083	0.
39	24.02993	1.02533	0.53375	24.90299	1.02533	0.53375	0.99438	-1.09083	0.
40	26.00000	1.02533	0.53375	26.90000	1.02533	0.53375	1.02506	-1.09083	0.
41	28.25000	1.02533	0.53375	29.37500	1.02533	0.53375	1.28132	-1.09083	0.
42	30.94863	1.00706	0.52424	32.26925	0.99040	0.51557	1.46739	-1.09083	-0.01422
43	14.25863	1.13314	0.02779	14.70330	1.14354	0.02805	0.54393	-1.54627	0.02339
44	15.22264	1.14759	0.02815	15.64465	1.15019	0.02821	0.52119	-1.54627	0.00616
45	16.15210	1.15048	0.02822	16.56657	1.15048	0.02822	0.51266	-1.54627	0.
46	17.07314	1.15048	0.02822	17.48760	1.15048	0.02822	0.51266	-1.54627	0.
47	17.99417	1.15048	0.02822	18.40864	1.15048	0.02822	0.51266	-1.54627	0.
48	18.91521	1.15048	0.02822	19.32967	1.15048	0.02822	0.51266	-1.54627	0.
49	19.83624	1.15048	0.02822	20.25070	1.15048	0.02822	0.51266	-1.54627	0.
50	20.75727	1.15048	0.02822	21.17174	1.15048	0.02822	0.51266	-1.54627	0.
51	21.67831	1.15048	0.02822	22.09277	1.15048	0.02822	0.51266	-1.54627	0.
52	22.59934	1.15048	0.02822	23.01381	1.15048	0.02822	0.51266	-1.54627	0.

53	24.02993	1.15048	0.02822	24.90299	1.15048	0.02822	1.07991	-1.54627	0.
54	26.00000	1.15048	0.02822	26.90000	1.15048	0.02822	1.11323	-1.54627	0.
55	28.25000	1.15048	0.02822	29.37500	1.15048	0.02822	1.39154	-1.54627	0.
56	30.94863	1.12998	0.02771	32.26925	1.11129	0.02726	1.59361	-1.54627	-0.01415
57	14.25863	1.01660	-0.49792	14.70330	1.02593	-0.50248	0.55609	-2.02624	0.02336
58	15.22264	1.02956	-0.50426	15.64465	1.03189	-0.50541	0.53284	-2.02624	0.00615
59	16.15210	1.03215	-0.50553	16.56657	1.03215	-0.50553	0.52412	-2.02624	0.
60	17.07314	1.03215	-0.50553	17.48760	1.03215	-0.50553	0.52412	-2.02624	0.
61	17.99417	1.03215	-0.50553	18.40864	1.03215	-0.50553	0.52412	-2.02624	0.
62	18.91521	1.03215	-0.50553	19.32967	1.03215	-0.50553	0.52412	-2.02624	0.
63	19.83624	1.03215	-0.50553	20.25070	1.03215	-0.50553	0.52412	-2.02624	0.
64	20.75727	1.03215	-0.50553	21.17174	1.03215	-0.50553	0.52412	-2.02624	0.
65	21.67831	1.03215	-0.50553	22.09277	1.03215	-0.50553	0.52412	-2.02624	0.
66	22.59934	1.03215	-0.50553	23.01381	1.03215	-0.50553	0.52412	-2.02624	0.
67	24.02993	1.03215	-0.50553	24.90299	1.03215	-0.50553	1.10406	-2.02624	0.
68	26.00000	1.03215	-0.50553	26.90000	1.03215	-0.50553	1.13812	-2.02624	0.
69	28.25000	1.03215	-0.50553	29.37500	1.03215	-0.50553	1.42265	-2.02624	0.
70	30.94863	1.01375	-0.49653	32.26925	0.99699	-0.48831	1.62924	-2.02624	-0.01413
71	14.25863	0.69308	-0.90324	14.70330	0.69944	-0.91153	0.50085	-2.48709	0.02349
72	15.22264	0.70192	-0.91476	15.64465	0.70351	-0.91683	0.47991	-2.48709	0.00619
73	16.15210	0.70369	-0.91706	16.56657	0.70369	-0.91706	0.47206	-2.48709	0.
74	17.07314	0.70369	-0.91706	17.48760	0.70369	-0.91706	0.47206	-2.48709	0.
75	17.99417	0.70369	-0.91706	18.40864	0.70369	-0.91706	0.47206	-2.48709	0.
76	18.91521	0.70369	-0.91706	19.32967	0.70369	-0.91706	0.47206	-2.48709	0.
77	19.83624	0.70369	-0.91706	20.25070	0.70369	-0.91706	0.47206	-2.48709	0.
78	20.75727	0.70369	-0.91706	21.17174	0.70369	-0.91706	0.47206	-2.48709	0.
79	21.67831	0.70369	-0.91706	22.09277	0.70369	-0.91706	0.47206	-2.48709	0.
80	22.59934	0.70369	-0.91706	23.01381	0.70369	-0.91706	0.47206	-2.48709	0.
81	24.02993	0.70369	-0.91706	24.90299	0.70369	-0.91706	0.99438	-2.48709	0.
82	26.00000	0.70369	-0.91706	26.90000	0.70369	-0.91706	1.02506	-2.48709	0.
83	28.25000	0.70369	-0.91706	29.37500	0.70369	-0.91706	1.28132	-2.48709	0.
84	30.94863	0.69115	-0.90072	32.26925	0.67972	-0.88583	1.46739	-2.48709	-0.01422
85	14.25863	0.24642	-1.11153	14.70330	0.24868	-1.12173	0.50085	-2.92343	0.02349
86	15.22264	0.24956	-1.12570	15.64465	0.25013	-1.12825	0.47991	-2.92343	0.00619
87	16.15210	0.25019	-1.12853	16.56657	0.25019	-1.12853	0.47206	-2.92343	0.
88	17.07314	0.25019	-1.12853	17.48760	0.25019	-1.12853	0.47206	-2.92343	0.
89	17.99417	0.25019	-1.12853	18.40864	0.25019	-1.12853	0.47206	-2.92343	0.
90	18.91521	0.25019	-1.12853	19.32967	0.25019	-1.12853	0.47206	-2.92343	0.
91	19.83624	0.25019	-1.12853	20.25070	0.25019	-1.12853	0.47206	-2.92343	0.
92	20.75727	0.25019	-1.12853	21.17174	0.25019	-1.12853	0.47206	-2.92343	0.
93	21.67831	0.25019	-1.12853	22.09277	0.25019	-1.12853	0.47206	-2.92343	0.
94	22.59934	0.25019	-1.12853	23.01381	0.25019	-1.12853	0.47206	-2.92343	0.
95	24.02993	0.25019	-1.12853	24.90299	0.25019	-1.12853	0.99438	-2.92343	0.
96	26.00000	0.25019	-1.12853	26.90000	0.25019	-1.12853	1.02506	-2.92343	0.
97	28.25000	0.25019	-1.12853	29.37500	0.25019	-1.12853	1.28132	-2.92343	0.
98	30.94863	0.24573	-1.10843	32.26925	0.24157	-1.09010	1.46739	-2.92343	-0.01422

WING PANEL 8P

```
      WING PANEL ROUTINE

THERE ARE  10 VERTICAL PLANES THAT INTERSECT THE WING TO DEFINE PANEL SIDE EDGES

LEADING-EDGE
Y-INTERCEPT

   1.70000
   2.62000
   3.54000
   4.46000
   5.38000
   6.30000
   7.22000
   8.14000
   8.60300

          OUTBOARD CUTTING PLANE
       LEADING-EDGE
X-INTERCEPT      Y-INTERCEPT         SLOPE
 40.27000         10.77400           -1.49608

ZERO-SLOPE INDICATES STREAMWISE CUTTING PLANE
```

WING PANEL CORNER POINT COORDINATES
1 AND 2 INDICATE WING PANEL LEADING-EDGE POINTS, 3 AND 4 INDICATE TRAILING-EDGE POINTS

PANEL NO	PARTS	X 1	Y 1	Z 1	X 2	Y 2	Z 2	X 3	Y 3	Z 3	X 4	Y 4	Z 4
1	1	13.83186	1.15087	-0.25001	15.34050	1.70000	-0.25001	14.75289	1.15087	-0.25001	16.26155	1.70000	-0.25001
2	1	14.77055	1.15730	-0.25001	16.26155	1.70000	-0.25001	15.69159	1.15730	-0.25001	17.18260	1.70000	-0.25001
3	1	15.69159	1.15730	-0.25001	17.18260	1.70000	-0.25001	16.61262	1.15730	-0.25001	18.10365	1.70000	-0.25001
4	1	16.61262	1.15730	-0.25001	18.10365	1.70000	-0.25001	17.53366	1.15730	-0.25001	19.02470	1.70000	-0.25001
5	1	17.53366	1.15730	-0.25001	19.02470	1.70000	-0.25001	18.45469	1.15730	-0.25001	19.94575	1.70000	-0.25001
6	1	18.45469	1.15730	-0.25001	19.94575	1.70000	-0.25001	19.37572	1.15730	-0.25001	20.86680	1.70000	-0.25001
7	1	19.37572	1.15730	-0.25001	20.86680	1.70000	-0.25001	20.29676	1.15730	-0.25001	21.78785	1.70000	-0.25001
8	1	20.29676	1.15730	-0.25001	21.78785	1.70000	-0.25001	21.21779	1.15730	-0.25001	22.70890	1.70000	-0.25001
9	1	21.21779	1.15730	-0.25001	22.70890	1.70000	-0.25001	22.13882	1.15730	-0.25001	23.62995	1.70000	-0.25001
10	1	22.13882	1.15730	-0.25001	23.62995	1.70000	-0.25001	23.05986	1.15730	-0.25001	24.55100	1.70000	-0.25001
11	1	15.34050	1.70000	-0.25001	17.86807	2.62000	-0.25001	16.26155	1.70000	-0.25001	18.78915	2.62000	-0.25001
12	1	16.26155	1.70000	-0.25001	18.78915	2.62000	-0.25001	17.18260	1.70000	-0.25001	19.71022	2.62000	-0.25001
13	1	17.18260	1.70000	-0.25001	19.71022	2.62000	-0.25001	18.10365	1.70000	-0.25001	20.63130	2.62000	-0.25001
14	1	18.10365	1.70000	-0.25001	20.63130	2.62000	-0.25001	19.02470	1.70000	-0.25001	21.55237	2.62000	-0.25001
15	1	19.02470	1.70000	-0.25001	21.55237	2.62000	-0.25001	19.94575	1.70000	-0.25001	22.47345	2.62000	-0.25001
16	1	19.94575	1.70000	-0.25001	22.47345	2.62000	-0.25001	20.86680	1.70000	-0.25001	23.39453	2.62000	-0.25001
17	1	20.86680	1.70000	-0.25001	23.39453	2.62000	-0.25001	21.78785	1.70000	-0.25001	24.31560	2.62000	-0.25001
18	1	21.78785	1.70000	-0.25001	24.31560	2.62000	-0.25001	22.70890	1.70000	-0.25001	25.23668	2.62000	-0.25001
19	1	22.70890	1.70000	-0.25001	25.23668	2.62000	-0.25001	23.62995	1.70000	-0.25001	26.15776	2.62000	-0.25001
20	1	23.62995	1.70000	-0.25001	26.15776	2.62000	-0.25001	24.55100	1.70000	-0.25001	27.07883	2.62000	-0.25001
21	1	17.86807	2.62000	-0.25001	20.39564	3.54000	-0.25001	18.78915	2.62000	-0.25001	21.31674	3.54000	-0.25001
22	1	18.78915	2.62000	-0.25001	21.31674	3.54000	-0.25001	19.71022	2.62000	-0.25001	22.23784	3.54000	-0.25001
23	1	19.71022	2.62000	-0.25001	22.23784	3.54000	-0.25001	20.63130	2.62000	-0.25001	23.15894	3.54000	-0.25001
24	1	20.63130	2.62000	-0.25001	23.15894	3.54000	-0.25001	21.55237	2.62000	-0.25001	24.08005	3.54000	-0.25001
25	1	21.55237	2.62000	-0.25001	24.08005	3.54000	-0.25001	22.47345	2.62000	-0.25001	25.00115	3.54000	-0.25001
26	1	22.47345	2.62000	-0.25001	25.00115	3.54000	-0.25001	23.39453	2.62000	-0.25001	25.92225	3.54000	-0.25001
27	1	23.39453	2.62000	-0.25001	25.92225	3.54000	-0.25001	24.31560	2.62000	-0.25001	26.84336	3.54000	-0.25001
28	1	24.31560	2.62000	-0.25001	26.84336	3.54000	-0.25001	25.23668	2.62000	-0.25001	27.76446	3.54000	-0.25001
29	1	25.23668	2.62000	-0.25001	27.76446	3.54000	-0.25001	26.15776	2.62000	-0.25001	28.68556	3.54000	-0.25001
30	1	26.15776	2.62000	-0.25001	28.68556	3.54000	-0.25001	27.07883	2.62000	-0.25001	29.60667	3.54000	-0.25001
31	1	20.39564	3.54000	-0.25001	22.92320	4.46000	-0.25001	21.31674	3.54000	-0.25001	23.84433	4.46000	-0.25001
32	1	21.31674	3.54000	-0.25001	23.84433	4.46000	-0.25001	22.23784	3.54000	-0.25001	24.76546	4.46000	-0.25001
33	1	22.23784	3.54000	-0.25001	24.76546	4.46000	-0.25001	23.15894	3.54000	-0.25001	25.68659	4.46000	-0.25001
34	1	23.15894	3.54000	-0.25001	25.68659	4.46000	-0.25001	24.08005	3.54000	-0.25001	26.60772	4.46000	-0.25001
35	1	24.08005	3.54000	-0.25001	26.60772	4.46000	-0.25001	25.00115	3.54000	-0.25001	27.52885	4.46000	-0.25001
36	1	25.00115	3.54000	-0.25001	27.52885	4.46000	-0.25001	25.92225	3.54000	-0.25001	28.44998	4.46000	-0.25001
37	1	25.92225	3.54000	-0.25001	28.44998	4.46000	-0.25001	26.84336	3.54000	-0.25001	29.37111	4.46000	-0.25001
38	1	26.84336	3.54000	-0.25001	29.37111	4.46000	-0.25001	27.76446	3.54000	-0.25001	30.29224	4.46000	-0.25001
39	1	27.76446	3.54000	-0.25001	30.29224	4.46000	-0.25001	28.68556	3.54000	-0.25001	31.21337	4.46000	-0.25001
40	1	28.68556	3.54000	-0.25001	31.21337	4.46000	-0.25001	29.60667	3.54000	-0.25001	32.13450	4.46000	-0.25001
41	1	22.92320	4.46000	-0.25001	25.45077	5.38000	-0.25001	23.84433	4.46000	-0.25001	26.37192	5.38000	-0.25001
42	1	23.84433	4.46000	-0.25001	26.37192	5.38000	-0.25001	24.76546	4.46000	-0.25001	27.29307	5.38000	-0.25001
43	1	24.76546	4.46000	-0.25001	27.29307	5.38000	-0.25001	25.68659	4.46000	-0.25001	28.21422	5.38000	-0.25001
44	1	25.68659	4.46000	-0.25001	28.21422	5.38000	-0.25001	26.60772	4.46000	-0.25001	29.13538	5.38000	-0.25001
45	1	26.60772	4.46000	-0.25001	29.13538	5.38000	-0.25001	27.52885	4.46000	-0.25001	30.05653	5.38000	-0.25001
46	1	27.52885	4.46000	-0.25001	30.05653	5.38000	-0.25001	28.44998	4.46000	-0.25001	30.97769	5.38000	-0.25001
47	1	28.44998	4.46000	-0.25001	30.97769	5.38000	-0.25001	29.37111	4.46000	-0.25001	31.89885	5.38000	-0.25001
48	1	29.37111	4.46000	-0.25001	31.89885	5.38000	-0.25001	30.29224	4.46000	-0.25001	32.82001	5.38000	-0.25001
49	1	30.29224	4.46000	-0.25001	32.82001	5.38000	-0.25001	31.21337	4.46000	-0.25001	33.74117	5.38000	-0.25001
50	1	31.21337	4.46000	-0.25001	33.74117	5.38000	-0.25001	32.13450	4.46000	-0.25001	34.66234	5.38000	-0.25001
51	1	25.45077	5.38000	-0.25001	27.97833	6.30000	-0.25001	26.37192	5.38000	-0.25001	28.89950	6.30000	-0.25001

52	1	26.37192	5.38000	-0.25001	28.89950	6.30000	-0.25001	27.29307	5.38000	-0.25001	29.82067	6.30000	-0.25001
53	1	27.29307	5.38000	-0.25001	29.82067	6.30000	-0.25001	28.21422	5.38000	-0.25001	30.74184	6.30000	-0.25001
54	1	28.21422	5.38000	-0.25001	30.74184	6.30000	-0.25001	29.13538	5.38000	-0.25001	31.66301	6.30000	-0.25001
55	1	29.13538	5.38000	-0.25001	31.66301	6.30000	-0.25001	30.05653	5.38000	-0.25001	32.58419	6.30000	-0.25001
56	1	30.05653	5.38000	-0.25001	32.58419	6.30000	-0.25001	30.97769	5.38000	-0.25001	33.50537	6.30000	-0.25001
57	1	30.97769	5.38000	-0.25001	33.50537	6.30000	-0.25001	31.89885	5.38000	-0.25001	34.42656	6.30000	-0.25001
58	1	31.89885	5.38000	-0.25001	34.42656	6.30000	-0.25001	32.82001	5.38000	-0.25001	35.34775	6.30000	-0.25001
59	1	32.82001	5.38000	-0.25001	35.34775	6.30000	-0.25001	33.74117	5.38000	-0.25001	36.26895	6.30000	-0.25001
60	1	33.74117	5.38000	-0.25001	36.26895	6.30000	-0.25001	34.66234	5.38000	-0.25001	37.19017	6.30000	-0.25001
61	1	27.97833	6.30000	-0.25001	30.50590	7.22000	-0.25001	28.89950	6.30000	-0.25001	31.42708	7.22000	-0.25001
62	1	28.89950	6.30000	-0.25001	31.42708	7.22000	-0.25001	29.82067	6.30000	-0.25001	32.34826	7.22000	-0.25001
63	1	29.82067	6.30000	-0.25001	32.34826	7.22000	-0.25001	30.74184	6.30000	-0.25001	33.26945	7.22000	-0.25001
64	1	30.74184	6.30000	-0.25001	33.26945	7.22000	-0.25001	31.66301	6.30000	-0.25001	34.19064	7.22000	-0.25001
65	1	31.66301	6.30000	-0.25001	34.19064	7.22000	-0.25001	32.58419	6.30000	-0.25001	35.11184	7.22000	-0.25001
66	1	32.58419	6.30000	-0.25001	35.11184	7.22000	-0.25001	33.50537	6.30000	-0.25001	36.03305	7.22000	-0.25001
67	1	33.50537	6.30000	-0.25001	36.03305	7.22000	-0.25001	34.42656	6.30000	-0.25001	36.95426	7.22000	-0.25001
68	1	34.42656	6.30000	-0.25001	36.95426	7.22000	-0.25001	35.34775	6.30000	-0.25001	37.87549	7.22000	-0.25001
69	1	35.34775	6.30000	-0.25001	37.87549	7.22000	-0.25001	36.26895	6.30000	-0.25001	38.79674	7.22000	-0.25001
70	1	36.26895	6.30000	-0.25001	38.79674	7.22000	-0.25001	37.19017	6.30000	-0.25001	39.71800	7.22000	-0.25001
71	1	30.50590	7.22000	-0.25001	33.03347	8.14000	-0.25001	31.42708	7.22000	-0.25001	33.95466	8.14000	-0.25001
72	1	31.42708	7.22000	-0.25001	33.95466	8.14000	-0.25001	32.34826	7.22000	-0.25001	34.87586	8.14000	-0.25001
73	1	32.34826	7.22000	-0.25001	34.87586	8.14000	-0.25001	33.26945	7.22000	-0.25001	35.79706	8.14000	-0.25001
74	1	33.26945	7.22000	-0.25001	35.79706	8.14000	-0.25001	34.19064	7.22000	-0.25001	36.71827	8.14000	-0.25001
75	1	34.19064	7.22000	-0.25001	36.71827	8.14000	-0.25001	35.11184	7.22000	-0.25001	37.63949	8.14000	-0.25001
76	1	35.11184	7.22000	-0.25001	37.63949	8.14000	-0.25001	36.03305	7.22000	-0.25001	38.56073	8.14000	-0.25001
77	1	36.03305	7.22000	-0.25001	38.56073	8.14000	-0.25001	36.95426	7.22000	-0.25001	39.48197	8.14000	-0.25001
78	1	36.95426	7.22000	-0.25001	39.48197	8.14000	-0.25001	37.87549	7.22000	-0.25001	40.40324	8.14000	-0.25001
79	1	37.87549	7.22000	-0.25001	40.40324	8.14000	-0.25001	38.79674	7.22000	-0.25001	41.32452	8.14000	-0.25001
80	1	38.79674	7.22000	-0.25001	41.32452	8.14000	-0.25001	39.71800	7.22000	-0.25001	42.24584	8.14000	-0.25001
81	1	33.03347	8.14000	-0.25001	34.30549	8.60300	-0.25001	33.95466	8.14000	-0.25001	35.22669	8.60300	-0.25001
82	1	33.95466	8.14000	-0.25001	35.22669	8.60300	-0.25001	34.87586	8.14000	-0.25001	36.14790	8.60300	-0.25001
83	1	34.87586	8.14000	-0.25001	36.14790	8.60300	-0.25001	35.79706	8.14000	-0.25001	37.06911	8.60300	-0.25001
84	1	35.79706	8.14000	-0.25001	37.06911	8.60300	-0.25001	36.71827	8.14000	-0.25001	37.99033	8.60300	-0.25001
85	1	36.71827	8.14000	-0.25001	37.99033	8.60300	-0.25001	37.63949	8.14000	-0.25001	38.91156	8.60300	-0.25001
86	1	37.63949	8.14000	-0.25001	38.91156	8.60300	-0.25001	38.56073	8.14000	-0.25001	39.83281	8.60300	-0.25001
87	1	38.56073	8.14000	-0.25001	39.83281	8.60300	-0.25001	39.48197	8.14000	-0.25001	40.75407	8.60300	-0.25001
88	1	39.48197	8.14000	-0.25001	40.75407	8.60300	-0.25001	40.40324	8.14000	-0.25001	41.67537	8.60300	-0.25001
89	1	40.40324	8.14000	-0.25001	41.67537	8.60300	-0.25001	41.32452	8.14000	-0.25001	42.59668	8.60300	-0.25001
90	1	41.32452	8.14000	-0.25001	42.59668	8.60300	-0.25001	42.24584	8.14000	-0.25001	43.51800	8.60300	-0.25001
91	2	34.30549	8.60300	-0.25001	39.67355	10.55690	-0.25001	35.22669	8.60300	-0.25001	40.59480	10.55690	-0.25001
		39.67355	10.55690	-0.25001	40.27000	10.77400	-0.25001	40.59480	10.55690	-0.25001	40.27000	10.77400	-0.25001
92	2	35.22669	8.60300	-0.25001	39.99834	10.33980	-0.25001	36.14790	8.60300	-0.25001	40.91960	10.33980	-0.25001
		39.99834	10.33980	-0.25001	40.59480	10.55690	-0.25001	40.91960	10.33980	-0.25001	40.59480	10.55690	-0.25001
93	2	36.14790	8.60300	-0.25001	40.32314	10.12270	-0.25001	37.06911	8.60300	-0.25001	41.24440	10.12270	-0.25001
		40.32314	10.12270	-0.25001	40.91960	10.33980	-0.25001	41.24440	10.12270	-0.25001	40.91960	10.33980	-0.25001
94	2	37.06911	8.60300	-0.25001	40.64793	9.90560	-0.25001	37.99033	8.60300	-0.25001	41.56920	9.90560	-0.25001
		40.64793	9.90560	-0.25001	41.24440	10.12270	-0.25001	41.56920	9.90560	-0.25001	41.24440	10.12270	-0.25001
95	2	37.99033	8.60300	-0.25001	40.97272	9.68850	-0.25001	38.91156	8.60300	-0.25001	41.89400	9.68850	-0.25001
		40.97272	9.68850	-0.25001	41.56920	9.90560	-0.25001	41.89400	9.68850	-0.25001	41.56920	9.90560	-0.25001
96	2	38.91156	8.60300	-0.25001	41.29751	9.47140	-0.25001	39.83281	8.60300	-0.25001	42.21880	9.47140	-0.25001
		41.29751	9.47140	-0.25001	41.89400	9.68850	-0.25001	42.21880	9.47140	-0.25001	41.89400	9.68850	-0.25001
97	2	39.83281	8.60300	-0.25001	41.62230	9.25430	-0.25001	40.75407	8.60300	-0.25001	42.54360	9.25430	-0.25001
		41.62230	9.25430	-0.25001	42.21880	9.47140	-0.25001	42.54360	9.25430	-0.25001	42.21880	9.47140	-0.25001
98	2	40.75407	8.60300	-0.25001	41.94709	9.03720	-0.25001	41.67537	8.60300	-0.25001	42.86840	9.03720	-0.25001
		41.94709	9.03720	-0.25001	42.54360	9.25430	-0.25001	42.86840	9.03720	-0.25001	42.54360	9.25430	-0.25001
99	2	41.67537	8.60300	-0.25001	42.27188	8.82010	-0.25001	42.59668	8.60300	-0.25001	43.19320	8.82010	-0.25001
		42.27188	8.82010	-0.25001	42.86840	9.03720	-0.25001	43.19320	8.82010	-0.25001	42.86840	9.03720	-0.25001
100	1	42.59668	8.60300	-0.25001	43.19320	8.82010	-0.25001	43.51800	8.60300	-0.25001	43.19320	8.82010	-0.25001

WING PANEL CENTROID AND CONTROL POINT COORDINATES

PANEL	X C	Y C	Z C	X CP	Y CP	Z CP	AREA	Z THICK	ALPHA-THICK	Z CAMBER	ALPHA-CAMBER
1	15.02532	1.41894	-0.25001	15.44332	1.41894	-0.25001	0.51765	-0.16350	0.10829	-0.17193	0.04625
2	15.97125	1.42704	-0.25001	16.38662	1.42704	-0.25001	0.50281	-0.10921	0.05022	-0.14713	0.01667
3	16.89762	1.42865	-0.25001	17.31209	1.42865	-0.25001	0.49985	-0.07970	0.02498	-0.13461	0.01072
4	17.81866	1.42865	-0.25001	18.23313	1.42865	-0.25001	0.49985	-0.06237	0.00695	-0.12665	0.00484
5	18.73970	1.42865	-0.25001	19.15417	1.42865	-0.25001	0.49985	-0.05943	-0.00762	-0.12404	-0.00126
6	19.66074	1.42865	-0.25001	20.07521	1.42865	-0.25001	0.49985	-0.07285	-0.02197	-0.12792	-0.00782
7	20.58178	1.42865	-0.25001	20.99625	1.42865	-0.25001	0.49985	-0.10105	-0.03357	-0.14046	-0.01880
8	21.50283	1.42865	-0.25001	21.91729	1.42865	-0.25001	0.49985	-0.13855	-0.04034	-0.16352	-0.03000
9	22.42387	1.42865	-0.25001	22.83834	1.42865	-0.25001	0.49985	-0.17912	-0.04620	-0.19806	-0.04407
10	23.34491	1.42865	-0.25001	23.75938	1.42865	-0.25001	0.49985	-0.22164	-0.05180	-0.24687	-0.06404
11	17.06482	2.16000	-0.25001	17.47930	2.16000	-0.25001	0.84738	-0.16338	0.10829	-0.17193	0.04625
12	17.98589	2.16000	-0.25001	18.40036	2.16000	-0.25001	0.84738	-0.10918	0.05022	-0.14713	0.01667
13	18.90695	2.16000	-0.25001	19.32143	2.16000	-0.25001	0.84738	-0.07969	0.02498	-0.13461	0.01072
14	19.82801	2.16000	-0.25001	20.24249	2.16000	-0.25001	0.84738	-0.06237	0.00695	-0.12665	0.00484
15	20.74907	2.16000	-0.25001	21.16355	2.16000	-0.25001	0.84738	-0.05943	-0.00762	-0.12404	-0.00126
16	21.67014	2.16000	-0.25001	22.08462	2.16000	-0.25001	0.84738	-0.07285	-0.02197	-0.12792	-0.00782
17	22.59120	2.16000	-0.25001	23.00568	2.16000	-0.25001	0.84738	-0.10105	-0.03357	-0.14046	-0.01880
18	23.51226	2.16000	-0.25001	23.92674	2.16000	-0.25001	0.84738	-0.13855	-0.04034	-0.16351	-0.03000
19	24.43333	2.16000	-0.25001	24.84780	2.16000	-0.25001	0.84738	-0.17912	-0.04620	-0.19806	-0.04407
20	25.35439	2.16000	-0.25001	25.76887	2.16000	-0.25001	0.84738	-0.22164	-0.05180	-0.24687	-0.06404
21	19.59240	3.08000	-0.25001	20.00689	3.08000	-0.25001	0.84740	-0.16338	0.10829	-0.17192	0.04625
22	20.51349	3.08000	-0.25001	20.92798	3.08000	-0.25001	0.84740	-0.10917	0.05022	-0.14713	0.01667
23	21.43458	3.08000	-0.25001	21.84907	3.08000	-0.25001	0.84740	-0.07969	0.02498	-0.13460	0.01072
24	22.35567	3.08000	-0.25001	22.77016	3.08000	-0.25001	0.84740	-0.06236	0.00695	-0.12664	0.00484
25	23.27676	3.08000	-0.25001	23.69125	3.08000	-0.25001	0.84740	-0.05942	-0.00762	-0.12403	-0.00126
26	24.19785	3.08000	-0.25001	24.61234	3.08000	-0.25001	0.84740	-0.07284	-0.02197	-0.12792	-0.00782
27	25.11894	3.08000	-0.25001	25.53343	3.08000	-0.25001	0.84740	-0.10104	-0.03357	-0.14046	-0.01880
28	26.04003	3.08000	-0.25001	26.45452	3.08000	-0.25001	0.84740	-0.13855	-0.04034	-0.16351	-0.03000
29	26.96112	3.08000	-0.25001	27.37561	3.08000	-0.25001	0.84740	-0.17912	-0.04620	-0.19806	-0.04407
30	27.88221	3.08000	-0.25001	28.29670	3.08000	-0.25001	0.84740	-0.22164	-0.05180	-0.24687	-0.06404
31	22.11598	4.00000	-0.25001	22.53448	4.00000	-0.25001	0.84743	-0.16338	0.10829	-0.17192	0.04625
32	23.04110	4.00000	-0.25001	23.45560	4.00000	-0.25001	0.84743	-0.10917	0.05022	-0.14712	0.01667
33	23.96222	4.00000	-0.25001	24.37672	4.00000	-0.25001	0.84743	-0.07968	0.02498	-0.13460	0.01072
34	24.88333	4.00000	-0.25001	25.29783	4.00000	-0.25001	0.84743	-0.06236	0.00695	-0.12664	0.00484
35	25.80445	4.00000	-0.25001	26.21895	4.00000	-0.25001	0.84743	-0.05942	-0.00762	-0.12403	-0.00126
36	26.72557	4.00000	-0.25001	27.14007	4.00000	-0.25001	0.84743	-0.07284	-0.02197	-0.12791	-0.00782
37	27.64668	4.00000	-0.25001	28.06118	4.00000	-0.25001	0.84743	-0.10104	-0.03357	-0.14045	-0.01880
38	28.56780	4.00000	-0.25001	28.98230	4.00000	-0.25001	0.84743	-0.13854	-0.04034	-0.16351	-0.03000
39	29.48891	4.00000	-0.25001	29.90342	4.00000	-0.25001	0.84743	-0.17911	-0.04620	-0.19806	-0.04407
40	30.41003	4.00000	-0.25001	30.82453	4.00000	-0.25001	0.84743	-0.22164	-0.05180	-0.24687	-0.06404
41	24.64756	4.92000	-0.25001	25.06207	4.92000	-0.25001	0.84745	-0.16338	0.10829	-0.17192	0.04625
42	25.56870	4.92000	-0.25001	25.98321	4.92000	-0.25001	0.84745	-0.10917	0.05022	-0.14712	0.01667
43	26.48984	4.92000	-0.25001	26.90436	4.92000	-0.25001	0.84745	-0.07968	0.02498	-0.13460	0.01072
44	27.41098	4.92000	-0.25001	27.82550	4.92000	-0.25001	0.84745	-0.06235	0.00695	-0.12664	0.00484
45	28.33213	4.92000	-0.25001	28.74664	4.92000	-0.25001	0.84745	-0.05941	-0.00762	-0.12402	-0.00126
46	29.25327	4.92000	-0.25001	29.66778	4.92000	-0.25001	0.84745	-0.07283	-0.02197	-0.12791	-0.00782
47	30.17441	4.92000	-0.25001	30.58893	4.92000	-0.25001	0.84745	-0.10103	-0.03357	-0.14045	-0.01880
48	31.09556	4.92000	-0.25001	31.51007	4.92000	-0.25001	0.84745	-0.13854	-0.04034	-0.16351	-0.03000
49	32.01670	4.92000	-0.25001	32.43122	4.92000	-0.25001	0.84745	-0.17911	-0.04620	-0.19805	-0.04407
50	32.93785	4.92000	-0.25001	33.35237	4.92000	-0.25001	0.84746	-0.22164	-0.05180	-0.24687	-0.06404
51	27.17513	5.84000	-0.25001	27.58965	5.84000	-0.25001	0.84747	-0.16337	0.10829	-0.17192	0.04625
52	28.09629	5.84000	-0.25001	28.51081	5.84000	-0.25001	0.84747	-0.10916	0.05022	-0.14712	0.01667
53	29.01745	5.84000	-0.25001	29.43198	5.84000	-0.25001	0.84747	-0.07967	0.02498	-0.13459	0.01072

54	29.93862	5.84000	-0.25001	30.35314	5.84000	-0.25001	0.84747	-0.06235	0.00695	-0.12663	0.00484
55	30.85978	5.84000	-0.25001	31.27431	5.84000	-0.25001	0.84747	-0.05940	-0.00762	-0.12402	-0.00126
56	31.78095	5.84000	-0.25001	32.19548	5.84000	-0.25001	0.84748	-0.07282	-0.02197	-0.12791	-0.00782
57	32.70212	5.84000	-0.25001	33.11665	5.84000	-0.25001	0.84748	-0.10103	-0.03357	-0.14045	-0.01880
58	33.62330	5.84000	-0.25001	34.03783	5.84000	-0.25001	0.84748	-0.13853	-0.04034	-0.16350	-0.03000
59	34.54448	5.84000	-0.25001	34.95901	5.84000	-0.25001	0.84749	-0.17911	-0.04620	-0.19805	-0.04407
60	35.46567	5.84000	-0.25001	35.88020	5.84000	-0.25001	0.84750	-0.22164	-0.05180	-0.24687	-0.06404
61	29.70271	6.76000	-0.25001	30.11723	6.76000	-0.25001	0.84748	-0.16337	0.10829	-0.17191	0.04625
62	30.62388	6.76000	-0.25001	31.03841	6.76000	-0.25001	0.84748	-0.10916	0.05022	-0.14712	0.01667
63	31.54506	6.76000	-0.25001	31.95959	6.76000	-0.25001	0.84748	-0.07967	0.02498	-0.13459	0.01072
64	32.46624	6.76000	-0.25001	32.88077	6.76000	-0.25001	0.84749	-0.06234	0.00695	-0.12663	0.00484
65	33.38742	6.76000	-0.25001	33.80196	6.76000	-0.25001	0.84749	-0.05940	-0.00762	-0.12402	-0.00126
66	34.30862	6.76000	-0.25001	34.72315	6.76000	-0.25001	0.84750	-0.07282	-0.02197	-0.12790	-0.00782
67	35.22982	6.76000	-0.25001	35.64436	6.76000	-0.25001	0.84751	-0.10102	-0.03357	-0.14044	-0.01880
68	36.15102	6.76000	-0.25001	36.56557	6.76000	-0.25001	0.84751	-0.13853	-0.04034	-0.16350	-0.03000
69	37.07224	6.76000	-0.25001	37.48679	6.76000	-0.25001	0.84753	-0.17911	-0.04620	-0.19805	-0.04407
70	37.99348	6.76000	-0.25001	38.40803	6.76000	-0.25001	0.84754	-0.22164	-0.05180	-0.24687	-0.06404
71	32.23028	7.68000	-0.25001	32.64481	7.68000	-0.25001	0.84749	-0.16337	0.10829	-0.17191	0.04625
72	33.15147	7.68000	-0.25001	33.56600	7.68000	-0.25001	0.84750	-0.10916	0.05022	-0.14711	0.01667
73	34.07266	7.68000	-0.25001	34.48720	7.68000	-0.25001	0.84750	-0.07967	0.02498	-0.13459	0.01072
74	34.99386	7.68000	-0.25001	35.40840	7.68000	-0.25001	0.84751	-0.06234	0.00695	-0.12663	0.00484
75	35.91507	7.68000	-0.25001	36.32961	7.68000	-0.25001	0.84751	-0.05939	-0.00762	-0.12401	-0.00126
76	36.83628	7.68000	-0.25001	37.25083	7.68000	-0.25001	0.84752	-0.07281	-0.02197	-0.12790	-0.00782
77	37.75751	7.68000	-0.25001	38.17206	7.68000	-0.25001	0.84753	-0.10102	-0.03357	-0.14044	-0.01880
78	38.67875	7.68000	-0.25001	39.09331	7.68000	-0.25001	0.84755	-0.13852	-0.04034	-0.16350	-0.03000
79	39.60001	7.68000	-0.25001	40.01458	7.68000	-0.25001	0.84757	-0.17910	-0.04620	-0.19805	-0.04407
80	40.52129	7.68000	-0.25001	40.93587	7.68000	-0.25001	0.84759	-0.22163	-0.05180	-0.24687	-0.06404
81	34.13008	8.37150	-0.25001	34.54462	8.37150	-0.25001	0.42651	-0.16337	0.10829	-0.17191	0.04625
82	35.05128	8.37150	-0.25001	35.46582	8.37150	-0.25001	0.42652	-0.10915	0.05022	-0.14711	0.01667
83	35.97248	8.37150	-0.25001	36.38703	8.37150	-0.25001	0.42652	-0.07966	0.02498	-0.13458	0.01072
84	36.89370	8.37150	-0.25001	37.30824	8.37150	-0.25001	0.42652	-0.06233	0.00695	-0.12662	0.00484
85	37.81492	8.37150	-0.25001	38.22947	8.37150	-0.25001	0.42653	-0.05939	-0.00762	-0.12401	-0.00126
86	38.73615	8.37150	-0.25001	39.15071	8.37150	-0.25001	0.42653	-0.07281	-0.02197	-0.12790	-0.00782
87	39.65740	8.37150	-0.25001	40.07196	8.37150	-0.25001	0.42654	-0.10101	-0.03357	-0.14044	-0.01880
88	40.57867	8.37150	-0.25001	40.99324	8.37150	-0.25001	0.42655	-0.13852	-0.04034	-0.16349	-0.03000
89	41.49996	8.37150	-0.25001	41.91454	8.37150	-0.25001	0.42656	-0.17910	-0.04620	-0.19804	-0.04407
90	42.42126	8.37150	-0.25001	42.83585	8.37150	-0.25001	0.42657	-0.22163	-0.05180	-0.24686	-0.06404
91	37.59381	9.63519	-0.25001	38.01644	9.63519	-0.25001	1.89998	-0.18911	0.10829	-0.19733	0.04625
92	38.21616	9.52675	-0.25001	38.63975	9.52675	-0.25001	1.70000	-0.15092	0.05022	-0.18135	0.01667
93	38.83835	9.41834	-0.25001	39.26314	9.41834	-0.25001	1.50001	-0.13054	0.02498	-0.17137	0.01072
94	39.46029	9.30997	-0.25001	39.88666	9.30997	-0.25001	1.30002	-0.12643	0.00695	-0.16722	0.00484
95	40.08185	9.20167	-0.25001	40.51037	9.20167	-0.25001	1.10003	-0.13494	-0.00762	-0.16926	-0.00126
96	40.70277	9.09349	-0.25001	41.13440	9.09349	-0.25001	0.90003	-0.15123	-0.02197	-0.17751	-0.00782
97	41.32249	8.98551	-0.25001	41.75901	8.98551	-0.25001	0.70004	-0.17002	-0.03357	-0.19144	-0.01880
98	41.93959	8.87799	-0.25001	42.38489	8.87799	-0.25001	0.50004	-0.18988	-0.04034	-0.20985	-0.03000
99	42.54880	8.77186	-0.25001	43.01457	8.77186	-0.25001	0.30003	-0.20970	-0.04620	-0.23380	-0.04407
100	43.10263	8.67537	-0.25001	43.37902	8.67537	-0.25001	0.10001	-0.23039	-0.05180	-0.24791	-0.06404

PANEND 14P

TR-805 GIVEN SHAPE

DESCRIPTION OF CASE REQUESTED

SYMMETRICAL CONFIGURATION - PANELS LOCATED ON BOTH SIDES OF X-Z PLANE(SYM = 1.)

CASE = 2. CALCULATE CL, GIVEN SHAPE

CPCALC = 1. NON-LINEAR CP

POLAR = 1. POLARS REQUESTED

THICK = 1. WING THICKNESS PRESSURES TO BE ADDED

VOUT = 1. VELOCITY COMPONENTS TO BE PRINTED

WING REFERENCE AREA = 89.3750

POINT ABOUT WHICH THE MOMENTS ARE TO BE COMPUTED
X-COORDINATE = 0.
Z-COORDINATE = 0.

MACH NUMBER = 1.8000

TR-805 BODY CAMBER 6A

INCLINATION OF BODY AXIS WITH RESPECT TO DEFINING AXIS = 0. DEG.

ANGLE OF ATTACK WITH RESPECT TO BODY AXIS = -0. DEG.

HEIGHT OF WING PLANE ABOVE BODY AXIS = -0.2500

GIVEN WING CAMBER 8A

VELOCITY COMPONENTS ON BODY DUE TO BODY LINE SOURCES AND DOUBLETS

AXIAL(U)

THETA(DEG.) X	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
0.	-0.07462	-0.07237	-0.06830	-0.06293	-0.05723	-0.05289	-0.05065
0.7243	-0.07462	-0.07237	-0.06830	-0.06293	-0.05723	-0.05289	-0.05065
1.4485	-0.05806	-0.06705	-0.06520	-0.06276	-0.06018	-0.05821	-0.05719
2.1723	-0.03481	-0.03405	-0.03266	-0.03082	-0.02887	-0.02739	-0.02662
2.8970	-0.03290	-0.03244	-0.03161	-0.03051	-0.02934	-0.02845	-0.02799
3.6213	-0.02474	-0.02432	-0.02358	-0.02259	-0.02154	-0.02075	-0.02033
4.3456	-0.02436	-0.02400	-0.02336	-0.02251	-0.02160	-0.02092	-0.02056
5.0698	-0.01939	-0.01958	-0.01885	-0.01788	-0.01685	-0.01607	-0.01566
5.7941	-0.01970	-0.01927	-0.01850	-0.01748	-0.01639	-0.01556	-0.01514
6.5184	-0.01594	-0.01558	-0.01493	-0.01408	-0.01317	-0.01248	-0.01212
7.2426	-0.01454	-0.01431	-0.01390	-0.01335	-0.01278	-0.01233	-0.01211

7.9669	-0.01235	-0.01203	-0.01144	-0.01067	-0.00985	-0.00923	-0.00891
8.6911	-0.00986	-0.00981	-0.00973	-0.00962	-0.00951	-0.00942	-0.00938
9.4154	-0.00798	-0.00789	-0.00772	-0.00749	-0.00724	-0.00706	-0.00696
10.1397	-0.00613	-0.00614	-0.00615	-0.00617	-0.00619	-0.00620	-0.00621
10.8639	-0.00371	-0.00380	-0.00398	-0.00422	-0.00446	-0.00465	-0.00475
11.5882	-0.00220	-0.00224	-0.00229	-0.00237	-0.00246	-0.00252	-0.00255
12.3124	0.00013	-0.00000	-0.00023	-0.00054	-0.00087	-0.00111	-0.00124
13.0367	0.00326	0.00305	0.00266	0.00215	0.00161	0.00120	0.00098
13.7610	0.00492	0.00464	0.00412	0.00344	0.00271	0.00216	0.00187
14.5023	0.00929	0.00880	0.00790	0.00672	0.00547	0.00451	0.00402
15.2436	0.01245	0.01194	0.01102	0.00980	0.00851	0.00752	0.00701
15.9849	0.01952	0.01914	0.01846	0.01756	0.01660	0.01588	0.01550
16.7262	0.01593	0.01568	0.01521	0.01459	0.01394	0.01344	0.01318
17.4675	0.01305	0.01289	0.01260	0.01221	0.01180	0.01149	0.01132
18.2088	0.01078	0.01069	0.01053	0.01031	0.01008	0.00991	0.00982
18.9501	0.00898	0.00894	0.00888	0.00879	0.00869	0.00862	0.00858
19.6914	0.00757	0.00756	0.00756	0.00756	0.00755	0.00755	0.00754
20.4327	0.00645	0.00647	0.00650	0.00655	0.00660	0.00664	0.00666
21.1740	0.00555	0.00559	0.00564	0.00572	0.00580	0.00586	0.00590
21.9153	0.00484	0.00488	0.00494	0.00503	0.00513	0.00520	0.00523
22.6566	0.00426	0.00430	0.00437	0.00446	0.00455	0.00462	0.00466
23.3979	0.00379	0.00383	0.00389	0.00397	0.00406	0.00412	0.00416
24.1392	0.00340	0.00343	0.00349	0.00356	0.00363	0.00369	0.00372
24.8805	0.00308	0.00310	0.00315	0.00320	0.00327	0.00331	0.00334
25.6218	0.00280	0.00282	0.00285	0.00290	0.00295	0.00299	0.00301
26.3631	0.00256	0.00257	0.00260	0.00264	0.00267	0.00270	0.00272
27.1044	0.00235	0.00236	0.00238	0.00241	0.00243	0.00246	0.00247
27.8457	0.00216	0.00217	0.00218	0.00220	0.00223	0.00224	0.00225
28.5870	0.00200	0.00200	0.00201	0.00203	0.00204	0.00205	0.00206
29.3283	0.00185	0.00185	0.00186	0.00187	0.00188	0.00189	0.00189
30.0696	0.00295	0.00295	0.00295	0.00296	0.00296	0.00297	0.00297
30.8109	0.00678	0.00678	0.00678	0.00678	0.00679	0.00679	0.00679
31.5522	0.01065	0.01065	0.01065	0.01065	0.01065	0.01065	0.01066
32.2935	0.01303	0.01303	0.01303	0.01303	0.01303	0.01303	0.01303
33.0348	0.01481	0.01481	0.01481	0.01481	0.01482	0.01482	0.01482
33.7761	0.01799	0.01799	0.01799	0.01799	0.01799	0.01799	0.01799
34.5174	0.01888	0.01888	0.01888	0.01888	0.01888	0.01888	0.01888
35.2587	0.01381	0.01381	0.01381	0.01381	0.01381	0.01381	0.01381
36.0000	0.00524	0.00524	0.00524	0.00524	0.00524	0.00524	0.00524

RADIAL(VR)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
X							
0.	0.20701	0.19983	0.18683	0.16969	0.15148	0.13762	0.13044
0.7243	0.20701	0.19983	0.18683	0.16969	0.15148	0.13762	0.13044
1.4485	0.19200	0.18764	0.17973	0.16930	0.15823	0.14979	0.14543
2.1728	0.13935	0.13541	0.12825	0.11882	0.10879	0.10117	0.09722
2.8970	0.13125	0.12802	0.12217	0.11445	0.10625	0.10001	0.09678
3.6213	0.11497	0.11198	0.10656	0.09942	0.09183	0.08605	0.08307
4.3456	0.11102	0.10828	0.10331	0.09675	0.08979	0.08449	0.08174
5.0698	0.10179	0.09905	0.09408	0.08753	0.08056	0.07526	0.07252
5.7941	0.09920	0.09646	0.09149	0.08493	0.07797	0.07268	0.06993
6.5184	0.09157	0.08895	0.08419	0.07792	0.07126	0.06619	0.06356
7.2426	0.08744	0.08505	0.08074	0.07504	0.06900	0.06439	0.06201
7.9669	0.08225	0.07975	0.07521	0.06923	0.06287	0.05803	0.05553
8.6911	0.07640	0.07438	0.07072	0.06590	0.06078	0.05688	0.05486
9.4154	0.07140	0.06939	0.06573	0.06092	0.05580	0.05191	0.04990

10.1397	0.06535	0.06456	0.06138	0.05715	0.05266	0.04924	0.04747
10.8639	0.06027	0.05875	0.05599	0.05236	0.04850	0.04556	0.04404
11.5882	0.05552	0.05400	0.05125	0.04762	0.04377	0.04084	0.03932
12.3124	0.04939	0.04812	0.04582	0.04279	0.03957	0.03712	0.03585
13.0367	0.04171	0.04065	0.03885	0.03641	0.03383	0.03186	0.03084
13.7610	0.03603	0.03526	0.03387	0.03204	0.03009	0.02861	0.02784
14.5023	0.02561	0.02535	0.02488	0.02426	0.02360	0.02310	0.02285
15.2436	0.01636	0.01635	0.01633	0.01631	0.01629	0.01628	0.01627
15.9849	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
16.7262	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
17.4675	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
18.2088	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
18.9501	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
19.6914	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
20.4327	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
21.1740	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
21.9153	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
22.6566	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
23.3979	-0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
24.1392	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
24.8805	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
25.6218	-0.00000	-0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
26.3631	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
27.1044	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
27.8457	-0.00000	0.00000	-0.00000	-0.00000	-0.00000	0.00000	0.00000
28.5870	-0.00000	-0.00000	-0.00000	-0.00000	0.00000	0.00000	0.00000
29.3283	-0.00000	-0.00000	-0.00000	-0.00000	-0.00000	-0.00000	-0.00000
30.0696	-0.00209	-0.00209	-0.00209	-0.00209	-0.00209	-0.00209	-0.00209
30.8109	-0.00915	-0.00915	-0.00915	-0.00915	-0.00915	-0.00915	-0.00915
31.5522	-0.01740	-0.01740	-0.01740	-0.01740	-0.01740	-0.01740	-0.01740
32.2935	-0.02427	-0.02427	-0.02427	-0.02427	-0.02427	-0.02427	-0.02427
33.0348	-0.03092	-0.03092	-0.03092	-0.03092	-0.03092	-0.03092	-0.03092
33.7761	-0.04072	-0.04072	-0.04072	-0.04072	-0.04072	-0.04072	-0.04072
34.5174	-0.04786	-0.04786	-0.04786	-0.04786	-0.04786	-0.04786	-0.04786
35.2587	-0.04502	-0.04502	-0.04502	-0.04502	-0.04502	-0.04502	-0.04502
36.0000	-0.03418	-0.03418	-0.03418	-0.03418	-0.03418	-0.03418	-0.03418

TANGENTIAL(VT)

THETA(DEG.) X	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
0.	0.00628	0.01766	0.02572	0.02899	0.02605	0.01766	0.00628
0.7243	0.00628	0.01766	0.02572	0.02899	0.02605	0.01766	0.00628
1.4485	0.00448	0.01261	0.01837	0.02070	0.01860	0.01261	0.00448
2.1728	0.00425	0.01194	0.01740	0.01961	0.01762	0.01194	0.00425
2.8970	0.00374	0.01051	0.01531	0.01725	0.01550	0.01051	0.00374
3.6213	0.00342	0.00961	0.01400	0.01578	0.01417	0.00961	0.00342
4.3456	0.00314	0.00884	0.01289	0.01452	0.01305	0.00884	0.00314
5.0698	0.00303	0.00852	0.01241	0.01399	0.01256	0.00852	0.00303
5.7941	0.00298	0.00839	0.01222	0.01378	0.01237	0.00839	0.00298
6.5184	0.00295	0.00830	0.01209	0.01363	0.01225	0.00830	0.00295
7.2426	0.00285	0.00801	0.01167	0.01315	0.01181	0.00801	0.00285
7.9669	0.00281	0.00789	0.01150	0.01296	0.01164	0.00789	0.00281
8.6911	0.00265	0.00746	0.01087	0.01225	0.01101	0.00746	0.00265
9.4154	0.00250	0.00704	0.01026	0.01156	0.01039	0.00704	0.00250
10.1397	0.00234	0.00657	0.00957	0.01079	0.00969	0.00657	0.00234
10.8639	0.00213	0.00600	0.00874	0.00985	0.00885	0.00600	0.00213
11.5882	0.00197	0.00554	0.00807	0.00909	0.00817	0.00554	0.00197

12.3124	C.00179	0.00505	0.00735	0.00829	0.00745	0.00505	0.00179
13.0367	C.00159	0.00447	0.00651	0.00733	0.00559	0.00447	0.00159
13.7610	C.00135	0.00379	0.00553	0.00623	0.00560	0.00379	0.00135
14.5023	C.00101	0.00285	0.00415	0.00467	0.00420	0.00285	0.00101
15.2436	C.00064	0.00180	0.00262	0.00296	0.00266	0.00180	0.00064
15.9849	C.00033	0.00094	0.00137	0.00154	0.00138	0.00094	0.00033
16.7262	C.00011	0.00030	0.00044	0.00050	0.00045	0.00030	0.00011
17.4675	-C.00004	-0.00012	-C.00018	-0.00020	-0.00018	-0.00012	-0.00004
18.2088	-C.00014	-0.00038	-0.00056	-0.00063	-0.00057	-0.00038	-0.00014
18.9501	-C.00018	-0.00052	-0.00076	-0.00085	-0.00077	-0.00052	-0.00018
19.6914	-C.00020	-0.00057	-0.00082	-0.00093	-0.00084	-0.00057	-0.00020
20.4327	-C.00020	-0.00055	-0.00081	-0.00091	-0.00082	-0.00055	-0.00020
21.1740	-C.00018	-0.00051	-0.00074	-0.00083	-0.00075	-0.00051	-0.00018
21.9153	-C.00016	-0.00044	-0.00064	-0.00072	-0.00065	-0.00044	-0.00016
22.6566	-C.00013	-0.00037	-0.00054	-0.00061	-0.00054	-0.00037	-0.00013
23.3979	-C.00011	-0.0003C	-0.00044	-0.00049	-0.00044	-0.00030	-0.00011
24.1392	-C.00008	-0.00024	-0.00034	-0.00039	-0.00035	-0.00024	-0.00008
24.8805	-C.00006	-0.00018	-0.00026	-0.00030	-0.00027	-0.00018	-0.00006
25.6218	-C.00005	-0.00014	-0.00020	-0.00022	-0.00020	-0.00014	-0.00005
26.3631	-C.00004	-0.0001C	-0.00015	-0.00016	-0.00015	-0.00010	-0.00004
27.1044	-C.00003	-0.00007	-0.00011	-0.00012	-0.00011	-0.00007	-0.00003
27.8457	-C.00002	-0.00005	-0.00008	-0.00009	-0.00008	-0.00005	-0.00002
28.5870	-C.00001	-0.00004	-0.00005	-0.00006	-0.00005	-0.00004	-0.00001
29.3283	-C.00001	-0.00003	-0.00004	-0.00004	-0.00004	-0.00003	-0.00001
30.0696	-C.00001	-0.00002	-0.00003	-0.00003	-0.00003	-0.00002	-0.00001
30.8109	-C.00001	-0.00002	-0.00002	-0.00003	-0.00002	-0.00002	-0.00001
31.5522	-C.00000	-0.00001	-0.00002	-0.00002	-0.00002	-0.00001	-0.00000
32.2935	-C.00000	-0.00001	-0.00002	-0.00002	-0.00002	-0.00001	-0.00000
33.0348	-C.00000	-0.00001	-0.00002	-0.00002	-0.00002	-0.00001	-0.00000
33.7761	-C.00000	-0.00001	-0.00002	-0.00002	-0.00002	-0.00001	-0.00000
34.5174	-C.00000	-0.00001	-0.00002	-0.00002	-0.00002	-0.00001	-0.00000
35.2587	-C.00000	-0.00001	-0.00002	-0.00002	-0.00002	-0.00001	-0.00000
36.0000	-C.00000	-0.00001	-0.00002	-0.00002	-0.00002	-0.00001	-0.00000

VELOCITY COMPONENTS ON WING PANELS DUE TO BODY PANEL PRESSURE SINGULARITIES

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-C.00773	-0.00237	-0.00081	0.00479	0.00291	0.00237	0.00069	-0.00020	-0.00031	-0.00236
2	-0.00420	-0.00084	0.00223	0.00202	0.00342	0.00092	0.00010	-0.00113	-0.00061	-0.00144
3	-C.00157	0.00189	0.00391	0.00098	0.00116	0.00027	-0.00024	-0.00089	-0.00197	-0.00082
4	C.00184	0.00479	0.00354	-0.00053	0.00033	-0.00002	-0.00062	-0.00205	-0.00157	-0.00089
5	C.00635	0.00475	0.00269	-0.00042	-0.00151	-0.00057	-0.00272	-0.00048	-0.00100	0.00506
6	0.00571	0.00515	0.00175	-0.00013	-0.00060	-0.00326	-0.00179	-0.00108	0.00019	0.00031
7	C.00805	0.00257	0.00071	-0.00156	-0.00080	-0.00190	-0.00104	0.00032	0.00031	0.00031
8	C.00456	0.00131	0.00011	-0.00128	-0.00229	-0.00086	-0.00074	0.00030	0.00031	0.00012
9	G.00311	0.00013	-0.00060	-0.00310	-0.00126	-0.00109	0.00038	0.00034	0.00009	0.00009
10	C.00108	-0.00056	-0.00401	-0.00082	-0.00133	0.00053	0.00035	0.00009	0.00008	0.00010

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	C.01192	0.00588	0.00266	-0.00647	-0.00427	-0.00379	-0.00137	-0.00001	0.00021	0.00342

2	0.01117	0.00438	-0.00192	-0.00273	-0.00532	-0.00176	-0.00053	0.00139	0.00068	0.00211
3	0.01045	0.00065	-0.00503	-0.00159	-0.00219	-0.00088	-0.00003	0.00106	0.00284	0.00124
4	0.00650	-0.00461	-0.00509	0.00024	-0.00110	-0.00047	0.00057	0.00294	0.00234	0.00139
5	-0.00144	-0.00565	-0.00462	-0.00020	0.00157	0.00038	0.00385	0.00070	0.00154	-0.00750
6	-0.00329	-0.00783	-0.00376	-0.00080	0.00023	0.00450	0.00262	0.00166	-0.00018	-0.00040
7	-0.00984	-0.00551	-0.00261	0.00131	0.00057	0.00270	0.00159	-0.00037	-0.00040	-0.00042
8	-0.00881	-0.00467	-0.00190	0.00093	0.00312	0.00127	0.00121	-0.00038	-0.00041	-0.00014
9	-0.01024	-0.00394	-0.00088	0.00399	0.00179	0.00174	-0.00049	-0.00046	-0.00010	-0.00009
10	-0.01028	-0.00322	0.00458	0.00095	0.00204	-0.00070	-0.00045	-0.00009	-0.00009	-0.00011

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.00993	-0.00174	-0.00065	0.00009	0.00029	0.00023	0.00021	0.00022	0.00023	0.00020
2	-0.01448	-0.00201	-0.00054	0.00018	0.00040	0.00031	0.00026	0.00024	0.00029	0.00035
3	-0.01496	-0.00156	0.00010	0.00034	0.00035	0.00037	0.00031	0.00034	0.00032	0.00048
4	-0.01342	-0.00069	0.00084	0.00026	0.00045	0.00042	0.00039	0.00039	0.00051	0.00052
5	-0.00889	0.00081	0.00097	0.00059	0.00044	0.00053	0.00039	0.00065	0.00059	0.00085
6	-0.00199	0.00203	0.00143	0.00093	0.00072	0.00049	0.00069	0.00068	0.00075	0.00072
7	0.00792	0.00345	0.00183	0.00098	0.00094	0.00081	0.00084	0.00087	0.00082	0.00077
8	0.01790	0.00489	0.00210	0.00131	0.00102	0.00110	0.00092	0.00095	0.00088	0.00078
9	0.02978	0.00570	0.00246	0.00134	0.00141	0.00118	0.00115	0.00102	0.00086	0.00081
10	0.03859	0.00702	0.00244	0.00205	0.00153	0.00144	0.00123	0.00100	0.00088	0.00084

VELOCITY COMPONENTS ON WING PANELS DUE TO WING PANEL PRESSURE SINGULARITIES

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.01798	-0.01846	-0.01782	-0.01609	-0.01371	-0.01092	-0.00798	-0.00528	-0.00382	-0.00049
2	-0.00029	0.00103	0.00253	0.00436	0.00641	0.00843	0.01017	0.01150	0.01263	0.01365
3	0.00108	0.00331	0.00548	0.00750	0.00947	0.01113	0.01223	0.01298	0.01396	0.01400
4	0.00339	0.00612	0.00878	0.01088	0.01271	0.01383	0.01445	0.01496	0.01560	0.01522
5	0.00598	0.00906	0.01177	0.01395	0.01520	0.01578	0.01617	0.01667	0.01684	0.01637
6	0.00883	0.01203	0.01475	0.01641	0.01702	0.01727	0.01762	0.01795	0.01786	0.01724
7	0.01250	0.01540	0.01756	0.01831	0.01857	0.01878	0.01893	0.01917	0.01887	0.01831
8	0.01479	0.01723	0.01816	0.01830	0.01841	0.01856	0.01859	0.01868	0.01834	0.01787
9	0.01671	0.01712	0.01688	0.01689	0.01700	0.01697	0.01707	0.01700	0.01667	0.01654
10	0.01487	0.01312	0.01269	0.01287	0.01275	0.01277	0.01282	0.01269	0.01250	0.01487

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.04940	0.05072	0.04896	0.04421	0.03767	0.03001	0.02194	0.01450	0.01049	0.00133
2	0.00079	-0.00284	-0.00696	-0.01199	-0.01760	-0.02315	-0.02795	-0.03160	-0.03471	-0.03751
3	-0.00297	-0.00909	-0.01507	-0.02060	-0.02601	-0.03057	-0.03359	-0.03566	-0.03836	-0.03846
4	-0.00931	-0.01681	-0.02413	-0.02989	-0.03493	-0.03801	-0.03970	-0.04111	-0.04287	-0.04182
5	-0.01643	-0.02490	-0.03233	-0.03834	-0.04176	-0.04335	-0.04443	-0.04580	-0.04626	-0.04497
6	-0.02427	-0.03306	-0.04053	-0.04510	-0.04675	-0.04744	-0.04840	-0.04933	-0.04906	-0.04736
7	-0.03435	-0.04232	-0.04825	-0.05030	-0.05103	-0.05161	-0.05200	-0.05267	-0.05185	-0.05030
8	-0.04063	-0.04734	-0.04991	-0.05029	-0.05057	-0.05098	-0.05108	-0.05133	-0.05039	-0.04909
9	-0.04591	-0.04704	-0.04637	-0.04639	-0.04671	-0.04664	-0.04689	-0.04671	-0.04580	-0.04543
10	-0.04086	-0.03604	-0.03466	-0.03536	-0.03503	-0.03509	-0.03522	-0.03487	-0.03433	-0.04085

VERTICAL(w)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.06060	0.04762	0.04628	0.04566	0.04561	0.04579	0.04589	0.04593	0.04595	0.04600
2	0.03167	0.01806	0.01657	0.01604	0.01598	0.01617	0.01628	0.01635	0.01632	0.01628
3	0.02553	0.01152	0.01004	0.01000	0.01013	0.01020	0.01031	0.01031	0.01035	0.01020
4	0.01764	0.00477	0.00350	0.00427	0.00421	0.00430	0.00437	0.00440	0.00430	0.00429
5	0.00680	-0.00277	-0.00264	-0.00209	-0.00185	-0.00189	-0.00171	-0.00195	-0.00188	-0.00214
6	-0.00670	-0.01045	-0.00959	-0.00894	-0.00866	-0.00839	-0.00856	-0.00854	-0.00860	-0.00857
7	-0.02748	-0.02273	-0.02089	-0.01993	-0.01982	-0.01966	-0.01968	-0.01970	-0.01965	-0.01961
8	-0.04854	-0.03527	-0.03230	-0.03143	-0.03109	-0.03115	-0.03095	-0.03098	-0.03095	-0.03090
9	-0.07438	-0.05006	-0.04669	-0.04550	-0.04554	-0.04529	-0.04526	-0.04518	-0.04509	-0.04506
10	-0.10304	-0.07128	-0.06660	-0.06617	-0.06562	-0.06552	-0.06533	-0.06522	-0.06515	-0.06509

VELOCITY COMPONENTS ON WING PANELS DUE TO BODY LINE SOURCES AND DOUBLETS

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.00778	0.01212	0.00792	0.00562	0.00416	0.00317	0.00250	0.00200	0.00172	0.00134
2	0.01504	0.00980	0.00676	0.00478	0.00361	0.00277	0.00221	0.00179	0.00155	0.00124
3	0.01185	0.00825	0.00568	0.00412	0.00312	0.00244	0.00196	0.00161	0.00140	0.00116
4	0.00988	0.00699	0.00483	0.00358	0.00273	0.00217	0.00176	0.00145	0.00127	0.00108
5	0.00830	0.00597	0.00417	0.00312	0.00242	0.00193	0.00158	0.00132	0.00116	0.00101
6	0.00703	0.00503	0.00363	0.00274	0.00216	0.00174	0.00144	0.00121	0.00107	0.00095
7	0.00588	0.00434	0.00318	0.00243	0.00193	0.00157	0.00131	0.00111	0.00098	0.00146
8	0.00503	0.00378	0.00279	0.00217	0.00174	0.00143	0.00120	0.00113	0.00205	0.00304
9	0.00436	0.00332	0.00248	0.00195	0.00158	0.00131	0.00110	0.00225	0.00377	0.00443
10	0.00380	0.00291	0.00222	0.00176	0.00144	0.00120	0.00154	0.00403	0.00509	0.00510

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.01302	-0.00416	-0.00400	-0.00342	-0.00278	-0.00222	-0.00182	-0.00148	-0.00129	-0.00102
2	-0.00154	-0.00296	-0.00337	-0.00276	-0.00232	-0.00186	-0.00154	-0.00128	-0.00112	-0.00091
3	-0.00063	-0.00249	-0.00266	-0.00228	-0.00189	-0.00158	-0.00131	-0.00111	-0.00098	-0.00082
4	-0.00066	-0.00208	-0.00212	-0.00190	-0.00159	-0.00135	-0.00113	-0.00096	-0.00086	-0.00074
5	-0.00064	-0.00175	-0.00174	-0.00157	-0.00135	-0.00115	-0.00098	-0.00084	-0.00076	-0.00067
6	-0.00060	-0.00131	-0.00146	-0.00131	-0.00115	-0.00099	-0.00086	-0.00075	-0.00068	-0.00061
7	-0.00036	-0.00108	-0.00121	-0.00111	-0.00098	-0.00086	-0.00075	-0.00066	-0.00060	-0.00143
8	-0.00030	-0.00090	-0.00100	-0.00095	-0.00085	-0.00076	-0.00067	-0.00076	-0.00227	-0.00387
9	-0.00027	-0.00077	-0.00085	-0.00081	-0.00074	-0.00066	-0.00059	-0.00252	-0.00499	-0.00612
10	-0.00025	-0.00062	-0.00073	-0.00070	-0.00065	-0.00058	-0.00133	-0.00535	-0.00720	-0.00725

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.00442	0.00036	0.00062	0.00050	0.00034	0.00023	0.00015	0.00010	0.00007	0.00004
2	-0.00053	0.00061	0.00064	0.00044	0.00029	0.00019	0.00012	0.00008	0.00006	0.00004
3	0.00015	0.00075	0.00058	0.00037	0.00024	0.00015	0.00010	0.00006	0.00005	0.00003
4	0.00062	0.00076	0.00050	0.00031	0.00019	0.00012	0.00008	0.00005	0.00004	0.00003
5	0.00083	0.00070	0.00042	0.00025	0.00015	0.00009	0.00006	0.00004	0.00003	0.00003

6	0.00087	0.00059	0.00034	0.00019	0.00012	0.00008	0.00005	0.00004	0.00003	0.00002
7	0.00077	0.00048	0.00026	0.00015	0.00009	0.00006	0.00004	0.00003	0.00003	0.00005
8	0.00065	0.00036	0.00020	0.00012	0.00007	0.00005	0.00004	0.00003	0.00007	0.00012
9	0.00053	0.00030	0.00016	0.00009	0.00006	0.00004	0.00003	0.00009	0.00016	0.00018
10	0.00041	0.00022	0.00012	0.00007	0.00005	0.00004	0.00006	0.00018	0.00022	0.00021

VELOCITY COMPONENTS ON WING PANELS DUE TO WING SOURCES

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.03054	-0.02433	-0.01880	-0.01746	-0.01326	-0.00857	-0.00441	-0.00076	-0.00148	-0.00331
2	-0.00716	-0.00013	0.00337	0.00784	0.01289	0.01731	0.02137	0.02116	0.02006	0.01976
3	-0.00189	0.00375	0.00866	0.01475	0.01964	0.02390	0.02566	0.02337	0.02313	0.02314
4	-0.00060	0.00495	0.01303	0.01903	0.02368	0.02789	0.02409	0.02362	0.02363	0.02249
5	-0.00045	0.00850	0.01654	0.02209	0.02667	0.02357	0.02284	0.02278	0.02177	0.02115
6	0.00270	0.01161	0.01874	0.02411	0.02187	0.02069	0.02053	0.01941	0.01883	0.01855
7	0.00347	0.01297	0.01968	0.01900	0.01691	0.01658	0.01565	0.01475	0.01439	0.01431
8	0.00716	0.01399	0.01652	0.01237	0.01172	0.01114	0.00986	0.00933	0.00908	0.00913
9	0.00900	0.01548	0.00706	0.00593	0.00571	0.00393	0.00330	0.00293	0.00274	0.00307
10	0.01097	-0.00233	-0.00442	-0.00488	-0.00666	-0.00745	-0.00788	-0.00816	-0.00831	-0.00557

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.02346	0.03708	0.03491	0.03503	0.02960	0.02265	0.01605	0.01001	0.01062	0.01346
2	-0.02849	-0.02351	-0.02451	-0.02979	-0.03719	-0.04422	-0.05097	-0.05142	-0.05010	-0.04997
3	-0.03063	-0.02966	-0.03436	-0.04316	-0.05102	-0.05823	-0.06183	-0.05896	-0.05886	-0.05917
4	-0.02439	-0.02819	-0.03974	-0.04951	-0.05761	-0.06514	-0.06024	-0.05998	-0.06022	-0.05877
5	-0.01502	-0.02915	-0.04278	-0.05283	-0.06136	-0.05790	-0.05738	-0.05766	-0.05636	-0.05561
6	-0.00956	-0.02868	-0.04278	-0.05337	-0.05185	-0.05088	-0.05112	-0.04977	-0.04905	-0.04882
7	-0.00345	-0.02458	-0.03956	-0.04162	-0.03967	-0.03983	-0.03886	-0.03773	-0.03731	-0.03742
8	0.00316	-0.02012	-0.03011	-0.02594	-0.02587	-0.02555	-0.02394	-0.02333	-0.02304	-0.02340
9	0.00822	-0.01726	-0.00930	-0.00904	-0.00947	-0.00721	-0.00649	-0.00607	-0.00586	-0.00692
10	0.01231	0.02030	0.02082	0.02039	0.02246	0.02335	0.02383	0.02413	0.02430	0.01669

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829
2	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022
3	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498
4	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695
5	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762
6	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197
7	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357
8	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034
9	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620
10	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05130

VELOCITY COMPONENTS ON BODY PANELS DUE TO BODY PANEL PRESSURE SINGULARITIES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.00001	-0.00003	-0.00012	-0.00137	0.00127	0.00011	0.00003
2	0.00001	-0.00025	-0.00010	-0.01309	-0.00147	0.00104	0.00009
3	0.00038	0.00216	-0.00568	-0.00677	-0.00455	0.00035	0.00195
4	0.00765	-0.00236	-0.00417	-0.00250	-0.00223	0.00027	0.00566
5	0.00198	0.00311	0.00197	0.00661	0.00422	0.00159	0.00165
6	-0.00138	0.00066	0.00382	0.00902	0.00234	0.00158	-0.00072
7	-0.00285	-0.00034	0.00383	0.00995	0.00166	-0.00018	-0.00167
8	-0.00268	-0.00066	0.00349	0.01053	-0.00089	-0.00098	-0.00194
9	-0.00238	-0.00064	0.00370	0.01031	-0.00312	-0.00221	-0.00227
10	-0.00243	-0.00014	0.00418	0.01003	-0.00472	-0.00365	-0.00351
11	0.00034	0.00185	0.00306	0.00279	-0.00517	-0.00493	-0.00567
12	0.00352	0.00313	0.00073	-0.00406	-0.00538	-0.00441	-0.00457
13	0.00414	0.00320	0.00179	0.00005	-0.00138	-0.00236	-0.00357
14	0.00219	0.00192	0.00122	0.00024	-0.00060	-0.00133	-0.00174

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.00001	0.00004	0.00011	0.00344	-0.00269	-0.00018	-0.00004
2	0.00021	0.00034	-0.00158	0.03276	0.00021	-0.00041	0.00006
3	-0.00012	-0.00092	0.00259	0.04053	0.00101	0.00118	0.00071
4	-0.00119	-0.00132	0.00622	0.03840	0.00142	-0.00367	-0.00295
5	-0.00334	-0.00735	0.00098	0.02314	-0.00776	-0.00669	-0.00168
6	-0.00272	-0.00599	0.00101	0.01178	-0.00574	-0.00526	-0.00291
7	-0.00182	-0.00276	0.00190	-0.00291	-0.00112	-0.00405	-0.00149
8	-0.00011	-0.00054	0.00078	-0.02075	0.00620	-0.00086	-0.00053
9	0.00003	0.00032	-0.00136	-0.04040	0.01544	0.00086	-0.00029
10	0.00111	0.00179	-0.00394	-0.06064	0.02627	0.00373	0.00097
11	0.00255	0.00371	-0.01104	-0.07700	0.04091	0.00958	0.00172
12	0.00235	0.00083	-0.01468	-0.06735	0.05292	0.01486	0.00351
13	-0.00141	-0.00631	-0.02021	-0.06751	0.05418	0.01367	0.00319
14	-0.00256	-0.00955	-0.02300	-0.06508	0.05169	0.01600	0.00434

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.00000	-0.00003	-0.00022	0.00040	0.00169	-0.00014	-0.00001
2	-0.00005	-0.00026	-0.00400	0.00043	0.00372	0.00253	0.00001
3	-0.00094	-0.00746	-0.00183	0.00145	0.00218	0.00602	0.00384
4	-0.01483	-0.00523	0.00123	0.00129	0.00316	0.00631	0.01217
5	-0.01396	-0.00833	-0.00005	0.00177	0.00333	0.00632	0.01179
6	-0.01097	-0.00472	0.00234	-0.00103	0.00154	0.00576	0.00983
7	-0.00714	-0.00303	0.00130	0.00040	-0.00045	0.00382	0.00784
8	-0.00324	-0.00056	0.00206	-0.00096	-0.00407	0.00244	0.00513
9	0.00127	0.00251	0.00194	-0.00227	-0.01051	-0.00117	0.00167
10	0.00543	0.00438	0.00078	-0.00460	-0.01839	-0.00487	-0.00200
11	0.00987	0.00450	-0.00398	-0.00804	-0.02888	-0.01362	-0.00999
12	0.00720	-0.00032	-0.00781	-0.00869	-0.03333	-0.01776	-0.01427
13	-0.00250	-0.00568	-0.00947	-0.00900	-0.03556	-0.01925	-0.01581

14	-C.00756	-0.00908	-0.01050	-0.01662	-0.04239	-0.02145	-0.01858

VELOCITY COMPONENTS ON BODY PANELS DUE TO WING PANEL PRESSURE SINGULARITIES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	C.	0.	0.	-0.00441	0.00305	0.	0.
2	C.	0.	-0.00287	-0.00213	0.00148	0.00193	0.
3	-C.00093	-0.00225	-0.00146	-0.00055	0.00042	0.00096	0.00195
4	-C.00206	-0.00146	-0.00094	0.00052	-0.00025	0.00093	0.00112
5	-C.00105	-0.00094	-0.00015	0.00169	-0.00111	0.00015	0.00059
6	-C.00048	-0.00011	0.00094	0.00302	-0.00206	-0.00057	0.00004
7	C.00030	0.00074	0.00203	0.00457	-0.00316	-0.00136	-0.00065
8	C.00119	0.00174	0.00336	0.00591	-0.00413	-0.00231	-0.00141
9	C.00221	0.00293	0.00459	0.00704	-0.00496	-0.00320	-0.00231
10	C.00334	0.00407	0.00571	0.00715	-0.00509	-0.00405	-0.00318
11	C.00521	0.00567	0.00407	0.00175	-0.00147	-0.00313	-0.00432
12	C.00314	0.00286	0.00228	0.00089	-0.00077	-0.00171	-0.00200
13	C.00192	0.00175	0.00127	0.00048	-0.00042	-0.00103	-0.00125
14	C.00127	0.00111	0.00079	0.00030	-0.00025	-0.00063	-0.00081

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	C.	0.	0.	-0.00362	0.00552	0.	0.
2	C.	0.	-0.00187	-0.01446	0.01415	0.00273	0.
3	-C.00088	-0.00225	-0.00391	-0.01724	0.01562	0.00398	0.00164
4	-C.00027	-0.00174	-0.00452	-0.01662	0.01450	0.00353	0.00087
5	-C.00047	-0.00178	-0.00414	-0.01318	0.01105	0.00331	0.00097
6	-C.00042	-0.00154	-0.00320	-0.00703	0.00515	0.00221	0.00068
7	-C.00022	-0.00075	-0.00127	0.00231	-0.00362	0.00039	0.00025
8	C.00009	0.00028	0.00166	0.01500	-0.01521	-0.00225	-0.00036
9	C.00055	0.00188	0.00556	0.03025	-0.02893	-0.00560	-0.00126
10	C.00110	0.00384	0.01022	0.04685	-0.04332	-0.00945	-0.00227
11	C.00232	0.00823	0.01846	0.06321	-0.05476	-0.01475	-0.00437
12	C.00262	0.00911	0.02089	0.06492	-0.05594	-0.01593	-0.00423
13	C.00306	0.01030	0.02231	0.06560	-0.05646	-0.01668	-0.00457
14	C.00326	0.01078	0.02279	0.06388	-0.05161	-0.01656	-0.00463

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-C.	-0.	-0.	0.00882	0.00404	-0.	-0.
2	-C.	-0.	0.00480	0.00945	0.00060	0.00221	-0.
3	C.00116	0.00335	0.00348	0.00564	-0.00259	0.00051	0.00144
4	0.00208	0.00185	0.00239	0.00218	-0.00474	-0.00058	-0.00038
5	C.00057	0.00075	0.00049	-0.00184	-0.00645	-0.00205	-0.00143
6	-C.00034	-0.00057	-0.00162	-0.00602	-0.00707	-0.00306	-0.00222
7	-C.00134	-0.00186	-0.00380	-0.01075	-0.00680	-0.00372	-0.00275

8	-0.00229	-0.00317	-0.00625	-0.01526	-0.00506	-0.00402	-0.00289
9	-0.00317	-0.00455	-0.00842	-0.01913	-0.00184	-0.00364	-0.00264
10	-0.00390	-0.00559	-0.01020	-0.02068	0.00351	-0.00255	-0.00176
11	-0.00393	-0.00578	-0.00634	-0.00597	0.01998	0.00442	0.00227
12	0.00136	0.00100	0.00040	0.00270	0.02816	0.01114	0.00917
13	0.00539	0.00526	0.00538	0.00834	0.03354	0.01582	0.01341
14	0.00903	0.00927	0.01036	0.02071	0.04421	0.02061	0.01749

VELOCITY COMPONENTS ON BODY PANELS DUE TO WING SOURCES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.	0.	0.	-0.00275	-0.00157	0.	0.
2	0.	0.	-0.00645	-0.01262	-0.01134	-0.00520	0.
3	0.	-0.00686	-0.00902	-0.00734	-0.00732	-0.00785	-0.00585
4	-0.01113	-0.00699	-0.00646	-0.00369	-0.00407	-0.00956	-0.01252
5	-0.01124	-0.01126	-0.00895	-0.00581	-0.00707	-0.00877	-0.01019
6	-0.00891	-0.00826	-0.00613	-0.00292	-0.00375	-0.00602	-0.00737
7	-0.00613	-0.00531	-0.00300	0.00021	-0.00054	-0.00297	-0.00441
8	-0.00328	-0.00224	0.00013	0.00300	0.00234	0.00008	-0.00138
9	-0.00025	0.00076	0.00290	0.00529	0.00473	0.00290	0.00167
10	0.00267	0.00351	0.00541	0.00753	0.00705	0.00542	0.00439
11	0.00632	0.00717	0.00881	0.00308	0.00388	0.00862	0.00773
12	0.00740	0.00597	0.00467	0.00365	0.00410	0.00548	0.00453
13	0.00208	0.00191	0.00159	0.00128	0.00131	0.00154	0.00169
14	0.00092	0.00084	0.00070	0.00059	0.00061	0.00071	0.00078

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.	0.	0.	-0.00015	-0.00106	0.	0.
2	0.	0.	-0.00215	-0.01927	-0.02232	-0.00479	0.
3	0.	-0.00460	-0.00837	-0.02398	-0.02413	-0.01141	-0.00699
4	-0.00260	-0.00750	-0.01101	-0.02218	-0.02128	-0.00715	-0.00056
5	-0.00052	0.00001	-0.00376	-0.01012	-0.00750	-0.00338	-0.00047
6	-0.00028	-0.00024	-0.00255	-0.00459	-0.00255	-0.00168	-0.00015
7	-0.00016	-0.00001	-0.00119	0.00100	0.00295	-0.00000	0.00020
8	-0.00004	0.00046	0.00043	0.00642	0.00817	0.00181	0.00071
9	0.00015	0.00109	0.00215	0.01104	0.01251	0.00342	0.00112
10	0.00032	0.00154	0.00361	0.01487	0.01607	0.00468	0.00134
11	0.00066	0.00205	0.00560	0.01433	0.01246	0.00665	0.00165
12	-0.00173	-0.00105	0.00067	0.00264	0.00152	-0.00122	-0.00010
13	0.00013	0.00054	0.00127	0.00194	0.00169	0.00090	0.00026
14	0.00015	0.00048	0.00086	0.00112	0.00099	0.00059	0.00019

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.	-0.	-0.	0.00415	-0.00212	-0.	-0.

2	-C.	-0.	0.00997	0.03008	-0.02056	-0.00664	-0.
3	-C.	0.01007	0.01697	0.02118	-0.01490	-0.01133	-0.00630
4	C.01365	0.01149	0.01425	0.01298	-0.00936	-0.01133	-0.01238
5	C.01435	0.01457	0.01285	0.00678	-0.00548	-0.00945	-0.01063
6	C.01206	0.01125	0.00839	0.00032	-0.00087	-0.00633	-0.00813
7	C.00897	0.00761	0.00357	-0.00594	0.00360	-0.00291	-0.00532
8	C.00555	0.00359	-0.00133	-0.01125	0.00744	0.00059	-0.00228
9	C.00174	-0.00049	-0.00573	-0.01520	0.01034	0.00380	0.00088
10	-C.00209	-0.00429	-0.00956	-0.01861	0.01287	0.00662	0.00377
11	-0.00716	-0.00946	-0.01471	-0.00827	0.00607	0.01034	0.00750
12	-C.00928	-0.00750	-0.00580	-0.00246	0.00212	0.00487	0.00492
13	-C.00328	-0.00306	-0.00239	-0.00094	0.00081	0.00182	0.00215
14	-C.00165	-0.00150	-0.00111	-0.00043	0.00035	0.00086	0.00106

PRESSURES, FORCES, AND MOMENTS ON ISOLATED BODY

CD = 0.00073 CL = 0.00032 CM = -0.00557

BODY PRESSURE COEFFICIENTS(CP)

THETA(DEG.) X	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
0.	C.11882	0.11623	0.11149	0.10511	0.09818	0.09280	0.08998
0.7243	C.11882	0.11623	0.11149	0.10511	0.09818	0.09280	0.08998
1.4485	C.10962	0.10879	0.10728	0.10526	0.10308	0.10141	0.10053
2.1728	C.05290	0.05221	0.05095	0.04927	0.04747	0.04609	0.04537
2.8970	C.05099	0.05074	0.05030	0.04971	0.04908	0.04861	0.04836
3.6213	C.03761	0.03734	0.03685	0.03619	0.03549	0.03496	0.03468
4.3456	C.03771	0.03749	0.03710	0.03658	0.03602	0.03560	0.03537
5.0698	C.03050	0.03014	0.02949	0.02862	0.02769	0.02698	0.02661
5.7941	C.03042	0.03000	0.02924	0.02823	0.02715	0.02632	0.02588
6.5184	C.02406	0.02373	0.02313	0.02235	0.02151	0.02086	0.02053
7.2426	C.02190	0.02178	0.02158	0.02130	0.02102	0.02080	0.02069
7.9669	C.01827	0.01795	0.01739	0.01664	0.01584	0.01522	0.01490
8.6911	C.01408	0.01425	0.01455	0.01496	0.01540	0.01575	0.01593
9.4154	C.01101	0.01105	0.01114	0.01125	0.01138	0.01149	0.01154
10.1397	C.00794	0.00815	0.00853	0.00904	0.00960	0.01003	0.01025
10.8639	C.00381	0.00415	0.00479	0.00563	0.00654	0.00724	0.00761
11.5882	C.00133	0.00153	0.00191	0.00241	0.00294	0.00336	0.00357
12.3124	-C.00270	-0.00234	-0.00169	-0.00082	0.00011	0.00083	0.00120
13.0367	-C.00825	-0.00776	-0.00686	-0.00567	-0.00440	-0.00342	-0.00292
13.7610	-C.01109	-0.01048	-0.00938	-0.00791	-0.00634	-0.00514	-0.00452
14.5023	-C.01905	-0.01807	-0.01630	-0.01395	-0.01144	-0.00952	-0.00852
15.2436	-C.02483	-0.02384	-0.02204	-0.01966	-0.01712	-0.01518	-0.01418
15.9849	-C.03819	-0.03747	-0.03616	-0.03443	-0.03259	-0.03119	-0.03046
16.7262	-C.03130	-0.03080	-0.02990	-0.02871	-0.02744	-0.02648	-0.02598
17.4675	-C.02573	-0.02541	-0.02484	-0.02409	-0.02329	-0.02267	-0.02236
18.2088	-C.02129	-0.02112	-0.02080	-0.02038	-0.01994	-0.01960	-0.01943
18.9501	-C.01778	-0.01771	-0.01758	-0.01740	-0.01722	-0.01708	-0.01700
19.6914	-C.01500	-0.01500	-0.01499	-0.01498	-0.01497	-0.01497	-0.01496
20.4327	-C.01280	-0.01284	-0.01291	-0.01300	-0.01310	-0.01318	-0.01322
21.1740	-C.01104	-0.01110	-0.01122	-0.01137	-0.01153	-0.01165	-0.01171
21.9153	-C.00963	-0.00970	-0.00984	-0.01001	-0.01019	-0.01033	-0.01041
22.6566	-C.00849	-0.00856	-0.00869	-0.00887	-0.00905	-0.00920	-0.00927

23.3979	-C.00756	-0.00762	-0.00774	-0.00791	-0.00808	-0.00821	-0.00827
24.1392	-C.00678	-0.00684	-0.00695	-0.00709	-0.00724	-0.00735	-0.00741
24.8805	-C.00613	-0.00618	-0.00627	-0.00639	-0.00651	-0.00660	-0.00665
25.6218	-C.00558	-0.00562	-0.00569	-0.00578	-0.00588	-0.00596	-0.00600
26.3631	-C.00510	-0.00513	-0.00518	-0.00526	-0.00533	-0.00539	-0.00542
27.1044	-C.00468	-0.00470	-0.00474	-0.00480	-0.00486	-0.00490	-0.00492
27.8457	-C.00431	-0.00433	-0.00436	-0.00440	-0.00444	-0.00447	-0.00449
28.5870	-C.00399	-0.00400	-0.00402	-0.00404	-0.00407	-0.00410	-0.00411
29.3283	-C.00369	-0.00370	-0.00372	-0.00373	-0.00375	-0.00377	-0.00378
30.0696	-C.00588	-0.00588	-0.00589	-0.00590	-0.00591	-0.00592	-0.00593
30.8109	-C.01353	-0.01354	-0.01354	-0.01355	-0.01356	-0.01356	-0.01356
31.5522	-C.02134	-0.02134	-0.02135	-0.02135	-0.02135	-0.02136	-0.02136
32.2935	-C.02626	-0.02626	-0.02626	-0.02627	-0.02627	-0.02627	-0.02627
33.0346	-C.03009	-0.03009	-0.03009	-0.03009	-0.03010	-0.03010	-0.03010
33.7761	-C.03691	-0.03691	-0.03691	-0.03691	-0.03691	-0.03691	-0.03691
34.5174	-C.03925	-0.03925	-0.03925	-0.03925	-0.03925	-0.03925	-0.03925
35.2587	-C.02921	-0.02921	-0.02922	-0.02922	-0.02922	-0.02922	-0.02922
36.0000	-C.01158	-0.01158	-0.01158	-0.01158	-0.01158	-0.01159	-0.01159

INCREMENTAL PRESSURES, FORCES, AND MOMENTS ON BODY PANELS DUE TO WING

CD = 0.00001 CL = 0.00271 CM = -0.07155

BCDY PANEL PRESSURE CCEFFICIENT(CP)

THETA(CEG.)	12.5000	37.500C	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	C.00002	0.00006	0.00024	0.01706	-0.00548	-0.00022	-0.00007
2	-C.00001	0.00051	0.01889	0.05583	0.02264	0.00447	-0.00017
3	C.00111	0.01398	0.03247	0.02901	0.02290	0.01311	0.00390
4	0.01113	0.0217C	0.02304	0.01115	0.01303	0.01679	0.01156
5	C.02083	0.01823	0.01416	-0.00501	0.00787	0.01410	0.01605
6	0.02179	0.01547	0.00264	-0.01811	0.00691	0.01004	0.01624
7	C.01754	0.00988	-0.00569	-0.02925	0.00408	0.00905	0.01357
8	C.00959	0.00232	-0.01388	-0.C3880	0.00538	0.00642	0.00950
9	C.00084	-0.00609	-0.02228	-0.04548	0.00671	0.00504	0.00585
10	-C.00714	-0.01483	-0.03053	-0.04997	0.00555	0.00456	0.00461
11	-C.02345	-0.02920	-0.03209	-0.01561	0.00552	-0.00112	0.00454
12	-C.02771	-0.02372	-0.01544	-0.00101	0.00409	0.00126	0.00409
13	-C.01614	-0.01365	-0.00931	-0.00360	0.00099	0.00368	0.00630
14	-C.00872	-0.00772	-0.00542	-0.00227	0.00047	0.00252	0.00356

BCDY PANEL SLCPE.

THETA(CEG.)	12.5000	37.500C	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	C.02349	0.02349	0.02349	0.02339	0.02336	0.02349	0.02349
2	C.00619	0.00619	0.00619	0.00616	0.00615	0.00619	0.00619
3	C.	0.	0.	0.	-0.	-0.	-0.
4	C.	0.	0.	0.	-0.	-0.	-0.
5	C.	0.	0.	0.	-0.	-0.	-0.

6	C.	0.	0.	0.	-0.	-0.	-0.
7	C.	0.	0.	0.	-0.	-0.	-0.
8	C.	0.	0.	0.	-0.	-0.	-0.
9	C.	0.	0.	0.	-0.	-0.	-0.
10	C.	0.	0.	0.	-0.	-0.	-0.
11	C.	0.	0.	0.	-0.	-0.	-0.
12	C.	0.	0.	0.	-0.	-0.	-0.
13	C.	0.	0.	0.	-0.	-0.	-0.
14	-C.01422	-0.01422	-0.01422	-0.01415	-0.01413	-0.01422	-0.01422

PRESSURES, FORCES, AND MOMENTS ON WING PANELS IN PRESENCE OF BODY

CD = C.00073 CL = 0.04198 CM = -1.36562

SPANWISE CD DISTRIBUTION

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
	C.00019	0.00022	0.00017	0.00012	0.00008	0.00005	0.00003	0.00002	0.00002	-0.00017

SPANWISE CL DISTRIBUTION

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
	C.00154	0.00324	0.00379	0.00427	0.00465	0.00498	0.00526	0.00550	0.00282	0.00594

UPPER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	C.06876	0.03662	0.03029	0.01881	0.01317	0.00230	-0.00651	-0.01591	-0.01647	-0.01449
2	-C.01156	-0.02459	-0.03511	-0.04391	-0.05947	-0.06642	-0.07617	-0.07550	-0.07647	-0.07589
3	-C.02058	-0.03666	-0.05076	-0.05887	-0.07213	-0.08190	-0.08633	-0.08130	-0.08042	-0.08254
4	-C.02946	-0.04734	-0.06349	-0.07020	-0.08464	-0.09460	-0.08606	-0.08271	-0.08493	-0.08272
5	-C.04064	-0.05862	-0.07427	-0.08284	-0.09214	-0.08820	-0.08241	-0.08775	-0.08463	-0.09484
6	-C.04954	-0.07100	-0.08309	-0.09311	-0.08802	-0.07979	-0.08283	-0.08236	-0.08335	-0.08135
7	-C.06656	-0.07591	-0.08959	-0.08423	-0.08126	-0.07810	-0.07781	-0.07899	-0.07732	-0.07691
8	-C.06797	-0.07998	-0.08383	-0.07167	-0.06763	-0.06921	-0.06644	-0.06765	-0.06831	-0.06907
9	-C.07436	-0.08210	-0.06159	-0.05317	-0.05607	-0.05217	-0.05375	-0.05517	-0.05669	-0.05854
10	-C.07428	-0.03970	-0.02639	-0.03131	-0.02587	-0.02760	-0.02714	-0.03081	-0.03223	-0.04294

LOWER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	C.01197	-0.01960	-0.02282	-0.02655	-0.02240	-0.02165	-0.01850	-0.01693	-0.01172	0.00358
2	-C.00936	-0.01718	-0.02191	-0.02367	-0.03164	-0.03095	-0.03428	-0.02840	-0.02497	-0.02032
3	-C.01527	-0.02276	-0.02865	-0.02913	-0.03523	-0.03897	-0.03935	-0.03112	-0.02632	-0.02842
4	-C.01611	-0.02365	-0.02992	-0.02869	-0.03669	-0.04286	-0.03139	-0.02581	-0.02564	-0.02479
5	-C.01744	-0.02390	-0.02963	-0.03009	-0.03506	-0.02857	-0.02083	-0.02456	-0.02054	-0.03332
6	-C.01473	-0.02452	-0.02658	-0.03069	-0.02284	-0.01299	-0.01486	-0.01300	-0.01445	-0.01477
7	-C.01620	-0.01452	-0.02053	-0.01174	-0.00745	-0.00317	-0.00228	-0.00255	-0.00196	-0.00379
8	-C.00620	-0.00916	-0.00949	0.00421	0.00899	0.00791	0.01105	0.01013	0.00801	0.00527

9	-0.00184	-0.00638	0.01272	0.02179	0.01917	0.02324	0.02193	0.02014	0.01715	0.01463
10	-0.00364	0.02603	0.03833	0.03390	0.03915	0.03742	0.03809	0.03369	0.03138	0.02986

WING PANEL PRESSURE DIFFERENCE(CL)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.07192	-0.07385	-0.07128	-0.06437	-0.05484	-0.04369	-0.03194	-0.02111	-0.01528	-0.00194
2	-0.00115	0.00414	0.01014	0.01746	0.02563	0.03371	0.04070	0.04601	0.05053	0.05461
3	0.00432	0.01324	0.02194	0.03000	0.03787	0.04451	0.04891	0.05192	0.05584	0.05599
4	0.01355	0.02447	0.03513	0.04352	0.05086	0.05534	0.05780	0.05986	0.06241	0.06089
5	0.02392	0.03625	0.04707	0.05582	0.06080	0.06312	0.06468	0.06668	0.06736	0.06547
6	0.03534	0.04814	0.05901	0.06566	0.06807	0.06908	0.07046	0.07182	0.07143	0.06895
7	0.05001	0.06161	0.07024	0.07323	0.07429	0.07514	0.07570	0.07669	0.07549	0.07323
8	0.05915	0.06892	0.07266	0.07321	0.07363	0.07422	0.07437	0.07474	0.07336	0.07147
9	0.06684	0.06849	0.06751	0.06754	0.06801	0.06790	0.06827	0.06800	0.06668	0.06614
10	0.05949	0.05247	0.05075	0.05148	0.05101	0.05110	0.05128	0.05077	0.04999	0.05948

UPPER SURFACE WING PANEL SLOPE

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.15454	0.15454	0.15454	0.15454	0.15454	0.15454	0.15454	0.15454	0.15454	0.15454
2	0.06688	0.06688	0.06688	0.06688	0.06688	0.06688	0.06688	0.06688	0.06688	0.06688
3	0.03570	0.03570	0.03570	0.03570	0.03570	0.03570	0.03570	0.03570	0.03570	0.03570
4	0.01179	0.01179	0.01179	0.01179	0.01179	0.01179	0.01179	0.01179	0.01179	0.01179
5	-0.00888	-0.00888	-0.00888	-0.00888	-0.00888	-0.00888	-0.00888	-0.00888	-0.00888	-0.00888
6	-0.02980	-0.02980	-0.02980	-0.02980	-0.02980	-0.02980	-0.02980	-0.02980	-0.02980	-0.02980
7	-0.05236	-0.05236	-0.05236	-0.05236	-0.05236	-0.05236	-0.05236	-0.05236	-0.05236	-0.05236
8	-0.07034	-0.07034	-0.07034	-0.07034	-0.07034	-0.07034	-0.07034	-0.07034	-0.07034	-0.07034
9	-0.09027	-0.09027	-0.09027	-0.09027	-0.09027	-0.09027	-0.09027	-0.09027	-0.09027	-0.09027
10	-0.11584	-0.11584	-0.11584	-0.11584	-0.11584	-0.11584	-0.11584	-0.11584	-0.11584	-0.11584

LOWER SURFACE WING PANEL SLOPE

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.06204	-0.06204	-0.06204	-0.06204	-0.06204	-0.06204	-0.06204	-0.06204	-0.06204	-0.06204
2	-0.03355	-0.03355	-0.03355	-0.03355	-0.03355	-0.03355	-0.03355	-0.03355	-0.03355	-0.03355
3	-0.01427	-0.01427	-0.01427	-0.01427	-0.01427	-0.01427	-0.01427	-0.01427	-0.01427	-0.01427
4	-0.00210	-0.00210	-0.00210	-0.00210	-0.00210	-0.00210	-0.00210	-0.00210	-0.00210	-0.00210
5	0.00636	0.00636	0.00636	0.00636	0.00636	0.00636	0.00636	0.00636	0.00636	0.00636
6	0.01415	0.01415	0.01415	0.01415	0.01415	0.01415	0.01415	0.01415	0.01415	0.01415
7	0.01477	0.01477	0.01477	0.01477	0.01477	0.01477	0.01477	0.01477	0.01477	0.01477
8	0.01035	0.01035	0.01035	0.01035	0.01035	0.01035	0.01035	0.01035	0.01035	0.01035
9	0.00212	0.00212	0.00212	0.00212	0.00212	0.00212	0.00212	0.00212	0.00212	0.00212
10	-0.01224	-0.01224	-0.01224	-0.01224	-0.01224	-0.01224	-0.01224	-0.01224	-0.01224	-0.01224

FORCES AND MOMENTS ON WING-BODY COMBINATION

CD = 0.00147 CL = 0.04500 CM = -1.44274

TR-805 GIVEN SHAPE

DELTA ANGLE OF ATTACK = 2.0000 DEG.

ANGLE OF ATTACK WITH RESPECT TO BODY AXIS = 2.0000 DEG.

VELOCITY COMPONENTS ON BODY DUE TO BODY LINE SOURCES AND DOUBLETS

AXIAL(U)

THETA(DEG.) X	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
0.	-0.06452	-0.06416	-0.06352	-0.06268	-0.06178	-0.06110	-0.06075
0.7243	-0.06452	-0.06416	-0.06352	-0.06268	-0.06178	-0.06110	-0.06075
1.4485	-0.05796	-0.05884	-0.06042	-0.06251	-0.06473	-0.06642	-0.06729
2.1728	-0.02660	-0.02737	-0.02877	-0.03062	-0.03257	-0.03406	-0.03483
2.8970	-0.02545	-0.02639	-0.02809	-0.03032	-0.03270	-0.03451	-0.03544
3.6213	-0.01800	-0.01885	-0.02039	-0.02242	-0.02458	-0.02622	-0.02707
4.3456	-0.01800	-0.01883	-0.02035	-0.02235	-0.02447	-0.02609	-0.02692
5.0698	-0.01404	-0.01475	-0.01603	-0.01773	-0.01953	-0.02091	-0.02162
5.7941	-0.01403	-0.01466	-0.01581	-0.01733	-0.01894	-0.02017	-0.02081
6.5184	-0.01059	-0.01123	-0.01240	-0.01395	-0.01558	-0.01683	-0.01748
7.2426	-0.00944	-0.01017	-0.01149	-0.01323	-0.01507	-0.01648	-0.01721
7.9669	-0.00753	-0.00811	-0.00916	-0.01055	-0.01202	-0.01315	-0.01373
8.6911	-0.00529	-0.00610	-0.00757	-0.00951	-0.01156	-0.01313	-0.01394
9.4154	-0.00369	-0.00440	-0.00568	-0.00738	-0.00918	-0.01055	-0.01126
10.1397	-0.00209	-0.00285	-0.00424	-0.00607	-0.00801	-0.00949	-0.01025
10.8639	0.00006	-0.00074	-0.00220	-0.00412	-0.00616	-0.00772	-0.00852
11.5882	0.00128	0.00060	-0.00065	-0.00229	-0.00403	-0.00535	-0.00604
12.3124	0.00331	0.00259	0.00127	-0.00046	-0.00230	-0.00370	-0.00443
13.0367	0.00611	0.00536	0.00401	0.00222	0.00033	-0.00112	-0.00186
13.7610	0.00743	0.00667	0.00530	0.00350	0.00158	0.00012	-0.00063
14.5023	0.01139	0.01050	0.00889	0.00677	0.00452	0.00281	0.00192
15.2436	0.01407	0.01326	0.01179	0.00984	0.00778	0.00620	0.00539
15.9849	0.02042	0.01987	0.01889	0.01758	0.01620	0.01515	0.01460
16.7262	0.01629	0.01597	0.01538	0.01460	0.01378	0.01315	0.01283
17.4675	0.01304	0.01288	0.01259	0.01221	0.01181	0.01150	0.01134
18.2088	0.01053	0.01048	0.01041	0.01030	0.01020	0.01011	0.01007
18.9501	0.00860	0.00864	0.00870	0.00878	0.00886	0.00893	0.00896
19.6914	0.00713	0.00721	0.00736	0.00754	0.00775	0.00790	0.00798
20.4327	0.00601	0.00611	0.00630	0.00654	0.00680	0.00699	0.00709
21.1740	0.00515	0.00526	0.00545	0.00571	0.00598	0.00619	0.00630
21.9153	0.00448	0.00459	0.00478	0.00502	0.00529	0.00549	0.00559
22.6566	0.00396	0.00406	0.00423	0.00445	0.00469	0.00487	0.00496
23.3979	0.00355	0.00363	0.00377	0.00396	0.00417	0.00432	0.00440
24.1392	0.00321	0.00327	0.00339	0.00355	0.00372	0.00385	0.00392
24.8805	0.00292	0.00298	0.00307	0.00320	0.00334	0.00344	0.00349
25.6218	0.00268	0.00272	0.00280	0.00290	0.00300	0.00308	0.00312
26.3631	0.00247	0.00250	0.00256	0.00263	0.00271	0.00277	0.00281
27.1044	0.00228	0.00231	0.00235	0.00240	0.00246	0.00251	0.00253
27.8457	0.00211	0.00213	0.00216	0.00220	0.00225	0.00228	0.00230
28.5870	0.00196	0.00198	0.00200	0.00203	0.00206	0.00208	0.00209
29.3283	0.00183	0.00183	0.00185	0.00187	0.00189	0.00191	0.00192
30.0696	0.00287	0.00289	0.00292	0.00296	0.00300	0.00303	0.00304
30.8109	0.00649	0.00655	0.00665	0.00678	0.00692	0.00702	0.00708
31.5522	0.00998	0.01011	0.01033	0.01063	0.01095	0.01119	0.01132

32.2935	C.01189	0.01210	0.01249	0.01300	0.01354	0.01396	0.01417
33.0348	C.01313	0.01345	0.01402	0.01477	0.01557	0.01618	0.01650
33.7761	C.01563	0.01607	0.01687	0.01793	0.01905	0.01990	0.02034
34.5174	C.01582	0.01639	0.01743	0.01880	0.02026	0.02136	0.02194
35.2587	C.01036	0.01100	0.01218	0.01372	0.01536	0.01661	0.01726
36.0000	C.00193	0.00255	0.00367	0.00516	0.00673	0.00793	0.00855

RADIAL(VR)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
X							
0.	C.17475	C.17362	0.17157	0.16888	0.16601	0.16383	0.16271
0.7243	C.17475	0.17362	0.17157	0.16888	0.16601	0.16383	0.16271
1.4485	C.15974	0.16142	0.16447	0.16849	0.17276	0.17601	0.17769
2.1728	C.10628	0.10853	0.11261	0.11798	0.12370	0.12804	0.13029
2.8970	C.09805	0.10104	0.10646	0.11362	0.12121	0.12699	0.12998
3.6213	C.08157	0.08484	0.09077	0.09858	0.10688	0.11319	0.11646
4.3456	C.07757	0.08110	0.08749	0.09591	0.10486	0.11167	0.11520
5.0698	C.06824	0.07179	0.07821	0.08668	0.09568	0.10253	0.10607
5.7941	C.06561	0.06916	0.07560	0.08409	0.09311	0.09997	0.10353
6.5184	C.05791	0.06160	0.06827	0.07707	0.08642	0.09354	0.09722
7.2426	C.05374	0.05768	0.06480	0.07420	0.08418	0.09177	0.09570
7.9669	C.04851	0.05233	0.05925	0.06838	0.07807	0.08546	0.08927
8.6911	C.04262	0.04693	0.05475	0.06505	0.07600	0.08433	0.08864
9.4154	C.03758	0.04191	0.04974	0.06007	0.07104	0.07939	0.08371
10.1397	C.03250	0.03708	0.04537	0.05630	0.06791	0.07675	0.08132
10.8639	C.02639	0.03122	0.03997	0.05151	0.06376	0.07309	0.07792
11.5882	C.02160	0.02644	0.03521	0.04677	0.05905	0.06840	0.07324
12.3124	C.01545	0.02054	0.02977	0.04194	0.05486	0.06470	0.06979
13.0367	C.00773	0.01308	0.02278	0.03556	0.04914	0.05947	0.06482
13.7610	C.00203	0.00763	0.01779	0.03118	0.04541	0.05624	0.06184
14.5023	-C.00842	-0.00230	0.00879	0.02341	0.03894	0.05076	0.05687
15.2436	-C.01770	-0.01132	0.00023	0.01546	0.03164	0.04395	0.05032
15.9849	-C.03408	-0.02769	-0.01611	-0.00085	0.01536	0.02770	0.03408
16.7262	-C.03408	-0.02769	-0.01612	-0.00085	0.01536	0.02769	0.03408
17.4675	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
18.2088	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
18.9501	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
19.6914	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
20.4327	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
21.1740	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
21.9153	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
22.6566	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
23.3979	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
24.1392	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
24.8805	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
25.6218	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
26.3631	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
27.1044	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
27.8457	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
28.5870	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
29.3283	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
30.0696	-C.03617	-0.02978	-0.01820	-0.00294	0.01327	0.02561	0.03199
30.8109	-C.04323	-0.03684	-0.02527	-0.01001	0.00620	0.01854	0.02492
31.5522	-C.05147	-0.04508	-0.03351	-0.01826	-0.00205	0.01028	0.01667
32.2935	-C.05832	-0.05194	-0.04038	-0.02513	-0.00893	0.00340	0.00978
33.0348	-C.06495	-0.05857	-0.04701	-0.03178	-0.01559	-0.00327	0.00311
33.7761	-C.07470	-0.06834	-0.05679	-0.04157	-0.02541	-0.01310	-0.00673

34.5174	-C.08180	-0.07544	-0.06391	-0.04871	-0.03257	-0.02028	-0.01392
35.2587	-C.07395	-0.07259	-0.06107	-0.04587	-0.02974	-0.01745	-0.01110
36.0000	-C.06814	-0.06178	-0.05024	-0.03503	-0.01887	-0.00658	-0.00021

TANGENTIAL(VT)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
X							
0.	C.00099	0.00278	0.00405	0.00456	0.00410	0.00278	0.00099
0.7243	C.00099	0.00278	0.00405	0.00456	0.00410	0.00278	0.00099
1.4465	-C.00081	-0.00227	-0.00331	-0.00373	-0.00335	-0.00227	-0.00081
2.1728	-C.00190	-0.00533	-0.00777	-0.00876	-0.00787	-0.00533	-0.00190
2.8970	-C.00263	-0.00739	-0.01078	-0.01214	-0.01091	-0.00739	-0.00263
3.6213	-C.00318	-0.00895	-0.01304	-0.01469	-0.01320	-0.00895	-0.00318
4.3456	-C.00354	-0.00995	-0.01450	-0.01634	-0.01468	-0.00995	-0.00354
5.0698	-0.00377	-C.01061	-0.01546	-0.01742	-0.01565	-0.01061	-0.00377
5.7941	-C.00388	-0.01091	-0.01590	-0.01792	-0.01610	-0.01091	-0.00388
6.5184	-C.00400	-0.01124	-0.01638	-0.01846	-0.01658	-0.01124	-0.00400
7.2426	-C.00416	-0.01170	-0.01704	-0.01921	-0.01726	-0.01170	-0.00416
7.9669	-C.00427	-0.01202	-0.01751	-0.01973	-0.01773	-0.01202	-0.00427
8.6911	-C.00448	-0.01261	-0.01837	-0.02070	-0.01860	-0.01261	-0.00448
9.4154	-C.00470	-0.01321	-0.01925	-0.02169	-0.01949	-0.01321	-0.00470
10.1397	-C.00492	-0.01384	-0.02017	-0.02273	-0.02042	-0.01384	-0.00492
10.8639	-C.00519	-0.01459	-0.02126	-0.02396	-0.02153	-0.01459	-0.00519
11.5882	-C.00542	-0.01523	-0.02220	-0.02502	-0.02247	-0.01523	-0.00542
12.3124	-C.00566	-0.01591	-0.02318	-0.02612	-0.02347	-0.01591	-0.00566
13.0367	-C.00594	-0.01671	-0.02435	-0.02744	-0.02465	-0.01671	-0.00594
13.7610	-C.00625	-0.01758	-0.02561	-0.02887	-0.02593	-0.01758	-0.00625
14.5023	-C.00668	-0.01878	-0.02736	-0.03083	-0.02770	-0.01878	-0.00668
15.2436	-C.00715	-0.02010	-0.02929	-0.03301	-0.02966	-0.02010	-0.00715
15.9849	-C.00762	-0.02144	-0.03124	-0.03521	-0.03163	-0.02144	-0.00762
16.7262	-C.00793	-0.02231	-0.03251	-0.03664	-0.03292	-0.02231	-0.00793
17.4675	-C.00811	-0.02281	-0.03324	-0.03747	-0.03366	-0.02281	-0.00811
18.2088	-C.00819	-0.02303	-0.03356	-0.03783	-0.03398	-0.02303	-0.00819
18.9501	-C.00820	-0.02306	-0.03360	-0.03787	-0.03402	-0.02306	-0.00820
19.6914	-C.00816	-0.02296	-0.03345	-0.03770	-0.03387	-0.02296	-0.00816
20.4327	-C.00810	-0.02279	-0.03320	-0.03742	-0.03361	-0.02279	-0.00810
21.1740	-C.00803	-0.02258	-0.03290	-0.03708	-0.03331	-0.02258	-0.00803
21.9153	-C.00795	-0.02237	-0.03260	-0.03674	-0.03300	-0.02237	-0.00795
22.6566	-C.00788	-0.02218	-0.03231	-0.03642	-0.03271	-0.02218	-0.00788
23.3979	-C.00782	-0.02200	-0.03206	-0.03613	-0.03246	-0.02200	-0.00782
24.1392	-C.00777	-0.02185	-0.03184	-0.03589	-0.03224	-0.02185	-0.00777
24.8805	-C.00773	-0.02173	-0.03166	-0.03569	-0.03206	-0.02173	-0.00773
25.6218	-C.00769	-0.02163	-0.03152	-0.03553	-0.03191	-0.02163	-0.00769
26.3631	-C.00766	-0.02156	-0.03141	-0.03540	-0.03180	-0.02156	-0.00766
27.1044	-C.00764	-0.02150	-0.03133	-0.03531	-0.03172	-0.02150	-0.00764
27.8457	-C.00763	-0.02146	-0.03127	-0.03524	-0.03166	-0.02146	-0.00763
28.5870	-C.00762	-0.02143	-0.03122	-0.03519	-0.03161	-0.02143	-0.00762
29.3283	-C.00761	-0.02141	-0.03119	-0.03515	-0.03158	-0.02141	-0.00761
30.0696	-C.00762	-0.02143	-0.03123	-0.03520	-0.03162	-0.02143	-0.00762
30.8109	-C.00768	-0.02159	-0.03146	-0.03546	-0.03185	-0.02159	-0.00768
31.5522	-C.00777	-0.02186	-0.03184	-0.03589	-0.03224	-0.02186	-0.00777
32.2935	-C.00787	-0.02214	-0.03227	-0.03637	-0.03267	-0.02214	-0.00787
33.0348	-C.00797	-0.02243	-0.03268	-0.03683	-0.03308	-0.02243	-0.00797
33.7761	-C.00809	-0.02274	-0.03314	-0.03735	-0.03355	-0.02274	-0.00809
34.5174	-C.00818	-0.02300	-0.03352	-0.03777	-0.03393	-0.02300	-0.00818
35.2587	-C.00817	-0.02297	-0.03347	-0.03772	-0.03389	-0.02297	-0.00817

36.0000	-0.00802	-0.02257	-0.03288	-0.03706	-0.03329	-0.02257	-0.00802

VELOCITY COMPONENTS ON WING PANELS DUE TO BODY PANEL PRESSURE SINGULARITIES

AXIAL(U)

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
CHORDWISE STATION										
1	-0.00632	-0.00169	-0.00012	0.00413	0.00252	0.00214	0.00051	-0.00014	-0.00047	-0.00265
2	-0.00303	-0.00009	0.00331	0.00169	0.00314	0.00078	0.00002	-0.00126	-0.00072	-0.00158
3	-0.00031	0.00244	0.00404	0.00081	0.00091	0.00011	-0.00045	-0.00092	-0.00223	-0.00080
4	0.00288	0.00512	0.00330	-0.00061	-0.00004	-0.00026	-0.00075	-0.00214	-0.00181	-0.00097
5	0.00707	0.00451	0.00235	-0.00041	-0.00248	-0.00073	-0.00306	-0.00010	-0.00108	0.00604
6	0.00573	0.00487	0.00151	-0.00006	-0.00077	-0.00370	-0.00206	-0.00117	0.00016	0.00037
7	0.00791	0.00235	0.00053	-0.00179	-0.00091	-0.00209	-0.00110	0.00030	0.00037	0.00045
8	0.00442	0.00099	-0.00022	-0.00134	-0.00251	-0.00091	-0.00095	0.00037	0.00044	0.00023
9	0.00269	-0.00045	-0.00087	-0.00326	-0.00123	-0.00121	0.00046	0.00050	0.00019	0.00015
10	0.00032	-0.00088	-0.00454	-0.00031	-0.00148	0.00063	0.00050	0.00019	0.00012	0.00013

TRANSVERSE(V)

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
CHORDWISE STATION										
1	0.00992	0.00428	0.00120	-0.00581	-0.00390	-0.00358	-0.00117	-0.00015	0.00043	0.00385
2	0.00851	0.00238	-0.00408	-0.00260	-0.00511	-0.00167	-0.00048	0.00157	0.00084	0.00233
3	0.00673	-0.00140	-0.00584	-0.00167	-0.00201	-0.00073	0.00025	0.00110	0.00322	0.00123
4	0.00220	-0.00662	-0.00541	0.00003	-0.00072	-0.00019	0.00074	0.00308	0.00271	0.00153
5	-0.00609	-0.00701	-0.00479	-0.00052	0.00287	0.00055	0.00434	0.00014	0.00168	-0.00893
6	-0.00751	-0.00922	-0.00408	-0.00118	0.00037	0.00512	0.00302	0.00181	-0.00012	-0.00048
7	-0.01422	-0.00703	-0.00298	0.00141	0.00065	0.00297	0.00169	-0.00032	-0.00048	-0.00061
8	-0.01341	-0.00621	-0.00197	0.00081	0.00340	0.00135	0.00155	-0.00046	-0.00060	-0.00030
9	-0.01451	-0.00481	-0.00098	0.00408	0.00171	0.00192	-0.00058	-0.00068	-0.00024	-0.00018
10	-0.01399	-0.00429	0.00495	0.00008	0.00226	-0.00083	-0.00067	-0.00024	-0.00014	-0.00016

VERTICAL(W)

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
CHORDWISE STATION										
1	0.00015	0.00205	0.00158	0.00158	0.00171	0.00165	0.00160	0.00159	0.00148	0.00133
2	0.00721	0.00366	0.00218	0.00219	0.00223	0.00204	0.00189	0.00170	0.00168	0.00156
3	0.01781	0.00579	0.00359	0.00296	0.00262	0.00239	0.00207	0.00196	0.00177	0.00177
4	0.03047	0.00824	0.00524	0.00352	0.00310	0.00264	0.00236	0.00208	0.00205	0.00183
5	0.04498	0.01186	0.00639	0.00443	0.00338	0.00300	0.00244	0.00246	0.00214	0.00225
6	0.06211	0.01534	0.00782	0.00534	0.00399	0.00311	0.00290	0.00248	0.00234	0.00210
7	0.08261	0.01897	0.00917	0.00571	0.00453	0.00358	0.00309	0.00272	0.00241	0.00218
8	0.10290	0.02256	0.01004	0.00653	0.00475	0.00402	0.00318	0.00280	0.00247	0.00218
9	0.12437	0.02485	0.01121	0.00677	0.00536	0.00410	0.00345	0.00287	0.00240	0.00222
10	0.14031	0.02781	0.01161	0.00789	0.00551	0.00441	0.00354	0.00279	0.00241	0.00227

VELOCITY COMPONENTS ON WING PANELS DUE TO WING PANEL PRESSURE SINGULARITIES

AXIAL(U)

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
CHORDWISE STATION										

1	0.00692	0.01455	0.01982	0.02537	0.03133	0.03726	0.04278	0.04747	0.04988	0.05525
2	0.01517	0.02048	0.02448	0.02865	0.03266	0.03622	0.03908	0.04118	0.04298	0.04452
3	0.01543	0.01874	0.02282	0.02652	0.02970	0.03212	0.03368	0.03477	0.03627	0.03623
4	0.01631	0.01971	0.02390	0.02713	0.02954	0.03088	0.03162	0.03226	0.03318	0.03252
5	0.01761	0.02165	0.02540	0.02810	0.02938	0.02987	0.03024	0.03085	0.03104	0.03026
6	0.02013	0.02388	0.02702	0.02860	0.02888	0.02893	0.02922	0.02963	0.02942	0.02845
7	0.02356	0.02642	0.02832	0.02845	0.02833	0.02832	0.02836	0.02866	0.02817	0.02728
8	0.02534	0.02704	0.02699	0.02643	0.02613	0.02610	0.02604	0.02613	0.02557	0.02480
9	0.02614	0.02486	0.02344	0.02284	0.02269	0.02248	0.02255	0.02241	0.02187	0.02152
10	0.02158	0.01780	0.01653	0.01643	0.01609	0.01603	0.01608	0.01585	0.01553	0.01811

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.01902	-0.03997	-0.05444	-0.06970	-0.08608	-0.10237	-0.11752	-0.13041	-0.13705	-0.15180
2	-0.04444	-0.05627	-0.06724	-0.07870	-0.08973	-0.09950	-0.10737	-0.11314	-0.11809	-0.12230
3	-0.04239	-0.05148	-0.06268	-0.07286	-0.08159	-0.08826	-0.09253	-0.09553	-0.09964	-0.09953
4	-0.04482	-0.05414	-0.06566	-0.07453	-0.08114	-0.08483	-0.08687	-0.08864	-0.09117	-0.08934
5	-0.04892	-0.05948	-0.06977	-0.07719	-0.08071	-0.08207	-0.08307	-0.08477	-0.08528	-0.08313
6	-0.05532	-0.06561	-0.07424	-0.07859	-0.07935	-0.07947	-0.08027	-0.08140	-0.08081	-0.07817
7	-0.06474	-0.07260	-0.07780	-0.07817	-0.07783	-0.07781	-0.07792	-0.07875	-0.07738	-0.07495
8	-0.06961	-0.07430	-0.07415	-0.07261	-0.07180	-0.07172	-0.07155	-0.07179	-0.07025	-0.06813
9	-0.07181	-0.06831	-0.06440	-0.06274	-0.06233	-0.06177	-0.06196	-0.06158	-0.06009	-0.05913
10	-0.05929	-0.04890	-0.04541	-0.04514	-0.04419	-0.04405	-0.04419	-0.04355	-0.04265	-0.04975

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.00758	-0.00247	0.00314	0.00562	0.00688	0.00778	0.00834	0.00870	0.00899	0.00937
2	-0.04851	-0.03392	-0.02695	-0.02440	-0.02310	-0.02210	-0.02147	-0.02096	-0.02077	-0.02044
3	-0.06591	-0.04204	-0.03412	-0.03096	-0.02931	-0.02833	-0.02755	-0.02714	-0.02678	-0.02661
4	-0.08488	-0.05021	-0.04144	-0.03726	-0.03558	-0.03440	-0.03368	-0.03311	-0.03293	-0.03254
5	-0.10551	-0.05965	-0.04849	-0.04414	-0.04187	-0.04081	-0.03983	-0.03957	-0.03910	-0.03907
6	-0.12899	-0.06945	-0.05632	-0.05151	-0.04899	-0.04745	-0.04683	-0.04614	-0.04586	-0.04550
7	-0.16011	-0.08379	-0.06849	-0.06277	-0.06046	-0.05887	-0.05798	-0.05735	-0.05690	-0.05658
8	-0.19125	-0.09836	-0.08043	-0.07473	-0.07184	-0.07049	-0.06926	-0.06862	-0.06820	-0.06786
9	-0.22647	-0.11454	-0.09559	-0.08899	-0.08650	-0.08463	-0.08360	-0.08282	-0.08227	-0.08204
10	-0.26210	-0.13733	-0.11590	-0.11005	-0.10660	-0.10490	-0.10368	-0.10278	-0.10229	-0.10209

VELOCITY COMPONENTS ON WING PANELS DUE TO BODY LINE SOURCES AND DOUBLETS

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.00754	0.01210	0.00794	0.00565	0.00418	0.00318	0.00251	0.00201	0.00173	0.00134
2	0.01493	0.00981	0.00679	0.00480	0.00363	0.00278	0.00221	0.00179	0.00155	0.00125
3	0.01183	0.00829	0.00572	0.00414	0.00313	0.00245	0.00196	0.00161	0.00141	0.00116
4	0.00991	0.00704	0.00486	0.00360	0.00275	0.00218	0.00176	0.00146	0.00127	0.00108
5	0.00837	0.00602	0.00419	0.00313	0.00243	0.00194	0.00159	0.00132	0.00116	0.00101
6	0.00711	0.00507	0.00365	0.00275	0.00217	0.00174	0.00144	0.00121	0.00107	0.00095
7	0.00595	0.00437	0.00320	0.00244	0.00194	0.00158	0.00131	0.00111	0.00099	0.00147
8	0.00509	0.00381	0.00281	0.00218	0.00174	0.00143	0.00120	0.00113	0.00205	0.00304
9	0.00441	0.00334	0.00249	0.00195	0.00158	0.00131	0.00110	0.00226	0.00377	0.00444

10	0.00384	0.00293	0.00223	0.00176	0.00144	0.00120	0.00154	0.00403	0.00510	0.00511

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.02129	-0.00164	-0.00314	-0.00304	-0.00259	-0.00211	-0.00175	-0.00143	-0.00125	-0.00099
2	0.00669	-0.00048	-0.00252	-0.00239	-0.00213	-0.00175	-0.00147	-0.00123	-0.00108	-0.00089
3	0.00759	-0.00004	-0.00183	-0.00191	-0.00170	-0.00146	-0.00123	-0.00106	-0.00094	-0.00079
4	0.00753	0.00034	-0.00129	-0.00154	-0.00139	-0.00123	-0.00105	-0.00091	-0.00082	-0.00071
5	0.00750	0.00065	-0.00092	-0.00119	-0.00115	-0.00103	-0.00091	-0.00079	-0.00072	-0.00064
6	0.00749	0.00108	-0.00064	-0.00094	-0.00095	-0.00087	-0.00078	-0.00069	-0.00063	-0.00058
7	0.00770	0.00130	-0.00039	-0.00074	-0.00078	-0.00074	-0.00067	-0.00061	-0.00056	-0.00139
8	0.00772	0.00147	-0.00018	-0.00058	-0.00065	-0.00063	-0.00059	-0.00070	-0.00223	-0.00384
9	0.00772	0.00160	-0.00003	-0.00044	-0.00054	-0.00054	-0.00051	-0.00247	-0.00496	-0.00610
10	0.00772	0.00175	0.00010	-0.00032	-0.00044	-0.00046	-0.00125	-0.00531	-0.00717	-0.00723

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.01877	0.01176	0.00662	0.00413	0.00275	0.00192	0.00139	0.00105	0.00087	0.00065
2	0.02306	0.01202	0.00653	0.00398	0.00262	0.00183	0.00134	0.00102	0.00085	0.00065
3	0.02391	0.01206	0.00634	0.00382	0.00250	0.00175	0.00129	0.00099	0.00083	0.00065
4	0.02435	0.01191	0.00613	0.00367	0.00241	0.00169	0.00125	0.00096	0.00081	0.00065
5	0.02437	0.01166	0.00594	0.00354	0.00233	0.00165	0.00122	0.00095	0.00080	0.00066
6	0.02415	0.01136	0.00576	0.00344	0.00227	0.00161	0.00120	0.00093	0.00079	0.00067
7	0.02380	0.01112	0.00561	0.00335	0.00222	0.00158	0.00119	0.00092	0.00078	0.00070
8	0.02344	0.01090	0.00550	0.00329	0.00219	0.00156	0.00118	0.00092	0.00082	0.00078
9	0.02312	0.01072	0.00541	0.00325	0.00216	0.00155	0.00117	0.00097	0.00089	0.00084
10	0.02284	0.01057	0.00534	0.00321	0.00215	0.00154	0.00119	0.00105	0.00093	0.00087

VELOCITY COMPONENTS ON WING PANELS DUE TO WING SOURCES

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.03054	-0.02433	-0.01880	-0.01746	-0.01326	-0.00857	-0.00441	-0.00076	-0.00148	-0.00331
2	-0.00716	-0.00013	0.00337	0.00784	0.01289	0.01731	0.02137	0.02116	0.02006	0.01976
3	-0.00189	0.00375	0.00866	0.01475	0.01964	0.02390	0.02566	0.02337	0.02313	0.02314
4	-0.00060	0.00495	0.01303	0.01903	0.02368	0.02789	0.02409	0.02362	0.02363	0.02249
5	-0.00045	0.00850	0.01654	0.02209	0.02667	0.02357	0.02284	0.02278	0.02177	0.02115
6	0.00270	0.01161	0.01874	0.02411	0.02187	0.02069	0.02053	0.01941	0.01883	0.01855
7	0.00547	0.01297	0.01968	0.01900	0.01691	0.01658	0.01565	0.01475	0.01439	0.01431
8	0.00716	0.01399	0.01652	0.01237	0.01172	0.01114	0.00986	0.00933	0.00908	0.00913
9	0.00900	0.01548	0.00706	0.00593	0.00571	0.00393	0.00330	0.00293	0.00274	0.00307
10	0.01057	-0.00233	-0.00442	-0.00488	-0.00666	-0.00745	-0.00788	-0.00816	-0.00831	-0.00557

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.02346	0.03708	0.03491	0.03503	0.02960	0.02265	0.01605	0.01001	0.01062	0.01346
2	-0.02849	-0.02351	-0.02451	-0.02979	-0.03719	-0.04422	-0.05097	-0.05142	-0.05010	-0.04997

3	-0.03063	-0.02966	-0.03436	-0.04316	-0.05102	-0.05823	-0.06183	-0.05896	-0.05886	-0.05917
4	-0.02439	-0.02819	-0.03974	-0.04951	-0.05761	-0.06514	-0.06024	-0.05998	-0.06022	-0.05877
5	-0.01502	-0.02915	-0.04278	-0.05283	-0.06136	-0.05790	-0.05738	-0.05766	-0.05636	-0.05561
6	-0.00956	-0.02868	-0.04278	-0.05337	-0.05185	-0.05088	-0.05112	-0.04977	-0.04905	-0.04882
7	-0.00345	-0.02458	-0.03956	-0.04162	-0.03967	-0.03983	-0.03886	-0.03773	-0.03731	-0.03742
8	0.00316	-0.02012	-0.03011	-0.02594	-0.02587	-0.02555	-0.02394	-0.02333	-0.02304	-0.02340
9	0.00822	-0.01726	-0.00930	-0.00904	-0.00947	-0.00721	-0.00649	-0.00607	-0.00586	-0.00692
10	0.01231	0.02030	0.02082	0.02039	0.02246	0.02335	0.02383	0.02413	0.02430	0.01669

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829
2	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022
3	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498
4	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695
5	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762
6	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197
7	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357
8	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034
9	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620
10	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180

VELOCITY COMPONENTS ON BODY PANELS DUE TO BODY PANEL PRESSURE SINGULARITIES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.00000	0.00001	0.00004	0.00050	-0.00050	-0.00004	-0.00001
2	-0.00002	0.00001	0.00064	-0.00546	-0.00808	0.00014	-0.00012
3	-0.00007	0.00194	-0.00191	0.00060	-0.01056	-0.00301	0.00175
4	0.00675	-0.00087	-0.00010	0.00492	-0.00864	-0.00265	0.00277
5	0.00285	0.00553	0.00638	0.01293	-0.00185	-0.00279	-0.00247
6	0.00087	0.00397	0.00841	0.01469	-0.00389	-0.00371	-0.00547
7	0.00065	0.00339	0.00823	0.01541	-0.00492	-0.00594	-0.00697
8	0.00152	0.00355	0.00807	0.01573	-0.00755	-0.00710	-0.00759
9	0.00209	0.00393	0.00845	0.01519	-0.00964	-0.00847	-0.00817
10	0.00208	0.00452	0.00886	0.01438	-0.01071	-0.00979	-0.00961
11	0.00493	0.00626	0.00659	0.00465	-0.00887	-0.00997	-0.01145
12	0.00779	0.00672	0.00284	-0.00384	-0.00721	-0.00779	-0.00903
13	0.00673	0.00539	0.00319	0.00027	-0.00248	-0.00445	-0.00619
14	0.00364	0.00313	0.00196	0.00029	-0.00120	-0.00245	-0.00310

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.00000	-0.00002	-0.00004	-0.00126	0.00106	0.00007	0.00001
2	-0.00004	0.00005	-0.00141	0.01005	0.01727	0.00010	0.00012
3	0.00027	-0.00109	0.00049	0.00070	0.03087	0.00237	0.00143
4	0.00021	-0.00057	0.00072	-0.01713	0.04355	-0.00088	-0.00313

5	-C.00274	-0.00673	-0.00786	-0.04622	0.04586	-0.00091	-0.00159
6	-C.00214	-0.00680	-0.01146	-0.07028	0.05862	0.00331	-0.00192
7	-C.00161	-0.00456	-0.01397	-0.09701	0.07338	0.00720	0.00008
8	-C.00018	-0.00356	-0.01870	-0.12663	0.09073	0.01272	0.00167
9	-C.00038	-0.00431	-0.02447	-0.15733	0.10949	0.01692	0.00253
10	C.00035	-0.00427	-0.03058	-0.18722	0.12889	0.02221	0.00428
11	C.00090	-0.00528	-0.04426	-0.21174	0.15257	0.03332	0.00642
12	-C.00025	-0.01133	-0.05126	-0.20312	0.16744	0.04135	0.00958
13	-C.00525	-0.02130	-0.05952	-0.20379	0.17081	0.04242	0.01020
14	-C.00740	-0.02695	-0.06458	-0.19837	0.15982	0.04638	0.01221

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	C.00000	0.00001	0.00008	-0.00015	-0.00056	0.00005	0.00000
2	C.00001	-0.00004	-0.00265	-0.00189	-0.00694	0.00309	0.00003
3	C.00031	-0.00507	-0.00237	-0.00244	-0.01614	0.00353	0.00549
4	-C.01153	-0.00367	-0.00064	-0.00465	-0.02304	0.00159	0.01228
5	-C.01075	-0.00787	-0.00424	-0.00578	-0.03006	-0.00146	0.01001
6	-C.00814	-0.00552	-0.00391	-0.01063	-0.03845	-0.00509	0.00580
7	-C.00554	-0.00516	-0.00667	-0.01091	-0.04693	-0.01022	0.00103
8	-C.00308	-0.00444	-0.00798	-0.01385	-0.05715	-0.01513	-0.00440
9	C.00005	-0.00293	-0.00994	-0.01684	-0.06975	-0.02221	-0.01051
10	C.00287	-0.00278	-0.01278	-0.02021	-0.08287	-0.02927	-0.01695
11	C.00377	-0.00628	-0.02034	-0.02444	-0.09835	-0.04371	-0.03086
12	-0.00392	-0.01519	-0.02645	-0.02570	-0.10472	-0.05205	-0.04056
13	-C.01723	-0.02289	-0.02887	-0.02568	-0.10781	-0.05634	-0.04587
14	-C.02607	-0.02919	-0.03246	-0.05180	-0.13027	-0.06276	-0.05305

VELOCITY COMPONENTS ON BODY PANELS DUE TO WING PANEL PRESSURE SINGULARITIES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	C.	0.	0.	0.00170	-0.00117	0.	0.
2	C.	0.	0.00110	0.00478	-0.00328	-0.00074	0.
3	C.00036	0.00088	0.00307	0.00628	-0.00430	-0.00205	-0.00075
4	C.00159	0.00257	0.00473	0.00711	-0.00500	-0.00316	-0.00212
5	C.00316	0.00418	0.00570	0.00789	-0.00561	-0.00407	-0.00317
6	C.00452	0.00503	0.00659	0.00891	-0.00634	-0.00476	-0.00404
7	0.00533	0.00603	0.00767	0.01038	-0.00738	-0.00550	-0.00461
8	C.00611	0.00690	0.00907	0.01156	-0.00827	-0.00645	-0.00527
9	C.00714	0.00823	0.01017	0.01229	-0.00882	-0.00729	-0.00623
10	C.00844	0.00926	0.01100	0.01152	-0.00835	-0.00799	-0.00715
11	C.00992	0.01025	0.00733	0.00304	-0.00258	-0.00569	-0.00772
12	C.00596	0.00533	0.00416	0.00160	-0.00139	-0.00317	-0.00378
13	C.00370	0.00334	0.00240	0.00089	-0.00080	-0.00196	-0.00239
14	C.00247	0.00214	0.00152	0.00057	-0.00048	-0.00122	-0.00156

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000

ROW NO.							
1	0.	0.	0.	0.00139	-0.00212	0.	0.
2	0.	0.	0.00072	0.00882	-0.01030	-0.00105	0.
3	0.00034	0.00087	0.00314	0.02326	-0.02336	-0.00389	-0.00063
4	0.00085	0.00264	0.00770	0.03963	-0.03781	-0.00761	-0.00178
5	0.00133	0.00497	0.01256	0.05691	-0.05309	-0.01153	-0.00298
6	0.00205	0.00717	0.01776	0.07578	-0.06993	-0.01572	-0.00385
7	0.00267	0.00928	0.02347	0.09717	-0.08908	-0.02043	-0.00507
8	0.00348	0.01211	0.03051	0.12165	-0.11091	-0.02611	-0.00649
9	0.00456	0.01582	0.03826	0.14794	-0.13411	-0.03240	-0.00832
10	0.00565	0.01951	0.04644	0.17413	-0.15651	-0.03884	-0.01000
11	0.00761	0.02644	0.05942	0.19839	-0.17344	-0.04694	-0.01317
12	0.00824	0.02817	0.06333	0.20097	-0.17527	-0.04891	-0.01307
13	0.00895	0.03009	0.06558	0.20205	-0.17608	-0.05012	-0.01364
14	0.00932	0.03096	0.06652	0.19724	-0.16115	-0.04947	-0.01367

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
	-0.	-0.	-0.	-0.00340	-0.00156	-0.	-0.
2	-0.	-0.	-0.00185	-0.01155	-0.00385	-0.00085	-0.
3	-0.00045	-0.00129	-0.00551	-0.01800	-0.00290	-0.00211	-0.00055
4	-0.00178	-0.00365	-0.00864	-0.02141	0.00068	-0.00227	-0.00111
5	-0.00317	-0.00536	-0.01042	-0.02419	0.00546	-0.00123	-0.00050
6	-0.00385	-0.00584	-0.01168	-0.02707	0.01102	0.00061	0.00102
7	-0.00363	-0.00608	-0.01287	-0.03058	0.01730	0.00298	0.00323
8	-0.00318	-0.00607	-0.01421	-0.03349	0.02519	0.00566	0.00576
9	-0.00276	-0.00624	-0.01497	-0.03531	0.03466	0.00916	0.00860
10	-0.00218	-0.00580	-0.01505	-0.03372	0.04642	0.01348	0.01221
11	0.00136	-0.00207	-0.00484	-0.00740	0.07511	0.02801	0.02224
12	0.01181	0.01057	0.00779	0.00800	0.08973	0.04044	0.03484
13	0.01966	0.01879	0.01718	0.01835	0.09969	0.04931	0.04302
14	0.02726	0.02725	0.02820	0.05309	0.12921	0.05997	0.05170

VELOCITY COMPONENTS ON BODY PANELS DUE TO WING SOURCES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.	0.	0.	-0.00275	-0.00157	0.	0.
2	0.	0.	-0.00645	-0.01262	-0.01134	-0.00520	0.
3	0.	-0.00686	-0.00902	-0.00734	-0.00732	-0.00785	-0.00585
4	-0.01113	-0.00695	-0.00646	-0.00369	-0.00407	-0.00956	-0.01252
5	-0.01124	-0.01126	-0.00895	-0.00581	-0.00707	-0.00877	-0.01019
6	-0.00891	-0.00826	-0.00613	-0.00292	-0.00375	-0.00602	-0.00737
7	-0.00613	-0.00531	-0.00300	0.00021	-0.00054	-0.00297	-0.00441
8	-0.00328	-0.00224	0.00013	0.00300	0.00234	0.00008	-0.00138
9	-0.00025	0.00076	0.00290	0.00529	0.00473	0.00290	0.00167
10	0.00267	0.00351	0.00541	0.00753	0.00705	0.00542	0.00439
11	0.00632	0.00717	0.00881	0.00308	0.00388	0.00862	0.00773
12	0.00740	0.00597	0.00467	0.00365	0.00410	0.00548	0.00453

13	0.00208	0.00191	0.00159	0.00128	0.00131	0.00154	0.00169
14	0.00092	0.00084	0.00070	0.00059	0.00061	0.00071	0.00078

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.	0.	0.	-0.00015	-0.00106	0.	0.
2	0.	0.	-0.00215	-0.01927	-0.02232	-0.00479	0.
3	0.	-0.00460	-0.00837	-0.02398	-0.02413	-0.01141	-0.00699
4	-0.00260	-0.00750	-0.01101	-0.02218	-0.02128	-0.00715	-0.00056
5	-0.00052	0.00001	-0.00376	-0.01012	-0.00750	-0.00338	-0.00047
6	-0.00028	-0.00024	-0.00255	-0.00459	-0.00255	-0.00168	-0.00015
7	-0.00016	-0.00001	-0.00119	0.00100	0.00295	-0.00000	0.00020
8	-0.00004	0.00046	0.00043	0.00642	0.00817	0.00181	0.00071
9	0.00015	0.00109	0.00215	0.01104	0.01251	0.00342	0.00112
10	0.00032	0.00154	0.00361	0.01487	0.01607	0.00468	0.00134
11	0.00066	0.00205	0.00560	0.01433	0.01246	0.00665	0.00165
12	-0.00173	-0.00105	0.00067	0.00264	0.00152	-0.00122	-0.00010
13	0.00013	0.00054	0.00127	0.00194	0.00169	0.00090	0.00026
14	0.00015	0.00048	0.00086	0.00112	0.00099	0.00059	0.00019

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.	-0.	-0.	0.00415	-0.00212	-0.	-0.
2	-0.	-0.	0.00997	0.03008	-0.02056	-0.00664	-0.
3	-0.	0.01007	0.01697	0.02118	-0.01490	-0.01133	-0.00630
4	0.01365	0.01149	0.01425	0.01298	-0.00936	-0.01133	-0.01238
5	0.01435	0.01457	0.01285	0.00678	-0.00548	-0.00945	-0.01063
6	0.01206	0.01125	0.00839	0.00032	-0.00087	-0.00633	-0.00813
7	0.00897	0.00761	0.00357	-0.00594	0.00360	-0.00291	-0.00532
8	0.00555	0.00359	-0.00133	-0.01125	0.00744	0.00059	-0.00228
9	0.00174	-0.00049	-0.00573	-0.01520	0.01034	0.00380	0.00088
10	-0.00209	-0.00429	-0.00956	-0.01861	0.01287	0.00662	0.00377
11	-0.00716	-0.00946	-0.01471	-0.00827	0.00607	0.01034	0.00750
12	-0.00928	-0.00750	-0.00580	-0.00246	0.00212	0.00487	0.00492
13	-0.00328	-0.00306	-0.00239	-0.00094	0.00081	0.00182	0.00215
14	-0.00165	-0.00150	-0.00111	-0.00043	0.00035	0.00086	0.00106

PRESSURES, FORCES, AND MOMENTS ON ISOLATED BODY

CD = 0.00073 CL = 0.00103 CM = -0.00505

BODY PRESSURE COEFFICIENTS(CP)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
X							
0.	0.10782	0.10740	0.10663	0.10562	0.10454	0.10372	0.10329
0.7243	0.10782	0.10740	0.10663	0.10562	0.10454	0.10372	0.10329

1.4485	0.09793	0.09937	0.10196	0.10537	0.10898	0.11173	0.11315
2.1728	0.04349	0.04462	0.04666	0.04933	0.05216	0.05430	0.05541
2.8970	0.04274	0.04407	0.04649	0.04965	0.05298	0.05550	0.05680
3.6213	0.03006	0.03122	0.03330	0.03603	0.03891	0.04109	0.04221
4.3456	0.03069	0.03178	0.03376	0.03635	0.03907	0.04113	0.04219
5.0698	0.02365	0.02472	0.02629	0.02835	0.03052	0.03216	0.03301
5.7941	0.02418	0.02491	0.02622	0.02795	0.02977	0.03114	0.03185
6.5184	0.01805	0.01883	0.02022	0.02205	0.02397	0.02542	0.02617
7.2426	0.01617	0.01710	0.01878	0.02097	0.02327	0.02501	0.02590
7.9669	0.01282	0.01349	0.01470	0.01629	0.01796	0.01923	0.01989
8.6911	0.00881	0.00992	0.01193	0.01456	0.01731	0.01938	0.02044
9.4154	0.00597	0.00690	0.00859	0.01080	0.01312	0.01488	0.01578
10.1397	0.00311	0.00416	0.00605	0.00853	0.01113	0.01310	0.01410
10.8639	-0.00065	0.00030	0.00236	0.00505	0.00788	0.01001	0.01110
11.5882	-0.00306	-0.00212	-0.00044	0.00177	0.00410	0.00586	0.00676
12.3124	-0.00688	-0.00584	-0.00397	-0.00152	0.00105	0.00300	0.00400
13.0367	-0.01223	-0.01111	-0.00909	-0.00645	-0.00368	-0.00158	-0.00051
13.7610	-0.01478	-0.01362	-0.01152	-0.00878	-0.00589	-0.00372	-0.00260
14.5023	-0.02260	-0.02111	-0.01843	-0.01494	-0.01128	-0.00853	-0.00712
15.2436	-0.02807	-0.02666	-0.02412	-0.02079	-0.01730	-0.01466	-0.01330
15.9849	-0.04112	-0.04009	-0.03821	-0.03571	-0.03305	-0.03100	-0.02994
16.7262	-0.03321	-0.03262	-0.03154	-0.03007	-0.02845	-0.02718	-0.02651
17.4675	-0.02692	-0.02667	-0.02619	-0.02549	-0.02467	-0.02399	-0.02362
18.2088	-0.02203	-0.02202	-0.02196	-0.02160	-0.02155	-0.02130	-0.02114
18.9501	-0.01827	-0.01840	-0.01861	-0.01882	-0.01894	-0.01898	-0.01898
19.6914	-0.01538	-0.01560	-0.01597	-0.01638	-0.01674	-0.01695	-0.01704
20.4327	-0.01317	-0.01343	-0.01367	-0.01438	-0.01485	-0.01516	-0.01530
21.1740	-0.01147	-0.01173	-0.01218	-0.01272	-0.01323	-0.01357	-0.01374
21.9153	-0.01015	-0.01040	-0.01082	-0.01134	-0.01184	-0.01217	-0.01234
22.6566	-0.00911	-0.00933	-0.00972	-0.01018	-0.01063	-0.01094	-0.01109
23.3979	-0.00329	-0.00847	-0.00880	-0.00920	-0.00959	-0.00986	-0.00999
24.1392	-0.00761	-0.00777	-0.00804	-0.00837	-0.00869	-0.00891	-0.00902
24.8805	-0.00705	-0.00717	-0.00739	-0.00765	-0.00791	-0.00809	-0.00818
25.6218	-0.00657	-0.00666	-0.00683	-0.00704	-0.00724	-0.00738	-0.00745
26.3631	-0.00615	-0.00622	-0.00635	-0.00651	-0.00666	-0.00676	-0.00681
27.1044	-0.00577	-0.00583	-0.00593	-0.00604	-0.00616	-0.00623	-0.00627
27.8457	-0.00544	-0.00548	-0.00555	-0.00564	-0.00572	-0.00577	-0.00580
28.5870	-0.00514	-0.00517	-0.00522	-0.00528	-0.00534	-0.00538	-0.00539
29.3233	-0.00486	-0.00489	-0.00492	-0.00497	-0.00501	-0.00503	-0.00504
30.0696	-0.00710	-0.00711	-0.00712	-0.00714	-0.00715	-0.00715	-0.00715
30.8109	-0.01481	-0.01482	-0.01482	-0.01481	-0.01478	-0.01474	-0.01472
31.5522	-0.02245	-0.02249	-0.02257	-0.02264	-0.02268	-0.02269	-0.02269
32.2935	-0.02692	-0.02706	-0.02730	-0.02757	-0.02782	-0.02798	-0.02805
33.0348	-0.03016	-0.03043	-0.03088	-0.03142	-0.03194	-0.03229	-0.03245
33.7761	-0.03636	-0.03675	-0.03743	-0.03826	-0.03906	-0.03961	-0.03987
34.5174	-0.03784	-0.03840	-0.03939	-0.04061	-0.04181	-0.04265	-0.04306
35.2587	-0.02678	-0.02754	-0.02887	-0.03055	-0.03223	-0.03344	-0.03404
36.0000	-0.00855	-0.00940	-0.01092	-0.01285	-0.01483	-0.01627	-0.01700

INCREMENTAL PRESSURES, FORCES, AND MOMENTS ON BODY PANELS DUE TO WING

CD = 0.00030 CL = 0.00817 CM = -0.19262

BCDY PANEL PRESSURE CCEFFICIENT(CP)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-C.00001	-0.00002	-0.00009	0.00111	0.00650	0.00009	0.00003
2	C.00005	-0.00002	0.00943	0.02674	0.04535	0.01163	0.00024
3	-C.00058	0.00807	0.01575	0.00093	0.04402	0.02594	0.00972
4	C.00559	0.01059	0.00365	-0.01669	0.03488	0.03088	0.02403
5	0.01052	0.00309	-0.00626	-0.03003	0.02839	0.03142	0.03220
6	C.00707	-0.00146	-0.01762	-0.04179	0.02741	0.02913	0.03436
7	C.00032	-0.00823	-0.02575	-0.05274	0.02520	0.02900	0.03253
8	-C.00867	-0.01641	-0.03457	-0.06195	0.02662	0.02712	0.02891
9	-C.01779	-0.02571	-0.04319	-0.06769	0.02713	0.02586	0.02580
10	-C.02603	-0.03439	-C.05088	-0.06960	0.02365	0.02482	0.02507
11	-C.04142	-0.04695	-0.04631	-0.02290	0.01490	0.01412	0.02316
12	-C.04134	-0.03571	-0.02380	-0.00321	0.00882	0.01090	0.01671
13	-C.02468	-0.02117	-0.01450	-0.00494	0.00389	0.00970	0.01389
14	-C.01395	-0.01217	-0.00835	-0.00291	0.00215	0.00593	0.00781

BCDY PANEL SLCPE

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-C.01059	-0.00420	0.00737	0.02253	0.03871	0.05119	0.05757
2	-C.02789	-0.02151	-0.00993	0.00530	0.02150	0.03388	0.04026
3	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
4	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
5	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
6	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
7	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
8	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
9	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
10	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
11	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
12	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
13	-C.03408	-0.02769	-0.01612	-0.00086	0.01535	0.02769	0.03408
14	-C.04830	-0.04191	-0.03033	-0.01501	0.00122	0.01348	0.01986

PRESSURES, FORCES, AND MOMENTS ON WING PANELS IN PRESENCE OF BODY

CD = 0.00336 CL = 0.09979 CM = -3.08481

SPANWISE CD DISTRIBUTION

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
	C.00036	0.00051	0.00046	0.00043	0.00039	0.00035	0.00033	0.00033	0.00017	0.00004

SPANWISE CL DISTRIBUTION

	1	2	3	4	5	6	7	8	9	10

	0.00444	0.00843	0.00928	0.01001	0.01059	0.01108	0.01151	0.01186	0.00606	0.01653

UPPER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.03034	-0.01557	-0.03228	-0.05088	-0.06645	-0.08703	-0.10415	-0.12107	-0.12429	-0.12819
2	-0.04521	-0.06520	-0.08337	-0.09573	-0.11752	-0.12954	-0.14328	-0.14489	-0.14800	-0.14894
3	-0.05215	-0.07077	-0.08963	-0.10196	-0.11896	-0.13158	-0.13753	-0.13376	-0.13386	-0.13651
4	-0.05926	-0.07898	-0.09872	-0.10919	-0.12522	-0.13658	-0.12849	-0.12556	-0.12834	-0.12566
5	-0.06900	-0.08858	-0.10759	-0.11882	-0.12659	-0.12426	-0.11793	-0.12536	-0.12123	-0.13318
6	-0.07689	-0.10091	-0.11505	-0.12609	-0.11975	-0.11033	-0.11372	-0.11385	-0.11472	-0.11200
7	-0.09485	-0.10568	-0.11964	-0.11283	-0.10918	-0.10530	-0.10508	-0.10654	-0.10453	-0.10346
8	-0.09635	-0.10786	-0.10983	-0.09651	-0.09117	-0.09274	-0.08939	-0.09121	-0.09152	-0.09154
9	-0.10108	-0.10584	-0.08318	-0.07355	-0.07631	-0.07168	-0.07364	-0.07510	-0.07605	-0.07737
10	-0.09596	-0.05806	-0.04250	-0.04898	-0.04170	-0.04380	-0.04348	-0.04687	-0.04789	-0.05922

LOWER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.06476	0.04935	0.05384	0.05725	0.06563	0.06801	0.07240	0.07324	0.08026	0.10002
2	0.01514	0.01130	0.00791	0.01151	0.00370	0.00489	0.00111	0.00818	0.01222	0.01772
3	0.00581	-0.00066	-0.00455	-0.00297	-0.00889	-0.01312	-0.01343	-0.00459	0.00153	-0.00165
4	0.00337	-0.00401	-0.00849	-0.00686	-0.01490	-0.02216	-0.00995	-0.00397	-0.00331	-0.00300
5	0.00094	-0.00457	-0.01016	-0.01158	-0.01498	-0.01031	-0.00168	-0.00747	-0.00204	-0.01869
6	0.00460	-0.00624	-0.00900	-0.01478	-0.00649	0.00427	0.00168	0.00325	0.00141	0.00051
7	0.00254	0.00257	-0.00510	0.00318	0.00680	0.01112	0.01155	0.01118	0.01137	0.00891
8	0.01168	0.00623	0.00398	0.01655	0.02127	0.01943	0.02290	0.02125	0.01859	0.01537
9	0.01429	0.00412	0.02335	0.03144	0.02780	0.03203	0.03018	0.02800	0.02472	0.02184
10	0.00794	0.03368	0.04512	0.03775	0.04411	0.04164	0.04221	0.03761	0.03515	0.03378

WING PANEL PRESSURE DIFFERENCE(CL)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.02769	0.05820	0.07926	0.10147	0.12532	0.14905	0.17111	0.18987	0.19953	0.221[illegible]
2	0.06470	0.08193	0.09790	0.11458	0.13064	0.14486	0.15632	0.16473	0.17194	0.1780[illegible]
	0.06171	0.07495	0.09126	0.10609	0.11879	0.12850	0.13472	0.13908	0.14506	0.1449
	0.06525	0.07883	0.09560	0.10852	0.11814	0.12350	0.12648	0.12906	0.13273	0.130[illegible]
	0.07122	0.08659	0.10158	0.11238	0.11752	0.11949	0.12094	0.12342	0.12416	0.121[illegible]
	0.08054	0.09553	0.10808	0.11442	0.11553	0.11570	0.11686	0.11851	0.11766	0.113[illegible]
7	0.09426	0.10570	0.11327	0.11381	0.11332	0.11328	0.11345	0.11466	0.11266	0.1091
8	0.10134	0.10817	0.10796	0.10571	0.10454	0.10442	0.10418	0.10452	0.10228	0.0991
9	0.10455	0.09945	0.09376	0.09135	0.09075	0.08993	0.09021	0.08965	0.08749	0.0861[illegible]
10	0.08633	0.07119	0.06612	0.06572	0.06434	0.06413	0.06434	0.06341	0.06210	0.0724[illegible]

UPPER SURFACE WING PANEL SLOPE

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.11963	0.11963	0.11963	0.11963	0.11963	0.11963	0.11963	0.11963	0.11963	0.11963
2	0.03198	0.03198	0.03198	0.03198	0.03198	0.03198	0.03198	0.03198	0.03198	0.03198

3	0.00079	0.00079	0.00079	0.00079	0.00079	0.00079	0.00079	0.00079	0.00079	0.00079
4	-0.02312	-0.02312	-0.02312	-0.02312	-0.02312	-0.02312	-0.02312	-0.02312	-0.02312	-0.02312
5	-0.04378	-0.04378	-0.04378	-0.04378	-0.04378	-0.04378	-0.04378	-0.04378	-0.04378	-0.04378
6	-0.06470	-0.06470	-0.06470	-0.06470	-0.06470	-0.06470	-0.06470	-0.06470	-0.06470	-0.06470
7	-0.08727	-0.08727	-0.08727	-0.08727	-0.08727	-0.08727	-0.08727	-0.08727	-0.08727	-0.08727
8	-0.10525	-0.10525	-0.10525	-0.10525	-0.10525	-0.10525	-0.10525	-0.10525	-0.10525	-0.10525
9	-0.12518	-0.12518	-0.12518	-0.12518	-0.12518	-0.12518	-0.12518	-0.12518	-0.12518	-0.12518
10	-0.15075	-0.15075	-0.15075	-0.15075	-0.15075	-0.15075	-0.15075	-0.15075	-0.15075	-0.15075

LOWER SURFACE WING PANEL SLOPE

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
CHORDWISE STATION										
1	-0.09695	-0.09695	-0.09695	-0.09695	-0.09695	-0.09695	-0.09695	-0.09695	-0.09695	-0.09695
2	-0.06846	-0.06846	-0.06846	-0.06846	-0.06846	-0.06846	-0.06846	-0.06846	-0.06846	-0.06846
3	-0.04917	-0.04917	-0.04917	-0.04917	-0.04917	-0.04917	-0.04917	-0.04917	-0.04917	-0.04917
4	-0.03701	-0.03701	-0.03701	-0.03701	-0.03701	-0.03701	-0.03701	-0.03701	-0.03701	-0.03701
5	-0.02854	-0.02854	-0.02854	-0.02854	-0.02854	-0.02854	-0.02854	-0.02854	-0.02854	-0.02854
6	-0.02076	-0.02076	-0.02076	-0.02076	-0.02076	-0.02076	-0.02076	-0.02076	-0.02076	-0.02076
7	-0.02014	-0.02014	-0.02014	-0.02014	-0.02014	-0.02014	-0.02014	-0.02014	-0.02014	-0.02014
8	-0.02456	-0.02456	-0.02456	-0.02456	-0.02456	-0.02456	-0.02456	-0.02456	-0.02456	-0.02456
9	-0.03278	-0.03278	-0.03278	-0.03278	-0.03278	-0.03278	-0.03278	-0.03278	-0.03278	-0.03278
10	-0.04715	-0.04715	-0.04715	-0.04715	-0.04715	-0.04715	-0.04715	-0.04715	-0.04715	-0.04715

FORCES AND MOMENTS ON WING-BODY COMBINATION

CD = 0.00439 CL = 0.10898 CM = -3.28247

TR-805 GIVEN SHAPE

DELTA ANGLE OF ATTACK = 1.0000 DEG.

ANGLE OF ATTACK WITH RESPECT TO BODY AXIS = 3.0000 DEG.

VELOCITY COMPONENTS ON BODY DUE TO BODY LINE SOURCES AND DOUBLETS

AXIAL(U)

THETA(DEG.) X	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
0.	-0.05947	-0.06006	-0.06114	-0.06255	-0.06406	-0.06521	-0.06580
0.7243	-0.05947	-0.06006	-0.06114	-0.06255	-0.06406	-0.06521	-0.06580
1.4485	-0.05291	-0.05473	-0.05803	-0.06238	-0.06700	-0.07052	-0.07234
2.1728	-0.02250	-0.02404	-0.02683	-0.03051	-0.03442	-0.03740	-0.03894
2.8970	-0.02173	-0.02336	-0.02632	-0.03023	-0.03438	-0.03754	-0.03917
3.6213	-0.01463	-0.01611	-0.01880	-0.02234	-0.02610	-0.02896	-0.03044
4.3456	-0.01482	-0.01625	-0.01884	-0.02227	-0.02590	-0.02867	-0.03010
5.0698	-0.01106	-0.01233	-0.01463	-0.01766	-0.02087	-0.02332	-0.02459
5.7941	-0.01119	-0.01236	-0.01447	-0.01726	-0.02022	-0.02248	-0.02364
6.5184	-0.00791	-0.00906	-0.01114	-0.01388	-0.01679	-0.01901	-0.02015
7.2426	-0.00689	-0.00810	-0.01028	-0.01316	-0.01622	-0.01855	-0.01976
7.9669	-0.00512	-0.00615	-0.00802	-0.01049	-0.01311	-0.01510	-0.01613
8.6911	-0.00301	-0.00425	-0.00649	-0.00945	-0.01259	-0.01499	-0.01622
9.4154	-0.00154	-0.00265	-0.00467	-0.00733	-0.01015	-0.01230	-0.01341
10.1397	-0.00007	-0.00121	-0.00328	-0.00602	-0.00892	-0.01113	-0.01227
10.8639	0.00195	0.00079	-0.00131	-0.00407	-0.00701	-0.00925	-0.01041
11.5882	0.00303	0.00201	0.00018	-0.00224	-0.00481	-0.00677	-0.00778
12.3124	0.00491	0.00388	0.00203	-0.00042	-0.00302	-0.00500	-0.00602
13.0367	0.00753	0.00652	0.00468	0.00226	-0.00031	-0.00227	-0.00328
13.7610	0.00868	0.00769	0.00590	0.00353	0.00102	-0.00090	-0.00189
14.5023	0.01243	0.01135	0.00939	0.00680	0.00405	0.00196	0.00087
15.2436	0.01488	0.01392	0.01217	0.00986	0.00741	0.00555	0.00458
15.9849	0.02087	0.02024	0.01910	0.01759	0.01600	0.01478	0.01415
16.7262	0.01647	0.01611	0.01546	0.01461	0.01370	0.01301	0.01265
17.4675	0.01303	0.01287	0.01259	0.01221	0.01181	0.01151	0.01135
18.2088	0.01040	0.01038	0.01035	0.01030	0.01025	0.01022	0.01020
18.9501	0.00841	0.00848	0.00861	0.00877	0.00895	0.00908	0.00915
19.6914	0.00692	0.00704	0.00725	0.00754	0.00784	0.00807	0.00819
20.4327	0.00579	0.00594	0.00619	0.00653	0.00689	0.00717	0.00731
21.1740	0.00495	0.00509	0.00536	0.00571	0.00607	0.00636	0.00650
21.9153	0.00431	0.00444	0.00469	0.00502	0.00537	0.00563	0.00577
22.6566	0.00381	0.00393	0.00415	0.00445	0.00475	0.00499	0.00511
23.3979	0.00342	0.00353	0.00371	0.00396	0.00422	0.00442	0.00453
24.1392	0.00311	0.00319	0.00335	0.00355	0.00377	0.00393	0.00401
24.8805	0.00285	0.00291	0.00304	0.00320	0.00337	0.00350	0.00357
25.6218	0.00262	0.00267	0.00277	0.00290	0.00303	0.00313	0.00318
26.3631	0.00243	0.00247	0.00254	0.00263	0.00273	0.00281	0.00285
27.1044	0.00225	0.00228	0.00233	0.00240	0.00248	0.00253	0.00256
27.8457	0.00209	0.00211	0.00215	0.00220	0.00226	0.00230	0.00232
28.5870	0.00195	0.00196	0.00199	0.00203	0.00206	0.00209	0.00211
29.3283	0.00181	0.00182	0.00184	0.00187	0.00190	0.00192	0.00193
30.0696	0.00284	0.00286	0.00290	0.00296	0.00301	0.00306	0.00308
30.8109	0.00635	0.00643	0.00658	0.00677	0.00698	0.00714	0.00722

31.5522	C.00965	0.00984	0.01018	0.01063	0.01110	0.01147	0.01165
32.2935	0.01132	0.01164	0.01222	0.01299	0.01380	0.01442	0.01474
33.0348	C.01229	0.01277	0.01362	0.01475	0.01595	0.01686	0.01733
33.7761	C.01445	0.01512	0.01632	0.01790	0.01958	0.02086	0.02152
34.5174	C.01429	0.01515	0.01671	0.01876	0.02095	0.02261	0.02347
35.2587	C.00863	0.00960	0.01136	0.01368	0.01614	0.01801	0.01898
36.0000	C.00027	0.00120	0.00289	0.00511	0.00748	0.00928	0.01021

RADIAL(VR)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
X							
0.	C.15862	0.16051	0.16394	0.16847	0.17328	0.17694	0.17884
0.7243	C.15862	0.16051	0.16394	0.16847	0.17328	0.17694	0.17884
1.4485	C.14361	0.14632	0.15684	0.16809	0.18003	0.18912	0.19382
2.1728	C.08974	0.09509	0.10478	0.11757	0.13115	0.14148	0.14683
2.8970	C.06145	0.08755	0.09861	0.11320	0.12869	0.14048	0.14659
3.6213	C.06488	0.07127	0.08287	0.09816	0.11440	0.12676	0.13316
4.3456	C.06085	0.06751	0.07958	0.09549	0.11240	0.12526	0.13192
5.0698	C.05147	0.05815	0.07028	0.08626	0.10324	0.11616	0.12285
5.7941	C.04881	0.05551	0.06766	0.08367	0.10068	0.11362	0.12032
6.5184	C.04108	0.04792	0.06031	0.07665	0.09400	0.10721	0.11405
7.2426	C.03690	0.04399	0.05683	0.07377	0.09177	0.10546	0.11255
7.9669	C.03163	0.03862	0.05127	0.06795	0.08568	0.09917	0.10615
8.6911	C.02573	0.03321	0.04676	0.06463	0.08360	0.09805	0.10553
9.4154	C.02068	0.02817	0.04174	0.05964	0.07866	0.09313	0.10062
10.1397	C.01558	0.02332	0.03736	0.05587	0.07553	0.09050	0.09825
10.8639	C.00945	0.01745	0.03196	0.05108	0.07139	0.08686	0.09486
11.5882	C.00465	0.01266	0.02719	0.04635	0.06669	0.08218	0.09019
12.3124	-C.00153	0.00675	0.02174	0.04151	0.06251	0.07849	0.08676
13.0367	-C.00925	-0.00072	0.01474	0.03513	0.05679	0.07328	0.08181
13.7610	-C.01497	-0.00618	0.00975	0.03076	0.05307	0.07005	0.07884
14.5023	-C.02544	-0.01613	0.00074	0.02298	0.04660	0.06458	0.07389
15.2436	-0.03472	-0.02516	-0.00782	0.01503	0.03931	0.05778	0.06735
15.9849	-C.05112	-0.04154	-0.02417	-C.00128	0.02303	0.04154	0.05112
16.7262	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
17.4675	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
18.2088	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
18.9501	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
19.6914	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
20.4327	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
21.1740	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
21.9153	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
22.6566	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
23.3979	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
24.1392	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
24.8805	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
25.6218	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
26.3631	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
27.1044	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
27.8457	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
28.5870	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
29.3283	-C.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
30.0696	-C.05321	-0.04363	-0.02626	-0.00337	0.02094	0.03945	0.04903
30.8109	-C.06027	-0.05069	-0.03333	-0.01044	0.01388	0.03238	0.04196
31.5522	-C.06850	-0.05893	-0.04157	-0.01868	0.00562	0.02413	0.03370
32.2935	-C.07535	-0.06578	-0.04843	-0.02555	-0.00126	0.01724	0.02681

33.0348	-0.08196	-0.07240	-0.05506	-0.03220	-0.00792	0.01056	0.02012
33.7761	-0.09170	-0.08214	-0.06483	-0.04200	-0.01775	0.00071	0.01026
34.5174	-0.09876	-0.08923	-0.07194	-0.04914	-0.02493	-0.00650	0.00304
35.2587	-0.09591	-0.08638	-0.06909	-0.04630	-0.02209	-0.00367	0.00587
36.0000	-0.08513	-0.07558	-0.05828	-0.03546	-0.01122	0.00722	0.01677

TANGENTIAL(VT)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
X							
0.	-0.00166	-0.00466	-0.00679	-0.00766	-0.00688	-0.00466	-0.00166
0.7243	-0.00166	-0.00466	-0.00679	-0.00766	-0.00688	-0.00466	-0.00166
1.4485	-0.00345	-0.00971	-0.01415	-0.01595	-0.01433	-0.00971	-0.00345
2.1728	-0.00497	-0.01397	-0.02036	-0.02295	-0.02062	-0.01397	-0.00497
2.8970	-0.00581	-0.01635	-0.02382	-0.02684	-0.02411	-0.01635	-0.00581
3.6213	-0.00648	-0.01823	-0.02656	-0.02993	-0.02689	-0.01823	-0.00648
4.3456	-0.00688	-0.01934	-0.02819	-0.03177	-0.02854	-0.01934	-0.00688
5.0698	-0.00717	-0.02017	-0.02939	-0.03313	-0.02976	-0.02017	-0.00717
5.7941	-0.00731	-0.02056	-0.02996	-0.03376	-0.03033	-0.02056	-0.00731
6.5184	-0.00747	-0.02101	-0.03062	-0.03451	-0.03100	-0.02101	-0.00747
7.2426	-0.00766	-0.02155	-0.03140	-0.03539	-0.03179	-0.02155	-0.00766
7.9669	-0.00781	-0.02197	-0.03201	-0.03608	-0.03241	-0.02197	-0.00781
8.6911	-0.00805	-0.02264	-0.03299	-0.03718	-0.03340	-0.02264	-0.00805
9.4154	-0.00830	-0.02333	-0.03400	-0.03832	-0.03442	-0.02333	-0.00830
10.1397	-0.00855	-0.02405	-0.03504	-0.03949	-0.03548	-0.02405	-0.00855
10.8639	-0.00885	-0.02489	-0.03626	-0.04087	-0.03671	-0.02489	-0.00885
11.5882	-0.00911	-0.02562	-0.03733	-0.04207	-0.03779	-0.02562	-0.00911
12.3124	-0.00938	-0.02639	-0.03845	-0.04333	-0.03893	-0.02639	-0.00938
13.0367	-0.00971	-0.02730	-0.03977	-0.04483	-0.04027	-0.02730	-0.00971
13.7610	-0.01005	-0.02826	-0.04118	-0.04641	-0.04170	-0.02826	-0.01005
14.5023	-0.01052	-0.02959	-0.04311	-0.04859	-0.04365	-0.02959	-0.01052
15.2436	-0.01104	-0.03106	-0.04525	-0.05100	-0.04582	-0.03106	-0.01104
15.9849	-0.01160	-0.03263	-0.04754	-0.05358	-0.04813	-0.03263	-0.01160
16.7262	-0.01195	-0.03362	-0.04899	-0.05522	-0.04960	-0.03362	-0.01195
17.4675	-0.01215	-0.03416	-0.04978	-0.05610	-0.05040	-0.03416	-0.01215
18.2088	-0.01222	-0.03436	-0.05007	-0.05643	-0.05069	-0.03436	-0.01222
18.9501	-0.01221	-0.03433	-0.05002	-0.05637	-0.05064	-0.03433	-0.01221
19.6914	-0.01214	-0.03415	-0.04976	-0.05609	-0.05038	-0.03415	-0.01214
20.4327	-0.01205	-0.03390	-0.04940	-0.05567	-0.05001	-0.03390	-0.01205
21.1740	-0.01195	-0.03362	-0.04899	-0.05521	-0.04960	-0.03362	-0.01195
21.9153	-0.01185	-0.03334	-0.04858	-0.05475	-0.04918	-0.03334	-0.01185
22.6566	-0.01176	-0.03308	-0.04820	-0.05432	-0.04880	-0.03308	-0.01176
23.3979	-0.01168	-0.03285	-0.04787	-0.05395	-0.04847	-0.03285	-0.01168
24.1392	-0.01161	-0.03266	-0.04759	-0.05364	-0.04818	-0.03266	-0.01161
24.8805	-0.01156	-0.03251	-0.04736	-0.05338	-0.04795	-0.03251	-0.01156
25.6218	-0.01151	-0.03238	-0.04718	-0.05318	-0.04777	-0.03238	-0.01151
26.3631	-0.01148	-0.03229	-0.04704	-0.05302	-0.04763	-0.03229	-0.01148
27.1044	-0.01145	-0.03222	-0.04694	-0.05290	-0.04753	-0.03222	-0.01145
27.8457	-0.01144	-0.03216	-0.04686	-0.05282	-0.04745	-0.03216	-0.01144
28.5870	-0.01142	-0.03212	-0.04681	-0.05275	-0.04739	-0.03212	-0.01142
29.3283	-0.01141	-0.03210	-0.04677	-0.05271	-0.04735	-0.03210	-0.01141
30.0696	-0.01143	-0.03214	-0.04683	-0.05278	-0.04741	-0.03214	-0.01143
30.8109	-0.01151	-0.03238	-0.04718	-0.05317	-0.04777	-0.03238	-0.01151
31.5522	-0.01165	-0.03278	-0.04776	-0.05382	-0.04835	-0.03278	-0.01165
32.2935	-0.01181	-0.03321	-0.04839	-0.05454	-0.04899	-0.03321	-0.01181
33.0348	-0.01196	-0.03363	-0.04901	-0.05523	-0.04962	-0.03363	-0.01196
33.7761	-0.01213	-0.03411	-0.04970	-0.05602	-0.05032	-0.03411	-0.01213

34.5174	-0.01227	-0.03450	-0.05026	-0.05665	-0.05089	-0.03450	-0.01227
35.2587	-0.01225	-0.03445	-0.05020	-0.05658	-0.05083	-0.03445	-0.01225
36.0000	-0.01203	-0.03364	-0.04931	-0.05558	-0.04993	-0.03384	-0.01203

VELOCITY COMPONENTS ON WING PANELS DUE TO BODY PANEL PRESSURE SINGULARITIES

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.00561	-0.00134	0.00023	0.00381	0.00232	0.00203	0.00042	-0.00011	-0.00056	-0.00280
2	-0.00245	0.00028	0.00385	0.00153	0.00300	0.00071	-0.00002	-0.00133	-0.00078	-0.00165
3	0.00032	0.00271	0.00410	0.00072	0.00078	0.00003	-0.00056	-0.00094	-0.00235	-0.00079
4	0.00340	0.00528	0.00318	-0.00065	-0.00023	-0.00038	-0.00082	-0.00219	-0.00193	-0.00102
5	0.00743	0.00438	0.00218	-0.00041	-0.00297	-0.00080	-0.00323	0.00009	-0.00112	0.00652
6	0.00574	0.00472	0.00140	-0.00003	-0.00086	-0.00392	-0.00219	-0.00122	0.00014	0.00040
7	0.00784	0.00224	0.00044	-0.00190	-0.00097	-0.00219	-0.00113	0.00029	0.00041	0.00051
8	0.00436	0.00083	-0.00039	-0.00137	-0.00262	-0.00094	-0.00105	0.00040	0.00051	0.00029
9	0.00248	-0.00074	-0.00100	-0.00334	-0.00121	-0.00127	0.00049	0.00057	0.00024	0.00018
10	-0.00006	-0.00103	-0.00480	-0.00005	-0.00156	0.00068	0.00058	0.00024	0.00013	0.00015

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.00893	0.00348	0.00048	-0.00548	-0.00371	-0.00348	-0.00107	-0.00022	0.00054	0.00407
2	0.00717	0.00137	-0.00516	-0.00253	-0.00500	-0.00162	-0.00045	0.00165	0.00092	0.00244
3	0.00487	-0.00242	-0.00625	-0.00172	-0.00191	-0.00066	0.00039	0.00112	0.00341	0.00122
4	0.00005	-0.00763	-0.00557	-0.00008	-0.00053	-0.00005	0.00083	0.00315	0.00290	0.00160
5	-0.00842	-0.00769	-0.00487	-0.00068	0.00352	0.00063	0.00459	-0.00013	0.00175	-0.00964
6	-0.00963	-0.00991	-0.00424	-0.00137	0.00044	0.00543	0.00322	0.00189	-0.00008	-0.00052
7	-0.01641	-0.00779	-0.00316	0.00145	0.00068	0.00310	0.00174	-0.00029	-0.00052	-0.00070
8	-0.01571	-0.00688	-0.00201	0.00075	0.00354	0.00139	0.00171	-0.00050	-0.00069	-0.00039
9	-0.01664	-0.00525	-0.00103	0.00413	0.00167	0.00202	-0.00063	-0.00078	-0.00031	-0.00023
10	-0.01585	-0.00483	0.00514	-0.00035	0.00237	-0.00090	-0.00077	-0.00031	-0.00016	-0.00018

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.00519	0.00395	0.00269	0.00233	0.00242	0.00235	0.00230	0.00227	0.00211	0.00189
2	0.01805	0.00650	0.00354	0.00319	0.00315	0.00290	0.00271	0.00243	0.00238	0.00216
3	0.03420	0.00947	0.00534	0.00426	0.00375	0.00340	0.00295	0.00277	0.00249	0.00241
4	0.05241	0.01270	0.00745	0.00516	0.00443	0.00375	0.00334	0.00292	0.00283	0.00248
5	0.07191	0.01739	0.00911	0.00635	0.00484	0.00423	0.00347	0.00337	0.00292	0.00294
6	0.09415	0.02199	0.01102	0.00755	0.00563	0.00443	0.00400	0.00338	0.00314	0.00279
7	0.11996	0.02673	0.01285	0.00808	0.00633	0.00497	0.00421	0.00364	0.00321	0.00288
8	0.14540	0.03139	0.01400	0.00914	0.00662	0.00548	0.00431	0.00372	0.00327	0.00288
9	0.17167	0.03442	0.01558	0.00948	0.00733	0.00556	0.00460	0.00379	0.00317	0.00293
10	0.19117	0.03820	0.01620	0.01081	0.00750	0.00589	0.00470	0.00368	0.00317	0.00299

VELOCITY COMPONENTS ON WING PANELS DUE TO WING PANEL PRESSURE SINGULARITIES

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.01937	0.03106	0.03863	0.04610	0.05385	0.06135	0.06816	0.07384	0.07673	0.08313
2	0.02441	0.03020	0.03545	0.04079	0.04579	0.05011	0.05353	0.05602	0.05816	0.05995
3	0.02260	0.02645	0.03148	0.03603	0.03981	0.04262	0.04441	0.04567	0.04742	0.04734
4	0.02277	0.02650	0.03146	0.03525	0.03795	0.03940	0.04021	0.04091	0.04197	0.04117
5	0.02372	0.02794	0.03221	0.03517	0.03647	0.03692	0.03727	0.03795	0.03814	0.03721
6	0.02578	0.02980	0.03316	0.03470	0.03481	0.03475	0.03502	0.03546	0.03519	0.03406
7	0.02910	0.03194	0.03370	0.03352	0.03321	0.03309	0.03308	0.03341	0.03281	0.03177
8	0.03061	0.03195	0.03140	0.03049	0.03000	0.02988	0.02977	0.02985	0.02919	0.02826
9	0.03085	0.02673	0.02672	0.02581	0.02553	0.02524	0.02530	0.02512	0.02447	0.02402
10	0.02494	0.02014	0.01845	0.01821	0.01775	0.01766	0.01772	0.01743	0.01704	0.01973

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.05323	-0.08532	-0.10614	-0.12665	-0.14795	-0.16856	-0.18725	-0.20287	-0.21082	-0.22837
2	-0.06705	-0.08298	-0.09738	-0.11205	-0.12580	-0.13767	-0.14707	-0.15392	-0.15979	-0.16470
3	-0.06209	-0.07267	-0.08649	-0.09899	-0.10937	-0.11710	-0.12200	-0.12546	-0.13028	-0.13006
4	-0.06257	-0.07281	-0.08643	-0.09686	-0.10425	-0.10824	-0.11046	-0.11241	-0.11531	-0.11310
5	-0.06516	-0.07676	-0.08849	-0.09661	-0.10019	-0.10142	-0.10239	-0.10425	-0.10479	-0.10221
6	-0.07084	-0.08188	-0.09109	-0.09533	-0.09564	-0.09548	-0.09620	-0.09743	-0.09669	-0.09358
7	-0.07994	-0.08774	-0.09258	-0.09210	-0.09123	-0.09091	-0.09088	-0.09179	-0.09014	-0.08727
8	-0.08410	-0.08777	-0.08627	-0.08377	-0.08241	-0.08209	-0.08179	-0.08202	-0.08018	-0.07765
9	-0.08476	-0.07894	-0.07341	-0.07091	-0.07014	-0.06933	-0.06949	-0.06901	-0.06723	-0.06598
10	-0.06851	-0.05533	-0.05069	-0.05003	-0.04877	-0.04852	-0.04867	-0.04789	-0.04681	-0.05420

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.04167	-0.02751	-0.01843	-0.01439	-0.01248	-0.01123	-0.01043	-0.00991	-0.00950	-0.00895
2	-0.08860	-0.05992	-0.04871	-0.04463	-0.04263	-0.04124	-0.04034	-0.03961	-0.03931	-0.03880
3	-0.11163	-0.06882	-0.05621	-0.05145	-0.04903	-0.04760	-0.04648	-0.04586	-0.04535	-0.04501
4	-0.13614	-0.07770	-0.06392	-0.05803	-0.05547	-0.05375	-0.05270	-0.05186	-0.05154	-0.05096
5	-0.16167	-0.08815	-0.07142	-0.06516	-0.06188	-0.06027	-0.05889	-0.05838	-0.05772	-0.05754
6	-0.19013	-0.09895	-0.07968	-0.07279	-0.06916	-0.06699	-0.06596	-0.06495	-0.06449	-0.06396
7	-0.22643	-0.11432	-0.09229	-0.08419	-0.08078	-0.07847	-0.07713	-0.07617	-0.07552	-0.07507
8	-0.26260	-0.12990	-0.10450	-0.09637	-0.09222	-0.09015	-0.08841	-0.08744	-0.08682	-0.08635
9	-0.30252	-0.14678	-0.12004	-0.11074	-0.10698	-0.10430	-0.10277	-0.10164	-0.10086	-0.10053
10	-0.34162	-0.17035	-0.14055	-0.13199	-0.12709	-0.12458	-0.12285	-0.12156	-0.12086	-0.12059

VELOCITY COMPONENTS ON WING PANELS DUE TO BODY LINE SOURCES AND DOUBLETS

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.00742	0.01208	0.00795	0.00566	0.00419	0.00319	0.00251	0.00201	0.00173	0.00135
2	0.01488	0.00982	0.00681	0.00482	0.00364	0.00279	0.00222	0.00180	0.00156	0.00125
3	0.01182	0.00831	0.00574	0.00415	0.00314	0.00246	0.00197	0.00161	0.00141	0.00116
4	0.00993	0.00707	0.00488	0.00361	0.00275	0.00219	0.00176	0.00146	0.00128	0.00108
5	0.00840	0.00605	0.00421	0.00314	0.00244	0.00194	0.00159	0.00132	0.00116	0.00101
6	0.00715	0.00509	0.00367	0.00276	0.00217	0.00174	0.00144	0.00121	0.00107	0.00095

7	0.00598	0.00439	0.00320	0.00245	0.00194	0.00158	0.00131	0.00111	0.00099	0.00147
8	0.00512	0.00382	0.00281	0.00218	0.00175	0.00144	0.00120	0.00113	0.00205	0.00304
9	0.00443	0.00335	0.00250	0.00196	0.00158	0.00131	0.00110	0.00226	0.00378	0.00445
10	0.00386	0.00294	0.00223	0.00176	0.00144	0.00120	0.00154	0.00404	0.00510	0.00511

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.02543	-0.00038	-0.00270	-0.00285	-0.00249	-0.00205	-0.00171	-0.00141	-0.00123	-0.00098
2	0.01080	0.00076	-0.00210	-0.00220	-0.00203	-0.00169	-0.00143	-0.00120	-0.00106	-0.00087
3	0.01169	0.00119	-0.00141	-0.00172	-0.00160	-0.00140	-0.00120	-0.00103	-0.00092	-0.00078
4	0.01162	0.00155	-0.00088	-0.00135	-0.00129	-0.00118	-0.00102	-0.00089	-0.00080	-0.00070
5	0.01157	0.00185	-0.00051	-0.00101	-0.00105	-0.00097	-0.00087	-0.00077	-0.00070	-0.00063
6	0.01154	0.00227	-0.00023	-0.00075	-0.00085	-0.00081	-0.00074	-0.00067	-0.00061	-0.00056
7	0.01173	0.00249	0.00002	-0.00055	-0.00068	-0.00068	-0.00064	-0.00058	-0.00054	-0.00138
8	0.01174	0.00266	0.00023	-0.00039	-0.00054	-0.00057	-0.00055	-0.00068	-0.00221	-0.00383
9	0.01172	0.00279	0.00039	-0.00025	-0.00043	-0.00048	-0.00047	-0.00244	-0.00494	-0.00609
10	0.01171	0.00293	0.00051	-0.00013	-0.00034	-0.00040	-0.00121	-0.00528	-0.00716	-0.00722

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.03037	0.01746	0.00962	0.00595	0.00395	0.00276	0.00202	0.00153	0.00127	0.00095
2	0.03485	0.01773	0.00948	0.00574	0.00379	0.00265	0.00194	0.00148	0.00124	0.00095
3	0.03579	0.01771	0.00922	0.00554	0.00364	0.00255	0.00188	0.00145	0.00121	0.00096
4	0.03621	0.01748	0.00895	0.00535	0.00352	0.00248	0.00184	0.00142	0.00119	0.00097
5	0.03614	0.01715	0.00870	0.00519	0.00342	0.00242	0.00181	0.00140	0.00118	0.00098
6	0.03580	0.01677	0.00848	0.00506	0.00335	0.00238	0.00178	0.00138	0.00117	0.00099
7	0.03531	0.01644	0.00829	0.00496	0.00329	0.00235	0.00176	0.00137	0.00116	0.00103
8	0.03484	0.01616	0.00814	0.00488	0.00325	0.00232	0.00174	0.00137	0.00120	0.00111
9	0.03441	0.01593	0.00803	0.00482	0.00322	0.00230	0.00173	0.00141	0.00126	0.00117
10	0.03405	0.01575	0.00795	0.00476	0.00319	0.00229	0.00175	0.00148	0.00128	0.00120

VELOCITY COMPONENTS ON WING PANELS DUE TO WING SOURCES

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.03054	-0.02433	-0.01880	-0.01746	-0.01326	-0.00857	-0.00441	-0.00076	-0.00148	-0.00331
2	-0.00716	-0.00013	0.00337	0.00784	0.01289	0.01731	0.02137	0.02116	0.02006	0.01976
3	-0.00189	0.00375	0.00866	0.01475	0.01964	0.02390	0.02566	0.02337	0.02313	0.02314
4	-0.00060	0.00495	0.01303	0.01903	0.02368	0.02789	0.02409	0.02362	0.02363	0.02249
5	-0.00045	0.00850	0.01654	0.02209	0.02667	0.02357	0.02284	0.02278	0.02177	0.02115
6	0.00270	0.01161	0.01874	0.02411	0.02187	0.02069	0.02053	0.01941	0.01883	0.01855
7	0.00547	0.01297	0.01968	0.01900	0.01691	0.01658	0.01565	0.01475	0.01439	0.01431
8	0.00716	0.01399	0.01652	0.01237	0.01172	0.01114	0.00986	0.00933	0.00908	0.00913
9	0.00900	0.01548	0.00706	0.00593	0.00571	0.00393	0.00330	0.00293	0.00274	0.00307
10	0.01097	-0.00233	-0.00442	-0.00488	-0.00666	-0.00745	-0.00788	-0.00816	-0.00831	-0.00557

TRANSVERSE(V)

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10

CHORDWISE STATION										
1	0.02346	0.03708	0.03491	0.03503	0.02960	0.02265	0.01605	0.01001	0.01062	0.01346
2	-0.02849	-0.02351	-0.02451	-0.02979	-0.03719	-0.04422	-0.05097	-0.05142	-0.05010	-0.04997
3	-0.03063	-0.02966	-0.03436	-0.04316	-0.05102	-0.05823	-0.06183	-0.05896	-0.05886	-0.05917
4	-0.02439	-0.02819	-0.03974	-0.04951	-0.05761	-0.06514	-0.06024	-0.05998	-0.06022	-0.05877
5	-0.01502	-0.02915	-0.04278	-0.05283	-0.06136	-0.05790	-0.05738	-0.05766	-0.05636	-0.05561
6	-0.00956	-0.02868	-0.04278	-0.05337	-0.05185	-0.05088	-0.05112	-0.04977	-0.04905	-0.04882
7	-0.00345	-0.02458	-0.03956	-0.04162	-0.03967	-0.03983	-0.03886	-0.03773	-0.03731	-0.03742
8	0.00316	-0.02012	-0.03011	-0.02594	-0.02587	-0.02555	-0.02394	-0.02333	-0.02304	-0.02340
9	0.00822	-0.01726	-0.00930	-0.00904	-0.00947	-0.00721	-0.00649	-0.00607	-0.00586	-0.00692
10	0.01231	0.02030	0.02082	0.02039	0.02246	0.02335	0.02383	0.02413	0.02430	0.01669

VERTICAL(W)

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
CHORDWISE STATION										
1	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829
2	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022
3	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498
4	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695
5	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762
6	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197
7	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357
8	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034
9	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620
10	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180

VELOCITY COMPONENTS ON BODY PANELS DUE TO BODY PANEL PRESSURE SINGULARITIES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.00001	0.00003	0.00013	0.00144	-0.00139	-0.00012	-0.00004
2	-0.00004	0.00014	0.00101	-0.00165	-0.01139	-0.00030	-0.00022
3	-0.00030	0.00184	-0.00003	0.00428	-0.01357	-0.00469	0.00164
4	0.00630	-0.00013	0.00193	0.00862	-0.01185	-0.00411	0.00133
5	0.00328	0.00675	0.00859	0.01608	-0.00488	-0.00499	-0.00453
6	0.00199	0.00562	0.01070	0.01752	-0.00701	-0.00635	-0.00784
7	0.00240	0.00526	0.01044	0.01814	-0.00821	-0.00881	-0.00962
8	0.00362	0.00565	0.01036	0.01832	-0.01087	-0.01017	-0.01041
9	0.00433	0.00621	0.01083	0.01764	-0.01290	-0.01160	-0.01112
10	0.00433	0.00685	0.01120	0.01655	-0.01370	-0.01286	-0.01266
11	0.00723	0.00846	0.00835	0.00559	-0.01072	-0.01249	-0.01434
12	0.00993	0.00852	0.00390	-0.00373	-0.00812	-0.00948	-0.01126
13	0.00803	0.00648	0.00389	0.00038	-0.00304	-0.00549	-0.00750
14	0.00436	0.00374	0.00232	0.00032	-0.00151	-0.00301	-0.00378

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.00001	-0.00005	-0.00012	-0.00361	0.00293	0.00020	0.00004

2	-0.00017	-0.00010	-0.00133	-0.00131	0.02580	0.00036	0.00015
3	0.00047	-0.00118	-0.00055	-0.01922	0.04580	0.00297	0.00178
4	0.00091	-0.00020	-0.00203	-0.04489	0.06461	0.00051	-0.00321
5	-0.00244	-0.00642	-0.01228	-0.08089	0.07267	0.00198	-0.00154
6	-0.00165	-0.00720	-0.01770	-0.11131	0.09079	0.00760	-0.00142
7	-0.00150	-0.00546	-0.02191	-0.14406	0.11063	0.01283	0.00087
8	-0.00021	-0.00507	-0.02844	-0.17957	0.13299	0.01951	0.00277
9	-0.00058	-0.00662	-0.03602	-0.21580	0.15652	0.02496	0.00394
10	-0.00003	-0.00730	-0.04390	-0.25051	0.18020	0.03145	0.00594
11	0.00008	-0.00978	-0.06087	-0.27911	0.20840	0.04519	0.00878
12	-0.00155	-0.01740	-0.06956	-0.27101	0.22469	0.05459	0.01262
13	-0.00717	-0.02879	-0.07918	-0.27193	0.22912	0.05679	0.01370
14	-0.00982	-0.03565	-0.08538	-0.26502	0.21388	0.06157	0.01614

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.00000	0.00003	0.00023	-0.00043	-0.00183	0.00015	0.00001
2	0.00004	0.00007	-0.00198	-0.00305	-0.01227	0.00336	0.00003
3	0.00093	-0.00388	-0.00264	-0.00439	-0.02529	0.00229	0.00631
4	-0.00588	-0.00289	-0.00157	-0.00761	-0.03614	-0.00077	0.01233
5	-0.00914	-0.00764	-0.00633	-0.00955	-0.04675	-0.00535	0.00912
6	-0.00673	-0.00591	-0.00703	-0.01543	-0.05845	-0.01051	0.00378
7	-0.00474	-0.00622	-0.01066	-0.01656	-0.07017	-0.01725	-0.00237
8	-0.00301	-0.00637	-0.01300	-0.02029	-0.08369	-0.02392	-0.00916
9	-0.00055	-0.00565	-0.01587	-0.02413	-0.09936	-0.03273	-0.01660
10	0.00159	-0.00636	-0.01955	-0.02801	-0.11510	-0.04147	-0.02442
11	0.00072	-0.01167	-0.02852	-0.03264	-0.13309	-0.05875	-0.04130
12	-0.00948	-0.02263	-0.03577	-0.03420	-0.14042	-0.06919	-0.05371
13	-0.02459	-0.03150	-0.03857	-0.03402	-0.14393	-0.07488	-0.06090
14	-0.03532	-0.03925	-0.04344	-0.06939	-0.17421	-0.08341	-0.07028

VELOCITY COMPONENTS ON BODY PANELS DUE TO WING PANEL PRESSURE SINGULARITIES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.	0.	0.	0.00476	-0.00328	0.	0.
2	0.	0.	0.00309	0.00823	-0.00566	-0.00208	0.
3	0.00101	0.00246	0.00534	0.00969	-0.00665	-0.00356	-0.00210
4	0.00342	0.00459	0.00756	0.01041	-0.00738	-0.00521	-0.00374
5	0.00526	0.00674	0.00863	0.01099	-0.00785	-0.00617	-0.00505
6	0.00702	0.00760	0.00941	0.01185	-0.00848	-0.00686	-0.00608
7	0.00784	0.00868	0.01049	0.01328	-0.00948	-0.00757	-0.00659
8	0.00856	0.00949	0.01192	0.01438	-0.01034	-0.00852	-0.00720
9	0.00961	0.01089	0.01295	0.01491	-0.01076	-0.00934	-0.00819
10	0.01099	0.01189	0.01364	0.01370	-0.00998	-0.00995	-0.00914
11	0.01227	0.01254	0.00896	0.00368	-0.00314	-0.00697	-0.00942
12	0.00736	0.00657	0.00510	0.00195	-0.00170	-0.00391	-0.00467
13	0.00458	0.00414	0.00296	0.00110	-0.00098	-0.00242	-0.00296
14	0.00306	0.00265	0.00188	0.00071	-0.00059	-0.00151	-0.00194

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	C.	0.	0.	0.00390	-0.00594	0.	0.
2	C.	0.	0.00201	0.02046	-0.02253	-0.00294	0.
3	C.00095	0.00242	0.00667	0.04352	-0.04285	-0.00782	-0.00176
4	C.00140	0.00463	0.01381	0.06775	-0.06396	-0.01319	-0.00310
5	C.00224	0.00834	0.02091	0.09195	-0.08516	-0.01895	-0.00496
6	C.00329	0.01152	0.02824	0.11718	-0.10747	-0.02468	-0.00612
7	0.00411	0.01431	0.03584	C.14461	-0.13181	-0.03084	-0.00773
8	C.00518	0.01803	0.04494	0.17497	-0.15876	-0.03804	-0.00955
9	0.00656	0.02278	0.05461	0.20679	-0.18671	-0.04580	-0.01185
10	C.00792	0.02735	0.06455	0.23778	-0.21310	-0.05354	-0.01387
11	C.01025	0.03554	0.07990	0.26598	-0.23279	-0.06304	-0.01757
12	C.01106	0.03770	0.08455	0.26900	-0.23493	-0.06540	-0.01749
13	C.01189	0.03999	0.08722	0.27027	-0.23588	-0.06683	-0.01817
14	C.01235	0.04105	0.08839	0.26392	-0.21592	-0.06593	-0.01818

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-C.	-0.	-0.	-0.00950	-0.00436	-0.	-0.
2	-0.	-0.	-0.00517	-0.02205	-0.00608	-0.00239	-0.
3	-C.00125	-0.00361	-0.01001	-0.02982	-0.00306	-0.00343	-0.00155
4	-0.00371	-0.00639	-0.01416	-0.03321	0.00339	-0.00312	-0.00148
5	-C.00505	-0.00841	-0.01587	-0.03537	0.01141	-0.00081	-0.00004
6	-C.00560	-0.00847	-0.01671	-0.03759	0.02007	0.00245	0.00264
7	-C.00478	-0.00818	-0.01740	-0.04050	0.02935	0.00633	0.00621
8	-C.00363	-0.00751	-0.01819	-0.04261	0.04031	0.01050	0.01009
9	-C.00255	-0.00709	-0.01824	-0.04340	0.05291	0.01556	0.01422
10	-0.00132	-0.00591	-0.01748	-0.04024	0.06788	0.02149	0.01919
11	C.00401	-0.00022	-0.00409	-0.00811	0.10268	0.03981	0.03222
12	C.01703	0.01536	0.01149	0.01065	0.12052	0.05509	0.04768
13	C.02679	0.02555	0.02307	0.02336	0.13277	0.06606	0.05782
14	C.03638	0.03624	0.03711	0.06928	0.17171	0.07966	0.06881

VELOCITY COMPONENTS ON BODY PANELS DUE TO WING SOURCES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	C.	0.	0.	-0.00275	-0.00157	0.	0.
2	C.	0.	-0.00645	-0.01262	-0.01134	-0.00520	0.
3	C.	-0.00686	-0.00902	-0.00734	-0.00732	-0.00785	-0.00585
4	-C.01113	-0.00699	-0.00646	-0.00369	-0.00407	-0.00956	-0.01252
5	-0.01124	-0.01126	-0.00895	-0.00581	-0.00707	-0.00877	-0.01019
6	-C.00891	-0.00826	-0.00613	-0.00292	-0.00375	-0.00602	-0.00737
7	-C.00613	-0.00531	-0.00300	0.00021	-0.00054	-0.00297	-0.00441
	-C.00328	-0.00224	0.00013	0.00300	0.00234	0.00008	-0.00138

9	-C.00025	0.00076	0.00290	0.00529	0.00473	0.00290	0.00167
10	C.00267	0.00351	0.00541	0.00753	0.00705	0.00542	0.00439
11	C.00632	0.00717	C.00881	0.00308	0.00388	0.00862	0.00773
12	C.00740	0.00597	0.00467	0.00365	0.00410	0.00548	0.00453
13	C.00208	0.00191	0.00159	0.00128	0.00131	0.00154	0.00169
14	C.00092	0.00084	0.00070	0.00059	0.00061	0.00071	0.00078

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	C.	0.	0.	-0.00015	-0.00106	0.	0.
2	C.	0.	-0.00215	-0.01927	-0.02232	-0.00479	0.
3	C.	-0.00460	-0.00837	-0.02398	-0.02413	-0.01141	-0.00699
4	-C.00260	-0.00750	-0.01101	-0.02218	-0.02128	-0.00715	-0.00056
5	-C.00052	0.00001	-0.00376	-0.01012	-0.00750	-0.00338	-0.00047
6	-C.00028	-0.00024	-0.00255	-0.00459	-0.00255	-0.00168	-0.00015
7	-C.00016	-0.00001	-0.00119	0.00100	0.00295	-0.00000	0.00020
8	-C.00004	0.00046	0.00043	0.00642	0.00817	0.00181	0.00071
9	C.00015	0.00109	0.00215	0.01104	0.01251	0.00342	0.00112
10	C.00032	0.00154	0.00361	0.01487	0.01607	0.00468	0.00134
11	C.00066	0.00205	0.00560	0.01433	0.01246	0.00665	0.00165
12	-C.00173	-0.00105	0.00067	0.00264	0.00152	-0.00122	-0.00010
13	C.00013	0.00054	0.00127	0.00194	0.00169	0.00090	0.00026
14	C.00015	0.00048	0.00086	0.00112	0.00099	0.00059	0.00019

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-C.	-0.	-0.	0.00415	-0.00212	-0.	-0.
2	-C.	-0.	0.00997	0.03008	-0.02056	-0.00664	-0.
3	-C.	0.01007	0.01697	0.02118	-0.01490	-0.01133	-0.00630
4	C.01365	0.01149	0.01425	0.01298	-0.00936	-0.01133	-0.01238
5	C.01435	0.01457	0.01285	C.00678	-0.00548	-0.00945	-0.01063
6	C.01206	0.01125	0.00839	0.00032	-0.00087	-0.00633	-0.00813
7	C.00897	0.00761	0.00357	-0.00594	0.00360	-0.00291	-0.00532
8	C.00555	0.00359	-0.00133	-0.01125	0.00744	0.00059	-0.00228
9	C.00174	-0.00049	-0.00573	-0.01520	0.01034	0.00380	0.00088
10	-C.00209	-0.00429	-0.00956	-0.01861	0.01287	0.00662	0.00377
11	-C.00716	-0.00946	-0.01471	-0.00827	0.00607	0.01034	0.00750
12	-C.00928	-0.00750	-0.00580	-0.00246	0.00212	0.00487	0.00492
13	-C.00328	-0.00306	-0.00239	-0.00094	0.00081	0.00182	0.00215
14	-C.00165	-0.00150	-0.00111	-0.00043	0.00035	0.00086	0.00106

PRESSURES, FORCES, AND MOMENTS ON ISOLATED BODY

CD = 0.00072 CL = 0.00138 CM = -0.00479

BODY PRESSURE COEFFICIENTS(CP)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000

X							
0.	0.10169	0.10242	0.10372	0.10543	0.10724	0.10861	0.10931
0.7243	0.10169	0.10242	0.10372	0.10543	0.10724	0.10861	0.10931
1.4485	0.09146	0.09408	0.09881	0.10498	0.11145	0.11632	0.11883
2.1728	0.03805	0.04013	0.04388	0.04876	0.05388	0.05772	0.05969
2.8970	0.03784	0.04001	0.04391	0.04897	0.05426	0.05822	0.06026
3.6213	0.02549	0.02739	0.03081	0.03526	0.03991	0.04339	0.04518
4.3456	0.02637	0.02816	0.03136	0.03552	0.03986	0.04312	0.04479
5.0698	0.01970	0.02121	0.02393	0.02747	0.03118	0.03396	0.03539
5.7941	0.02023	0.02156	0.02394	0.02705	0.03031	0.03275	0.03401
6.5184	0.01422	0.01556	0.01798	0.02112	0.02442	0.02689	0.02816
7.2426	0.01247	0.01394	0.01658	0.02002	0.02360	0.02629	0.02766
7.9669	0.00924	0.01042	0.01254	0.01531	0.01821	0.02040	0.02152
8.6911	0.00531	0.00692	0.00980	0.01354	0.01744	0.02035	0.02184
9.4154	0.00258	0.00398	0.00648	0.00974	0.01316	0.01572	0.01703
10.1397	-0.00018	0.00130	0.00397	0.00743	0.01105	0.01377	0.01516
10.8639	-0.00406	-0.00250	0.00028	0.00390	0.00769	0.01052	0.01198
11.5882	-0.00613	-0.00463	-0.00249	0.00058	0.00380	0.00623	0.00748
12.3124	-0.00985	-0.00848	-0.00600	-0.00276	0.00064	0.00320	0.00451
13.0367	-0.01511	-0.01369	-0.01111	-0.00775	-0.00422	-0.00156	-0.00019
13.7610	-0.01752	-0.01609	-0.01351	-0.01013	-0.00659	-0.00391	-0.00254
14.5023	-0.02528	-0.02355	-0.02044	-0.01638	-0.01214	-0.00895	-0.00732
15.2436	-0.03060	-0.02900	-0.02611	-0.02233	-0.01834	-0.01532	-0.01377
15.9849	-0.04351	-0.04235	-0.04022	-0.03737	-0.03427	-0.03186	-0.03060
16.7262	-0.03508	-0.03449	-0.03337	-0.03178	-0.02997	-0.02849	-0.02770
17.4675	-0.02844	-0.02827	-0.02788	-0.02723	-0.02638	-0.02561	-0.02517
18.2088	-0.02332	-0.02343	-0.02355	-0.02355	-0.02337	-0.02310	-0.02292
18.9501	-0.01943	-0.01971	-0.02014	-0.02055	-0.02081	-0.02089	-0.02088
19.6914	-0.01649	-0.01685	-0.01745	-0.01810	-0.01862	-0.01889	-0.01900
20.4327	-0.01427	-0.01467	-0.01533	-0.01607	-0.01671	-0.01710	-0.01726
21.1740	-0.01260	-0.01299	-0.01364	-0.01439	-0.01506	-0.01548	-0.01566
21.9153	-0.01133	-0.01168	-0.01228	-0.01298	-0.01362	-0.01403	-0.01422
22.6566	-0.01034	-0.01065	-0.01118	-0.01180	-0.01237	-0.01274	-0.01292
23.3979	-0.00957	-0.00983	-0.01027	-0.01080	-0.01129	-0.01161	-0.01176
24.1392	-0.00894	-0.00916	-0.00952	-0.00995	-0.01035	-0.01062	-0.01074
24.8805	-0.00842	-0.00859	-0.00888	-0.00922	-0.00954	-0.00976	-0.00985
25.6218	-0.00798	-0.00811	-0.00833	-0.00860	-0.00885	-0.00901	-0.00909
26.3631	-0.00758	-0.00769	-0.00786	-0.00806	-0.00825	-0.00837	-0.00843
27.1044	-0.00723	-0.00731	-0.00744	-0.00759	-0.00773	-0.00782	-0.00786
27.8457	-0.00692	-0.00698	-0.00707	-0.00719	-0.00728	-0.00734	-0.00737
28.5870	-0.00663	-0.00667	-0.00674	-0.00683	-0.00690	-0.00694	-0.00695
29.3283	-0.00636	-0.00640	-0.00645	-0.00651	-0.00656	-0.00658	-0.00659
30.0696	-0.00862	-0.00864	-0.00867	-0.00869	-0.00869	-0.00868	-0.00867
30.8109	-0.01637	-0.01638	-0.01640	-0.01638	-0.01633	-0.01626	-0.01622
31.5522	-0.02392	-0.02400	-0.02413	-0.02424	-0.02430	-0.02429	-0.02427
32.2935	-0.02816	-0.02840	-0.02879	-0.02922	-0.02958	-0.02977	-0.02985
33.0348	-0.03111	-0.03154	-0.03226	-0.03310	-0.03385	-0.03433	-0.03454
33.7761	-0.03699	-0.03763	-0.03871	-0.03998	-0.04115	-0.04191	-0.04226
34.5174	-0.03803	-0.03894	-0.04049	-0.04236	-0.04412	-0.04530	-0.04586
35.2587	-0.02645	-0.02765	-0.02973	-0.03228	-0.03477	-0.03650	-0.03734
36.0000	-0.00793	-0.00926	-0.01159	-0.01451	-0.01745	-0.01956	-0.02061

INCREMENTAL PRESSURES, FORCES, AND MOMENTS ON BODY PANELS DUE TO WING

CD = 0.00058 CL = 0.01089 CM = -0.25309

BODY PANEL PRESSURE COEFFICIENT(CP)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.00002	-0.00006	-0.00025	-0.00688	0.01249	0.00024	0.00007
2	0.00008	-0.00029	0.00471	0.01215	0.05672	0.01521	0.00045
3	-0.00142	0.00511	0.00742	-0.01334	0.05445	0.03236	0.01264
4	0.00282	0.00506	-0.00604	-0.03093	0.04561	0.03793	0.03032
5	0.00540	-0.00445	-0.01650	-0.04296	0.03841	0.04010	0.04037
6	-0.00020	-0.00988	-0.02783	-0.05411	0.03739	0.03872	0.04354
7	-0.00818	-0.01722	-0.03588	-0.06500	0.03549	0.03903	0.04215
8	-0.01767	-0.02571	-0.04504	-0.07406	0.03696	0.03752	0.03876
9	-0.02597	-0.03546	-0.05379	-0.07933	0.03704	0.03631	0.03592
10	-0.03533	-0.04412	-0.06120	-0.07991	0.03242	0.03498	0.03546
11	-0.05027	-0.05579	-0.05355	-0.02676	0.01945	0.02175	0.03260
12	-0.04808	-0.04170	-0.02807	-0.00440	0.01111	0.01572	0.02307
13	-0.02892	-0.02494	-0.01713	-0.00564	0.00530	0.01270	0.01771
14	-0.01655	-0.01439	-0.00983	-0.00323	0.00298	0.00763	0.00994

BODY PANEL SLOPE

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.02763	-0.01805	-0.00069	0.02210	0.04639	0.06503	0.07461
2	-0.04493	-0.03535	-0.01799	0.00487	0.02918	0.04772	0.05730
3	-0.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
4	-0.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
5	-0.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
6	-0.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
7	-0.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
8	-0.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
9	-0.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
10	-0.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
11	-0.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
12	-0.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
13	-0.05112	-0.04154	-0.02418	-0.00128	0.02303	0.04154	0.05112
14	-0.06534	-0.05576	-0.03839	-0.01544	0.00890	0.02732	0.03690

PRESSURES, FORCES, AND MOMENTS ON WING PANELS IN PRESENCE OF BODY

CD = 0.00594 CL = 0.12870 CM = -3.94493

SPANWISE CD DISTRIBUTION

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
	0.00051	0.00078	0.00074	0.00072	0.00068	0.00064	0.00061	0.00061	0.00031	0.00034

SPANWISE CL DISTRIBUTION

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
	0.00588	0.01102	0.01202	0.01289	0.01357	0.01413	0.01463	0.01504	0.00768	0.02182

UPPER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	C.00847	-0.04673	-0.07012	-0.09339	-0.11520	-0.14183	-0.16411	-0.18565	-0.19056	-0.19828
2	-0.06361	-0.08783	-0.11036	-0.12488	-0.15019	-0.16507	-0.18106	-0.18399	-0.18834	-0.19015
3	-C.06935	-0.08963	-0.11116	-0.12585	-0.14490	-0.15908	-0.16586	-0.16279	-0.16346	-0.16638
4	-C.07549	-0.09641	-0.11814	-0.13064	-0.14753	-0.15962	-0.15179	-0.14908	-0.15217	-0.14922
5	-C.08445	-0.10505	-0.12589	-0.13851	-0.14549	-0.14399	-0.13738	-0.14589	-0.14124	-0.15407
6	-C.09181	-0.11730	-0.13253	-0.14408	-0.13708	-0.12704	-0.13061	-0.13105	-0.13184	-0.12874
7	-C.11024	-0.12193	-0.13603	-0.12844	-0.12442	-0.12017	-0.11997	-0.12158	-0.11939	-0.11798
8	-C.11176	-0.12307	-0.12404	-0.11010	-0.10409	-0.10564	-0.10199	-0.10412	-0.10425	-0.10388
9	-C.11559	-0.11882	-0.09505	-0.08479	-0.08748	-0.08247	-0.08463	-0.C8610	-0.08676	-0.08780
10	-0.10781	-0.06822	-0.05151	-0.05879	-0.05057	-0.05286	-0.05260	-0.05585	-0.05667	-0.06831

LCWER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	C.08830	0.07863	0.08544	0.09166	0.10081	0.10279	0.10677	0.10631	0.11394	0.13512
2	0.02569	0.02312	0.01980	0.02591	0.01778	0.01886	0.01459	0.02208	0.02628	0.03208
3	C.01482	0.00852	0.00539	0.00779	0.00178	-0.00283	-0.00317	0.00587	0.01263	0.00885
4	0.01169	0.00416	0.00044	0.00211	-0.00598	-0.01384	-0.00130	0.00486	0.00576	0.00580
5	C.00879	0.00361	-0.00203	-0.00404	-0.00653	-0.00286	0.00625	-0.00069	0.00550	-0.01317
6	C.01301	0.00149	-0.00169	-0.00833	0.00024	0.01150	0.00854	0.00993	0.00790	0.00672
7	C.01067	0.00976	0.00127	0.00934	0.01265	0.01701	0.01721	0.01678	0.01677	0.01401
8	C.01941	0.01267	0.00953	0.02155	0.02628	0.02406	0.02771	0.02568	0.02274	0.01931
9	0.02124	0.00927	0.02760	0.03522	0.03107	0.03540	0.03327	0.03090	0.02748	0.02443
10	C.01276	0.03654	0.04757	0.03869	0.04563	0.04280	0.04330	0.03861	0.03608	0.03478

WING PANEL PRESSURE DIFFERENCE(CL)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	C.07749	0.12422	0.15453	0.18439	0.21540	0.24542	0.27263	0.29536	0.30694	0.33250
2	0.09762	0.12082	0.14178	0.16314	0.18315	0.20044	0.21413	0.22410	0.23264	0.23979
3	C.09041	0.10581	0.12593	0.14413	0.15924	0.17049	0.17762	0.18267	0.18968	0.18936
4	C.09110	0.10601	0.12584	0.14102	0.15178	0.15759	0.16083	0.16366	0.16789	0.16467
5	C.09488	0.11177	0.12884	0.14067	0.14587	0.14767	0.14907	0.15179	0.15257	0.14882
6	C.10314	0.11922	0.13262	0.13880	0.13926	0.13902	0.14006	0.14186	0.14078	0.13625
7	C.11638	0.12774	0.13479	0.13410	0.13283	0.13235	0.13233	0.13364	0.13124	0.12707
8	0.12244	0.12779	0.12561	0.12197	0.11999	0.11952	0.11908	0.11941	0.11675	0.11306
9	C.12340	0.11494	0.10688	0.10325	0.10212	0.10095	0.10118	0.10048	0.09789	0.09607
10	C.09975	0.08056	0.07381	0.07284	0.07101	0.07065	0.07087	0.06973	0.06816	0.07893

UPPER SURFACE WING PANEL SLOPE

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10

CHORDWISE STATION										
1	0.10218	0.10218	0.10218	0.10218	0.10218	0.10218	0.10218	0.10218	0.10218	0.10218
2	0.01452	0.01452	0.01452	0.01452	0.01452	0.01452	0.01452	0.01452	0.01452	0.01452
3	-0.01666	-0.01666	-0.01666	-0.01666	-0.01666	-0.01666	-0.01666	-0.01666	-0.01666	-0.01666
4	-0.04057	-0.04057	-0.04057	-0.04057	-0.04057	-0.04057	-0.04057	-0.04057	-0.04057	-0.04057
5	-0.06124	-0.06124	-0.06124	-0.06124	-0.06124	-0.06124	-0.06124	-0.06124	-0.06124	-0.06124
6	-0.08216	-0.08216	-0.08216	-0.08216	-0.08216	-0.08216	-0.08216	-0.08216	-0.08216	-0.08216
7	-0.10472	-0.10472	-0.10472	-0.10472	-0.10472	-0.10472	-0.10472	-0.10472	-0.10472	-0.10472
8	-0.12270	-0.12270	-0.12270	-0.12270	-0.12270	-0.12270	-0.12270	-0.12270	-0.12270	-0.12270
9	-0.14263	-0.14263	-0.14263	-0.14263	-0.14263	-0.14263	-0.14263	-0.14263	-0.14263	-0.14263
10	-0.16820	-0.16820	-0.16820	-0.16820	-0.16820	-0.16820	-0.16820	-0.16820	-0.16820	-0.16820

LOWER SURFACE WING PANEL SLOPE

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
CHORDWISE STATION										
1	-0.11440	-0.11440	-0.11440	-0.11440	-0.11440	-0.11440	-0.11440	-0.11440	-0.11440	-0.11440
2	-0.08591	-0.08591	-0.08591	-0.08591	-0.08591	-0.08591	-0.08591	-0.08591	-0.08591	-0.08591
3	-0.06663	-0.06663	-0.06663	-0.06663	-0.06663	-0.06663	-0.06663	-0.06663	-0.06663	-0.06663
4	-0.05446	-0.05446	-0.05446	-0.05446	-0.05446	-0.05446	-0.05446	-0.05446	-0.05446	-0.05446
5	-0.04600	-0.04600	-0.04600	-0.04600	-0.04600	-0.04600	-0.04600	-0.04600	-0.04600	-0.04600
6	-0.03821	-0.03821	-0.03821	-0.03821	-0.03821	-0.03821	-0.03821	-0.03821	-0.03821	-0.03821
7	-0.03759	-0.03759	-0.03759	-0.03759	-0.03759	-0.03759	-0.03759	-0.03759	-0.03759	-0.03759
8	-0.04201	-0.04201	-0.04201	-0.04201	-0.04201	-0.04201	-0.04201	-0.04201	-0.04201	-0.04201
9	-0.05024	-0.05024	-0.05024	-0.05024	-0.05024	-0.05024	-0.05024	-0.05024	-0.05024	-0.05024
10	-0.06460	-0.06460	-0.06460	-0.06460	-0.06460	-0.06460	-0.06460	-0.06460	-0.06460	-0.06460

FORCES AND MOMENTS ON WING-BODY COMBINATION

CD = 0.00724 CL = 0.14097 CM = -4.20281

TR-305 GIVEN SHAPE

DELTA ANGLE OF ATTACK = 1.0000 DEG.

ANGLE OF ATTACK WITH RESPECT TO BODY AXIS = 4.0000 DEG.

VELOCITY COMPONENTS ON BODY DUE TO BODY LINE SOURCES AND DOUBLETS

AXIAL(U)

THETA(DEG.) X	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
0.	-0.05442	-0.05596	-0.05875	-0.06243	-0.06634	-0.06931	-0.07085
0.7243	-0.05442	-0.05596	-0.05875	-0.06243	-0.06634	-0.06931	-0.07085
1.4485	-0.04786	-0.05063	-0.05564	-0.06226	-0.06928	-0.07463	-0.07739
2.1728	-0.01839	-0.02070	-0.02489	-0.03041	-0.03627	-0.04074	-0.04305
2.8970	-0.01800	-0.02033	-0.02456	-0.03014	-0.03606	-0.04056	-0.04290
3.6213	-0.01126	-0.01338	-0.01720	-0.02225	-0.02761	-0.03169	-0.03381
4.3456	-0.01163	-0.01366	-0.01734	-0.02219	-0.02734	-0.03126	-0.03328
5.0698	-0.00809	-0.00991	-0.01322	-0.01758	-0.02221	-0.02574	-0.02757
5.7941	-0.00836	-0.01005	-0.01313	-0.01719	-0.02150	-0.02478	-0.02648
6.5184	-0.00523	-0.00688	-0.00987	-0.01381	-0.01800	-0.02118	-0.02283
7.2426	-0.00434	-0.00602	-0.00907	-0.01310	-0.01737	-0.02062	-0.02231
7.9669	-0.00271	-0.00420	-0.00689	-0.01043	-0.01419	-0.01706	-0.01854
8.6911	-0.00072	-0.00239	-0.00541	-0.00939	-0.01362	-0.01684	-0.01851
9.4154	0.00061	-0.00090	-0.00365	-0.00727	-0.01112	-0.01405	-0.01556
10.1397	0.00195	0.00043	-0.00233	-0.00597	-0.00983	-0.01277	-0.01429
10.8639	0.00383	0.00232	-0.00042	-0.00403	-0.00786	-0.01078	-0.01229
11.5882	0.00477	0.00343	0.00100	-0.00220	-0.00560	-0.00818	-0.00952
12.3124	0.00650	0.00518	0.00278	-0.00038	-0.00374	-0.00629	-0.00762
13.0367	0.00895	0.00767	0.00535	0.00229	-0.00095	-0.00343	-0.00471
13.7610	0.00994	0.00871	0.00649	0.00356	0.00045	-0.00191	-0.00314
14.5023	0.01348	0.01220	0.00988	0.00683	0.00358	0.00111	-0.00017
15.2436	0.01569	0.01458	0.01255	0.00988	0.00705	0.00489	0.00377
15.9849	0.02132	0.02060	0.01931	0.01761	0.01579	0.01442	0.01370
16.7262	0.01664	0.01625	0.01555	0.01461	0.01362	0.01286	0.01247
17.4675	0.01302	0.01287	0.01258	0.01221	0.01181	0.01151	0.01136
18.2088	0.01028	0.01028	0.01029	0.01030	0.01031	0.01032	0.01032
18.9501	0.00822	0.00833	0.00852	0.00877	0.00904	0.00924	0.00934
19.6914	0.00670	0.00686	0.00715	0.00753	0.00794	0.00825	0.00841
20.4327	0.00558	0.00576	0.00609	0.00653	0.00699	0.00734	0.00753
21.1740	0.00475	0.00493	0.00526	0.00570	0.00617	0.00652	0.00670
21.9153	0.00413	0.00430	0.00461	0.00501	0.00545	0.00578	0.00595
22.6566	0.00366	0.00381	0.00408	0.00444	0.00482	0.00511	0.00526
23.3979	0.00330	0.00343	0.00366	0.00396	0.00428	0.00452	0.00465
24.1392	0.00301	0.00311	0.00330	0.00355	0.00381	0.00401	0.00411
24.8805	0.00277	0.00285	0.00300	0.00320	0.00340	0.00356	0.00364
25.6218	0.00256	0.00263	0.00274	0.00289	0.00305	0.00318	0.00324
26.3631	0.00238	0.00243	0.00252	0.00263	0.00275	0.00284	0.00289
27.1044	0.00222	0.00225	0.00232	0.00240	0.00249	0.00256	0.00260
27.8457	0.00207	0.00209	0.00214	0.00220	0.00227	0.00232	0.00234
28.5870	0.00193	0.00195	0.00198	0.00203	0.00207	0.00211	0.00213
29.3283	0.00180	0.00181	0.00184	0.00187	0.00190	0.00193	0.00194
30.0696	0.00280	0.00283	0.00288	0.00295	0.00303	0.00308	0.00311
30.8109	0.00621	0.00631	0.00651	0.00677	0.00704	0.00725	0.00736

31.5522	C.00932	0.00957	0.01002	0.01062	0.01125	0.01174	0.01199
32.2935	C.01075	0.01117	0.01195	0.01297	0.01406	0.01488	0.01531
33.0348	0.01146	0.01208	0.01323	0.01473	0.01633	0.01754	0.01817
33.7761	C.01328	0.01416	0.01576	0.01787	0.02011	0.02182	0.02270
34.5174	0.01276	0.01391	0.01599	0.01872	0.02163	0.02385	0.02499
35.2587	C.00691	0.00820	0.01055	0.01363	0.01692	0.01941	0.02071
36.0000	-C.00139	-0.00015	0.00211	0.00507	0.00822	0.01062	0.01186

RADIAL(VR)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
X							
0.	C.14248	0.14740	0.15631	0.16807	0.18055	0.19005	0.19497
0.7243	C.14248	0.14740	0.15631	0.16807	0.18055	0.19005	0.19497
1.4485	C.12748	0.13521	0.14921	0.16768	0.18729	0.20222	0.20995
2.1728	C.07320	0.08165	0.09696	0.11715	0.13860	0.15492	0.16337
2.8970	0.06485	0.07406	0.09076	0.11278	0.13617	0.15397	0.16319
3.6213	C.04818	0.05770	0.07497	0.09774	0.12192	0.14033	0.14986
4.3456	0.04412	0.05391	0.07167	0.09507	0.11993	0.13885	0.14865
5.0698	C.03469	0.04452	0.06234	0.08584	0.11080	0.12979	0.13962
5.7941	C.03202	0.04186	0.05971	0.08325	0.10824	0.12727	0.13712
6.5184	C.02425	0.03424	0.05235	0.07623	0.10159	0.12089	0.13088
7.2426	C.02005	0.03030	0.04887	0.07335	0.09936	0.11915	0.12940
7.9669	C.01476	0.02491	0.04329	0.06753	0.09328	0.11288	0.12302
8.6911	C.00884	0.01948	0.03877	0.06420	0.09121	0.11177	0.12241
9.4154	C.00377	0.01443	0.03375	0.05922	0.08627	0.10687	0.11753
10.1397	-C.00134	0.00957	0.02936	0.05545	0.08316	0.10425	0.11517
10.8639	-C.00749	0.00368	0.02394	0.05066	0.07903	0.10062	0.11180
11.5882	-C.01231	-0.00112	0.01917	0.04592	0.07433	0.09596	0.10715
12.3124	-C.01850	-0.00705	0.01371	0.04108	0.07016	0.09228	0.10374
13.0367	-C.02624	-0.01453	0.00671	0.03471	0.06444	0.08708	0.09880
13.7610	-C.03197	-0.02000	0.00171	0.03033	0.06073	0.08387	0.09584
14.5023	-C.04245	-0.02996	-0.00731	0.02255	0.05427	0.07841	0.09090
15.2436	-C.05175	-0.03899	-0.01588	0.01460	0.04698	0.07162	0.08437
15.9849	-C.06815	-0.05538	-0.03223	-0.00171	0.03071	0.05539	0.06816
16.7262	-C.06816	-0.05539	-0.03223	-0.00171	0.03071	0.05539	0.06816
17.4675	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
18.2088	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
18.9501	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
19.6914	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
20.4327	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
21.1740	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
21.9153	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
22.6566	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
23.3979	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
24.1392	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
24.8805	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
25.6218	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
26.3631	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
27.1044	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
27.8457	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
28.5870	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
29.3283	-C.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
30.0696	-C.07024	-0.05747	-0.03432	-0.00380	0.02862	0.05330	0.06607
30.8109	-C.07731	-0.06454	-0.04139	-0.01087	0.02155	0.04623	0.05900
31.5522	-C.08554	-0.07277	-0.04963	-0.01911	0.01330	0.03797	0.05073
32.2935	-C.09237	-0.07961	-0.05648	-0.02598	0.00641	0.03107	0.04383

33.0348	-0.09898	-0.08622	-0.06311	-0.03263	-0.00026	0.02438	0.03713
33.7761	-0.10869	-0.09595	-0.07287	-0.04243	-0.01010	0.01451	0.02725
34.5174	-0.11573	-0.10301	-0.07996	-0.04957	-0.01728	0.00729	0.02001
35.2587	-0.11288	-0.10016	-0.07711	-0.04673	-0.01445	0.01012	0.02283
36.0000	-0.10211	-0.08938	-0.06631	-0.03588	-0.00357	0.02103	0.03376

TANGENTIAL(VT)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
X							
0.	-0.00430	-0.01210	-0.01763	-0.01987	-0.01785	-0.01210	-0.00430
0.7243	-0.00430	-0.01210	-0.01763	-0.01987	-0.01785	-0.01210	-0.00430
1.4485	-0.00610	-0.01715	-0.02499	-0.02816	-0.02530	-0.01715	-0.00610
2.1728	-0.00804	-0.02261	-0.03295	-0.03714	-0.03336	-0.02261	-0.00804
2.8970	-0.00899	-0.02530	-0.03686	-0.04154	-0.03732	-0.02530	-0.00899
3.6213	-0.00978	-0.02750	-0.04007	-0.04516	-0.04057	-0.02750	-0.00978
4.3456	-0.01022	-0.02874	-0.04188	-0.04720	-0.04240	-0.02874	-0.01022
5.0698	-0.01057	-0.02974	-0.04333	-0.04883	-0.04387	-0.02974	-0.01057
5.7941	-0.01074	-0.03021	-0.04402	-0.04961	-0.04457	-0.03021	-0.01074
6.5184	-0.01094	-0.03078	-0.04485	-0.05055	-0.04541	-0.03078	-0.01094
7.2426	-0.01116	-0.03140	-0.04575	-0.05156	-0.04632	-0.03140	-0.01116
7.9669	-0.01135	-0.03192	-0.04651	-0.05242	-0.04709	-0.03192	-0.01135
8.6911	-0.01162	-0.03267	-0.04761	-0.05366	-0.04820	-0.03267	-0.01162
9.4154	-0.01190	-0.03346	-0.04875	-0.05494	-0.04936	-0.03346	-0.01190
10.1397	-0.01218	-0.03426	-0.04991	-0.05626	-0.05054	-0.03426	-0.01218
10.8639	-0.01251	-0.03518	-0.05126	-0.05777	-0.05190	-0.03518	-0.01251
11.5882	-0.01280	-0.03600	-0.05246	-0.05912	-0.05311	-0.03600	-0.01280
12.3124	-0.01311	-0.03686	-0.05371	-0.06054	-0.05438	-0.03686	-0.01311
13.0367	-0.01347	-0.03788	-0.05520	-0.06221	-0.05589	-0.03788	-0.01347
13.7610	-0.01385	-0.03895	-0.05675	-0.06396	-0.05746	-0.03895	-0.01385
14.5023	-0.01436	-0.04040	-0.05886	-0.06634	-0.05960	-0.04040	-0.01436
15.2436	-0.01494	-0.04201	-0.06121	-0.06899	-0.06197	-0.04201	-0.01494
15.9849	-0.01558	-0.04381	-0.06384	-0.07195	-0.06464	-0.04381	-0.01558
16.7262	-0.01598	-0.04493	-0.06547	-0.07379	-0.06629	-0.04493	-0.01598
17.4675	-0.01618	-0.04551	-0.06631	-0.07473	-0.06713	-0.04551	-0.01618
18.2088	-0.01624	-0.04569	-0.06657	-0.07503	-0.06740	-0.04569	-0.01624
18.9501	-0.01621	-0.04560	-0.06644	-0.07488	-0.06727	-0.04560	-0.01621
19.6914	-0.01612	-0.04535	-0.06608	-0.07447	-0.06690	-0.04535	-0.01612
20.4327	-0.01601	-0.04502	-0.06559	-0.07393	-0.06641	-0.04502	-0.01601
21.1740	-0.01588	-0.04466	-0.06507	-0.07333	-0.06588	-0.04466	-0.01588
21.9153	-0.01575	-0.04430	-0.06456	-0.07276	-0.06536	-0.04430	-0.01575
22.6566	-0.01564	-0.04398	-0.06409	-0.07223	-0.06489	-0.04398	-0.01564
23.3979	-0.01554	-0.04370	-0.06368	-0.07177	-0.06447	-0.04370	-0.01554
24.1392	-0.01546	-0.04347	-0.06334	-0.07139	-0.06413	-0.04347	-0.01546
24.8805	-0.01539	-0.04328	-0.06306	-0.07108	-0.06385	-0.04328	-0.01539
25.6218	-0.01533	-0.04313	-0.06285	-0.07083	-0.06363	-0.04313	-0.01533
26.3631	-0.01529	-0.04302	-0.06268	-0.07064	-0.06346	-0.04302	-0.01529
27.1044	-0.01526	-0.04293	-0.06255	-0.07050	-0.06333	-0.04293	-0.01526
27.8457	-0.01524	-0.04287	-0.06246	-0.07039	-0.06324	-0.04287	-0.01524
28.5870	-0.01522	-0.04282	-0.06239	-0.07032	-0.06317	-0.04282	-0.01522
29.3283	-0.01521	-0.04279	-0.06234	-0.07026	-0.06312	-0.04279	-0.01521
30.0696	-0.01523	-0.04285	-0.06243	-0.07036	-0.06321	-0.04285	-0.01523
30.8109	-0.01535	-0.04317	-0.06290	-0.07089	-0.06368	-0.04317	-0.01535
31.5522	-0.01554	-0.04370	-0.06367	-0.07176	-0.06446	-0.04370	-0.01554
32.2935	-0.01574	-0.04428	-0.06451	-0.07271	-0.06532	-0.04428	-0.01574
33.0348	-0.01594	-0.04484	-0.06534	-0.07364	-0.06615	-0.04484	-0.01594
33.7761	-0.01617	-0.04548	-0.06626	-0.07468	-0.06709	-0.04548	-0.01617

34.5174	-C.01635	-0.04599	-0.06701	-0.07553	-0.06785	-0.04599	-0.01635
35.2587	-C.01633	-0.04593	-0.06693	-0.07543	-0.06776	-0.04593	-0.01633
36.0000	-C.01604	-0.04512	-0.06574	-0.07410	-0.06656	-0.04512	-0.01604

VELOCITY COMPONENTS ON WING PANELS DUE TO BODY PANEL PRESSURE SINGULARITIES

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-C.00491	-0.00100	0.00058	0.00348	0.00212	0.00191	0.00033	-0.00007	-0.00064	-0.00294
2	-C.00186	0.00066	0.00439	0.00137	0.00286	0.00064	-0.00006	-0.00140	-0.00084	-0.00172
3	C.00094	0.00298	0.00416	0.00064	0.00066	-0.00006	-0.00067	-0.00096	-0.00248	-0.00078
4	0.00392	0.00545	0.00306	-0.00069	-0.00041	-0.00050	-0.00089	-0.00223	-0.00205	-0.00106
5	0.00779	0.00426	0.00201	-0.00041	-0.00346	-0.00088	-0.00340	0.00028	-0.00116	0.00701
6	0.00575	0.00458	0.00128	-0.00000	-0.00095	-0.00414	-0.00232	-0.00126	0.00013	0.00043
7	0.00776	0.00213	0.00036	-0.00201	-0.00103	-0.00228	-0.00116	0.00028	0.00044	0.00058
8	C.00429	0.00067	-0.00055	-0.00140	-0.00273	-0.00096	-0.00116	0.00043	0.00058	0.00035
9	0.00226	-0.00103	-0.00114	-0.00342	-0.00119	-0.00133	0.00053	0.00065	0.00029	0.00021
10	-C.00045	-0.00119	-0.00507	0.00021	-0.00163	0.00073	0.00065	0.00029	0.00015	0.00016

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	C.00793	0.00269	-0.00025	-0.00515	-0.00353	-0.00337	-0.00098	-0.00029	0.00065	0.00429
2	C.00584	0.00037	-0.00624	-0.00246	-0.00489	-0.00158	-0.00042	0.00174	0.00100	0.00255
3	C.00302	-0.00344	-0.00666	-0.00176	-0.00182	-0.00059	0.00052	0.00113	0.00360	0.00122
4	-C.00211	-0.00863	-0.00574	-0.00019	-0.00034	0.00008	0.00091	0.00321	0.00309	0.00167
5	-0.01074	-0.00836	-0.00496	-0.00085	0.00417	0.00072	0.00484	-0.00041	0.00181	-0.01036
6	-C.01174	-0.01061	-0.00439	-0.00156	0.00051	0.00573	0.00341	0.00197	-0.00005	-0.00055
7	-C.01861	-0.00855	-0.00335	0.00150	0.00072	0.00323	0.00179	-0.00027	-0.00056	-0.00079
8	-C.01801	-0.00755	-0.00205	0.00069	0.00367	0.00143	0.00188	-0.00054	-0.00079	-0.00047
9	-C.01877	-0.00568	-0.00108	0.00418	0.00163	0.00211	-0.00067	-0.00089	-0.00039	-0.00028
10	-0.01770	-0.00537	0.00533	-0.00079	0.00247	-0.00097	-0.00088	-0.00039	-0.00018	-0.00021

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	C.01023	0.00584	0.00381	0.00308	0.00313	0.00306	0.00300	0.00295	0.00274	0.00245
2	C.02890	0.00933	0.00490	0.00419	0.00407	0.00376	0.00352	0.00317	0.00307	0.00276
3	C.05058	0.01314	0.00709	0.00557	0.00488	0.00441	0.00383	0.00358	0.00321	0.00305
4	C.07435	0.01717	0.00965	0.00679	0.00576	0.00487	0.00432	0.00377	0.00360	0.00313
5	C.09884	0.02291	0.01182	0.00828	0.00631	0.00546	0.00449	0.00427	0.00370	0.00364
6	C.12620	0.02864	0.01421	0.00976	0.00727	0.00574	0.00510	0.00428	0.00394	0.00348
7	0.15731	0.03448	0.01652	0.01045	0.00813	0.00636	0.00533	0.00457	0.00400	0.00358
8	C.18790	0.04022	0.01797	0.01175	0.00848	0.00694	0.00544	0.00464	0.00406	0.00358
9	C.21897	0.04400	0.01995	0.01220	0.00930	0.00702	0.00575	0.00472	0.00394	0.00363
10	C.24203	0.04860	0.02078	0.01373	0.00949	0.00738	0.00586	0.00458	0.00394	0.00371

VELOCITY COMPONENTS ON WING PANELS DUE TO WING PANEL PRESSURE SINGULARITIES

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.03182	0.04756	0.05745	0.06683	0.07637	0.08545	0.09354	0.10021	0.10359	0.11100
2	0.03264	0.03993	0.04642	0.05293	0.05892	0.06401	0.06799	0.07086	0.07334	0.07538
3	0.02977	0.03417	0.04015	0.04554	0.04993	0.05312	0.05513	0.05656	0.05857	0.05845
4	0.02924	0.03330	0.03902	0.04338	0.04636	0.04792	0.04879	0.04956	0.05076	0.04981
5	0.02963	0.03423	0.03902	0.04224	0.04356	0.04396	0.04430	0.04504	0.04524	0.04415
6	0.03143	0.03573	0.03929	0.04079	0.04075	0.04058	0.04082	0.04130	0.04097	0.03967
7	0.03463	0.03745	0.03907	0.03860	0.03809	0.03786	0.03780	0.03816	0.03746	0.03625
8	0.03588	0.03685	0.03581	0.03455	0.03386	0.03365	0.03350	0.03358	0.03280	0.03173
9	0.03556	0.03260	0.03000	0.02879	0.02837	0.02799	0.02804	0.02783	0.02708	0.02651
10	0.02829	0.02248	0.02037	0.01999	0.01942	0.01929	0.01935	0.01901	0.01855	0.02135

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.08743	-0.13067	-0.15784	-0.18360	-0.20982	-0.23475	-0.25698	-0.27532	-0.28458	-0.30494
2	-0.08966	-0.10970	-0.12752	-0.14541	-0.16186	-0.17584	-0.18678	-0.19469	-0.20148	-0.20710
3	-0.08180	-0.09387	-0.11030	-0.12513	-0.13716	-0.14594	-0.15147	-0.15540	-0.16092	-0.16059
4	-0.08032	-0.09147	-0.10720	-0.11918	-0.12736	-0.13165	-0.13405	-0.13617	-0.13946	-0.13686
5	-0.08141	-0.09405	-0.10721	-0.11604	-0.11966	-0.12078	-0.12171	-0.12374	-0.12429	-0.12130
6	-0.08636	-0.09816	-0.10794	-0.11208	-0.11194	-0.11149	-0.11214	-0.11347	-0.11256	-0.10899
7	-0.09513	-0.10288	-0.10735	-0.10604	-0.10464	-0.10400	-0.10385	-0.10483	-0.10291	-0.09960
8	-0.09859	-0.10125	-0.09839	-0.09493	-0.09303	-0.09246	-0.09202	-0.09224	-0.09011	-0.08717
9	-0.09771	-0.08958	-0.08242	-0.07909	-0.07795	-0.07690	-0.07703	-0.07644	-0.07438	-0.07283
10	-0.07773	-0.06176	-0.05597	-0.05492	-0.05335	-0.05300	-0.05316	-0.05223	-0.05097	-0.05865

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.07575	-0.05256	-0.04000	-0.03441	-0.03184	-0.03023	-0.02920	-0.02853	-0.02798	-0.02726
2	-0.12869	-0.08591	-0.07047	-0.06485	-0.06217	-0.06038	-0.05922	-0.05827	-0.05785	-0.05717
3	-0.15735	-0.09561	-0.07829	-0.07193	-0.06875	-0.06686	-0.06541	-0.06459	-0.06391	-0.06342
4	-0.18739	-0.10519	-0.08639	-0.07879	-0.07536	-0.07310	-0.07172	-0.07061	-0.07015	-0.06938
5	-0.21782	-0.11661	-0.09435	-0.08618	-0.08190	-0.07973	-0.07795	-0.07720	-0.07633	-0.07601
6	-0.25128	-0.12845	-0.10304	-0.09408	-0.08933	-0.08652	-0.08510	-0.08375	-0.08312	-0.08243
7	-0.29274	-0.14486	-0.11609	-0.10561	-0.10109	-0.09808	-0.09628	-0.09499	-0.09415	-0.09356
8	-0.33395	-0.16145	-0.12857	-0.11802	-0.11260	-0.10982	-0.10757	-0.10626	-0.10545	-0.10483
9	-0.37856	-0.17902	-0.14449	-0.13248	-0.12746	-0.12396	-0.12194	-0.12046	-0.11945	-0.11902
10	-0.42115	-0.20337	-0.16519	-0.15393	-0.14758	-0.14427	-0.14203	-0.14035	-0.13943	-0.13909

VELOCITY COMPONENTS ON WING PANELS DUE TO BODY LINE SOURCES AND DOUBLETS

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.00730	0.01207	0.00797	0.00568	0.00420	0.00319	0.00252	0.00201	0.00173	0.00135
2	0.01482	0.00983	0.00683	0.00483	0.00365	0.00279	0.00222	0.00180	0.00156	0.00125
3	0.01181	0.00833	0.00576	0.00417	0.00315	0.00246	0.00197	0.00162	0.00141	0.00116
4	0.00995	0.00709	0.00489	0.00362	0.00276	0.00219	0.00176	0.00146	0.00128	0.00108
5	0.00843	0.00607	0.00422	0.00315	0.00244	0.00194	0.00159	0.00132	0.00117	0.00101
6	0.00719	0.00512	0.00368	0.00276	0.00217	0.00175	0.00144	0.00121	0.00107	0.00095

7	C.00602	0.00441	0.00321	0.00245	0.00194	0.00158	0.00131	0.00111	0.00099	0.00147
8	C.00515	0.00384	0.00282	0.00219	0.00175	0.00144	0.00120	0.00113	0.00205	0.00305
9	C.00446	0.00336	0.00250	0.00196	0.00158	0.00131	0.00110	0.00226	0.00378	0.00445
10	C.00388	0.00294	0.00224	0.00177	0.00144	0.00120	0.00154	0.00404	0.00511	0.00512

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	C.02956	0.00088	-0.00227	-0.00266	-0.00239	-0.00199	-0.00167	-0.00138	-0.00121	-0.00097
2	C.01491	0.00201	-0.00168	-0.00201	-0.00194	-0.00163	-0.00139	-0.00118	-0.00104	-0.00086
3	C.01580	0.00242	-0.00100	-0.00154	-0.00150	-0.00135	-0.00116	-0.00101	-0.00090	-0.00077
4	0.01572	0.00276	-0.00046	-0.00117	-0.00119	-0.00112	-0.00098	-0.00086	-0.00078	-0.00068
5	C.01565	0.00306	-0.00010	-0.00082	-0.00095	-0.00091	-0.00083	-0.00074	-0.00068	-0.00061
6	C.01559	0.00347	0.00018	-0.00056	-0.00075	-0.00075	-0.00071	-0.00064	-0.00059	-0.00055
7	0.01576	0.00369	0.00043	-0.00036	-0.00058	-0.00062	-0.00060	-0.00056	-0.00052	-0.00136
8	C.01575	0.00385	0.00064	-0.00020	-0.00044	-0.00051	-0.00051	-0.00065	-0.00219	-0.00381
9	0.01572	0.00398	0.00080	-0.00006	-0.00033	-0.00042	-0.00043	-0.00242	-0.00492	-0.00607
10	C.01569	0.00412	0.00092	0.00006	-0.00024	-0.00034	-0.00117	-0.00526	-0.00715	-0.00721

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	C.04196	0.02315	0.01262	0.00777	0.00515	0.00360	0.00264	0.00201	0.00167	0.00125
2	C.04665	0.02343	0.01242	0.00751	0.00495	0.00347	0.00255	0.00195	0.00163	0.00125
3	C.04767	0.02336	0.01210	0.00726	0.00477	0.00336	0.00248	0.00191	0.00160	0.00126
4	C.04807	0.02305	0.01177	0.00703	0.00463	0.00327	0.00243	0.00188	0.00158	0.00128
5	C.04791	0.02263	0.01146	0.00684	0.00451	0.00320	0.00239	0.00185	0.00156	0.00130
6	C.04744	0.02217	0.01119	0.00668	0.00443	0.00315	0.00236	0.00183	0.00155	0.00132
7	C.04682	0.02176	0.01096	0.00656	0.00436	0.00311	0.00233	0.00182	0.00154	0.00136
8	C.04624	0.02142	C.01079	0.00647	0.00431	0.00308	0.00231	0.00181	0.00157	0.00144
9	0.04571	0.02114	0.01065	0.00640	0.00427	0.00306	0.00230	0.00186	0.00163	0.00150
10	0.04527	0.02092	0.01056	0.00635	0.00424	0.00304	0.00231	0.00191	0.00164	0.00153

VELOCITY COMPONENTS ON WING PANELS DUE TO WING SOURCES

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.03054	-0.02433	-0.01880	-0.01746	-0.01326	-0.00857	-0.00441	-0.00076	-0.00148	-0.00331
2	-C.00716	-0.00013	0.00337	0.00784	0.01289	0.01731	0.02137	0.02116	0.02006	0.01976
3	-C.00189	0.00375	0.00866	0.01475	0.01964	0.02390	0.02566	0.02337	0.02313	0.02314
4	-C.00060	0.00495	0.01303	0.01903	0.02368	0.02789	0.02409	0.02362	0.02363	0.02249
5	-C.00045	0.00850	0.01654	0.02209	0.02667	0.02357	0.02284	0.02278	0.02177	0.02115
6	C.00270	0.01161	C.01874	0.02411	0.02187	0.02069	0.02053	0.01941	0.01883	0.01855
7	C.00547	0.01297	0.01968	0.01900	0.01691	0.01658	0.01565	0.01475	0.01439	0.01431
8	C.00716	0.01395	0.01652	0.01237	0.01172	0.01114	0.00986	0.00933	0.00908	0.00913
9	C.00900	0.01548	0.00706	0.00593	0.00571	0.00393	0.00330	0.00293	0.00274	0.00307
10	C.01097	-0.00233	-0.00442	-0.00488	-0.00666	-0.00745	-0.00788	-0.00816	-0.00831	-0.00557

TRANSVERSE(V)

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10

CHORDWISE STATION										
1	0.02346	0.03708	0.03491	0.03503	0.02960	0.02265	0.01605	0.01001	0.01062	0.01346
2	-0.02849	-0.02351	-0.02451	-0.02979	-0.03719	-0.04422	-0.05097	-0.05142	-0.05010	-0.04997
3	-0.03063	-0.02966	-0.03436	-0.04316	-0.05102	-0.05823	-0.06183	-0.05896	-0.05886	-0.05917
4	-0.02439	-0.02819	-0.03974	-0.04951	-0.05761	-0.06514	-0.06024	-0.05998	-0.06022	-0.05877
5	-0.01502	-0.02915	-0.04278	-0.05283	-0.06136	-0.05790	-0.05738	-0.05766	-0.05636	-0.05561
6	-0.00956	-0.02868	-0.04278	-0.05337	-0.05185	-0.05088	-0.05112	-0.04977	-0.04905	-0.04882
7	-0.00345	-0.02458	-0.03956	-0.04162	-0.03967	-0.03983	-0.03886	-0.03773	-0.03731	-0.03742
8	0.00316	-0.02012	-0.03011	-0.02594	-0.02587	-0.02555	-0.02394	-0.02333	-0.02304	-0.02340
9	0.00822	-0.01726	-0.00930	-0.00904	-0.00947	-0.00721	-0.00649	-0.00607	-0.00586	-0.00692
10	0.01231	0.02030	0.02082	0.02039	0.02246	0.02335	0.02383	0.02413	0.02430	0.01669

VERTICAL(W)

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
CHORDWISE STATION										
1	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829
2	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022
3	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498
4	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695
5	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762
6	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197
7	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357
8	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034
9	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620
10	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180

VELOCITY COMPONENTS ON BODY PANELS DUE TO BODY PANEL PRESSURE SINGULARITIES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.00001	0.00005	0.00021	0.00237	-0.00227	-0.00020	-0.00006
2	-0.00005	0.00027	0.00138	0.00216	-0.01469	-0.00075	-0.00033
3	-0.00052	0.00173	0.00186	0.00797	-0.01657	-0.00637	0.00154
4	0.00585	0.00062	0.00396	0.01233	-0.01505	-0.00557	-0.00011
5	0.00371	0.00796	0.01080	0.01924	-0.00791	-0.00718	-0.00660
6	0.00312	0.00728	0.01299	0.02035	-0.01013	-0.00899	-0.01021
7	0.00414	0.00713	0.01264	0.02088	-0.01149	-0.01169	-0.01227
8	0.00573	0.00775	0.01265	0.02092	-0.01420	-0.01323	-0.01324
9	0.00656	0.00849	0.01321	0.02008	-0.01616	-0.01472	-0.01407
10	0.00658	0.00918	0.01354	0.01873	-0.01669	-0.01593	-0.01571
11	0.00953	0.01067	0.01012	0.00652	-0.01257	-0.01501	-0.01723
12	0.01206	0.01032	0.00496	-0.00362	-0.00903	-0.01117	-0.01349
13	0.00933	0.00758	0.00459	0.00050	-0.00359	-0.00654	-0.00881
14	0.00509	0.00434	0.00269	0.00035	-0.00181	-0.00357	-0.00446

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.00002	-0.00007	-0.00020	-0.00596	0.00481	0.00032	0.00007

2	-C.00030	-0.00024	-0.00124	-0.01267	0.03433	0.00062	0.00018
3	C.00066	-0.00126	-0.00160	-0.03914	0.06073	0.00357	0.00214
4	C.00161	0.00018	-0.00478	-0.07265	0.08568	0.00190	-0.00330
5	-C.00214	-0.00612	-0.01670	-0.11557	0.09947	0.00487	-0.00150
6	-C.00156	-0.00760	-0.02394	-0.15233	0.12297	0.01189	-0.00093
7	-C.00140	-0.00636	-0.02984	-0.19111	0.14787	0.01845	0.00166
8	-C.00024	-0.00658	-0.03818	-0.23251	0.17525	0.02630	0.00386
9	-C.00078	-0.00894	-0.04757	-0.27427	0.20354	0.03299	0.00535
10	-C.00041	-0.01033	-0.05722	-0.31380	0.23152	0.04069	0.00760
11	-C.00075	-0.01428	-0.07748	-0.34648	0.26423	0.05706	0.01113
12	-C.00284	-0.02348	-0.08785	-0.33889	0.28195	0.06784	0.01565
13	-C.00909	-0.03628	-0.09883	-0.34006	0.28743	0.07117	0.01720
14	-C.01224	-0.04436	-0.10617	-0.33167	0.26794	0.07675	0.02007

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	C.00000	0.00006	0.00038	-0.00070	-0.00300	0.00025	0.00001
2	C.00007	0.00019	-0.00130	-0.00422	-0.01759	0.00364	0.00004
3	C.00156	-0.00268	-0.00291	-0.00634	-0.03445	0.00104	0.00713
4	-C.00823	-0.00211	-0.00251	-0.01058	-0.04924	-0.00313	0.01238
5	-C.00754	-0.00742	-0.00842	-0.01333	-0.06344	-0.00924	0.00822
6	-C.00531	-0.00631	-0.01016	-0.02023	-0.07845	-0.01594	0.00177
7	-C.00394	-0.00728	-0.01464	-0.02221	-0.09342	-0.02427	-0.00578
8	-C.00293	-0.00831	-0.01802	-0.02674	-0.11023	-0.03270	-0.01392
9	-C.00116	-0.00837	-0.02181	-0.03142	-0.12898	-0.04325	-0.02270
10	C.00031	-0.00994	-0.02633	-0.03582	-0.14734	-0.05367	-0.03189
11	-C.00233	-0.01706	-0.03670	-0.04084	-0.16783	-0.07379	-0.05173
12	-C.01503	-0.03006	-0.04509	-0.04270	-0.17612	-0.08633	-0.06685
13	-C.03195	-0.04011	-0.04827	-0.04235	-0.18005	-0.09343	-0.07593
14	-C.04457	-0.04930	-0.05442	-0.08697	-0.21815	-0.10406	-0.08751

VELOCITY COMPONENTS ON BODY PANELS DUE TO WING PANEL PRESSURE SINGULARITIES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	C.	0.	0.	0.00781	-0.00539	0.	0.
2	C.	0.	0.00508	0.01168	-0.00805	-0.00341	0.
3	C.00165	0.00405	0.00761	0.01311	-0.00901	-0.00507	-0.00345
4	C.00524	0.00660	0.01039	0.01370	-0.00976	-0.00725	-0.00536
5	C.00737	0.00930	0.01156	0.01409	-0.01010	-0.00828	-0.00693
6	C.00952	0.01017	0.01224	0.01480	-0.01062	-0.00896	-0.00813
7	C.01036	0.01133	0.01331	0.01618	-0.01159	-0.00964	-0.00857
8	C.01102	0.01207	0.01478	0.01721	-0.01242	-0.01058	-0.00913
9	C.01207	0.01354	0.01574	0.01754	-0.01269	-0.01139	-0.01014
10	C.01354	0.01450	0.01628	0.01588	-0.01161	-0.01192	-0.01112
11	C.01462	0.01483	0.01059	0.00433	-0.00370	-0.00825	-0.01112
12	C.00877	0.00780	0.00604	0.00231	-0.00201	-0.00464	-0.00556
13	C.00547	0.00494	0.00353	0.00131	-0.00117	-0.00289	-0.00353
14	C.00366	0.00316	0.00225	0.00084	-0.00071	-0.00180	-0.00232

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.	0.	0.	0.00641	-0.00977	0.	0.
2	0.	0.	0.00331	0.03210	-0.03476	-0.00484	0.
3	0.00156	0.00398	0.01020	0.06377	-0.06234	-0.01175	-0.00290
4	0.00196	0.00702	0.01992	0.09588	-0.09011	-0.01876	-0.00442
5	0.00314	0.01172	0.02925	0.12699	-0.11723	-0.02637	-0.00693
6	0.00453	0.01588	0.03873	0.15859	-0.14501	-0.03365	-0.00838
7	0.00556	0.01934	0.04822	0.19204	-0.17454	-0.04125	-0.01039
8	0.00688	0.02395	0.05936	0.22829	-0.20662	-0.04996	-0.01261
9	0.00857	0.02975	0.07095	0.26563	-0.23930	-0.05920	-0.01538
10	0.01020	0.03518	0.08266	0.30142	-0.26969	-0.06823	-0.01774
11	0.01290	0.04465	0.10037	0.33357	-0.29213	-0.07914	-0.02197
12	0.01387	0.04723	0.10577	0.33703	-0.29459	-0.08190	-0.02191
13	0.01484	0.04988	0.10885	0.33849	-0.29569	-0.08355	-0.02271
14	0.01538	0.05114	0.11025	0.33059	-0.27069	-0.08239	-0.02270

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.	-0.	-0.	-0.01561	-0.00716	-0.	-0.
2	-0.	-0.	-0.00849	-0.03254	-0.00831	-0.00392	-0.
3	-0.00205	-0.00594	-0.01450	-0.04164	-0.00321	-0.00474	-0.00255
4	-0.00564	-0.00914	-0.01967	-0.04501	0.00610	-0.00396	-0.00184
5	-0.00692	-0.01147	-0.02132	-0.04654	0.01736	-0.00040	0.00043
6	-0.00735	-0.01110	-0.02173	-0.04812	0.02911	0.00429	0.00427
7	-0.00592	-0.01029	-0.02194	-0.05042	0.04140	0.00968	0.00920
8	-0.00408	-0.00896	-0.02217	-0.05173	0.05544	0.01534	0.01441
9	-0.00234	-0.00794	-0.02151	-0.05149	0.07116	0.02196	0.01984
10	-0.00046	-0.00602	-0.01990	-0.04675	0.08934	0.02950	0.02617
11	0.00666	0.00163	-0.00334	-0.00882	0.13024	0.05160	0.04220
12	0.02225	0.02014	0.01518	0.01330	0.15130	0.06974	0.06052
13	0.03393	0.03232	0.02897	0.02836	0.16584	0.08280	0.07262
14	0.04549	0.04523	0.04603	0.08547	0.21421	0.09934	0.08591

VELOCITY COMPONENTS ON BODY PANELS DUE TO WING SOURCES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.	0.	0.	-0.00275	-0.00157	0.	0.
2	0.	0.	-0.00645	-0.01262	-0.01134	-0.00520	0.
3	0.	-0.00666	-0.00902	-0.00734	-0.00732	-0.00785	-0.00585
4	-0.01113	-0.00699	-0.00646	-0.00369	-0.00407	-0.00956	-0.01252
5	-0.01124	-0.01126	-0.00895	-0.00581	-0.00707	-0.00877	-0.01019
6	-0.00891	-0.00826	-0.00613	-0.00292	-0.00375	-0.00602	-0.00737
7	-0.00613	-0.00531	-0.00300	0.00021	-0.00054	-0.00297	-0.00441
8	-0.00328	-0.00224	0.00013	0.00300	0.00234	0.00008	-0.00138

9	-C.00025	0.00076	0.00290	0.00529	C.00473	0.00290	0.00167
10	C.00267	0.00351	0.00541	0.00753	0.00705	0.00542	0.00439
11	C.00632	0.00717	C.00881	0.00308	0.00388	0.00862	0.00773
12	C.00740	0.00597	0.00467	0.00365	0.00410	0.00548	0.00453
13	C.00208	0.00191	0.00159	0.00128	0.00131	0.00154	0.00169
14	C.00092	0.00084	0.00070	0.00059	0.00061	0.00071	0.00078

TRANSVERSE(v)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	C.	0.	C.	-0.00015	-0.00106	0.	0.
2	C.	0.	-0.00215	-0.01927	-0.02232	-0.00479	0.
3	C.	-0.00460	-0.00837	-0.02398	-0.02413	-0.01141	-0.00699
4	-C.00260	-0.00750	-0.01101	-0.02218	-0.02128	-0.00715	-0.00056
5	-C.00052	0.00001	-0.00376	-0.01012	-0.00750	-0.00338	-0.00047
6	-C.00028	-0.00024	-0.00255	-0.00459	-0.00255	-0.00168	-0.00015
7	-C.00016	-0.00001	-0.00119	0.00100	0.00295	-0.00000	0.00020
8	-C.00004	0.00046	0.00043	0.00642	0.00817	0.00181	0.00071
9	C.00015	0.00109	0.00215	0.01104	0.01251	0.00342	0.00112
10	C.00032	0.00154	0.00361	0.01487	0.01607	0.00468	0.00134
11	C.00066	0.00205	0.00560	0.01433	0.01246	0.00665	0.00165
12	-0.00173	-0.00105	0.00067	0.00264	0.00152	-0.00122	-0.00010
13	C.00013	0.00054	0.00127	0.00194	0.00169	0.00090	0.00026
14	C.00015	0.00048	0.00086	0.00112	0.00099	0.00059	0.00019

VERTICAL(w)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-C.	-0.	-0.	0.00415	-0.00212	-0.	-0.
2	-C.	-0.	0.00997	0.03008	-0.02056	-0.00664	-0.
3	-C.	0.01007	0.01697	0.02118	-0.01490	-0.01133	-0.00630
4	0.01365	0.01149	0.01425	0.01298	-0.00936	-0.01133	-0.01238
5	C.01435	0.01457	0.01285	0.00678	-0.00548	-0.00945	-0.01063
6	C.01206	0.01125	0.00839	0.00032	-0.00087	-0.00633	-0.00813
7	C.00897	0.00761	0.00357	-0.00594	0.00360	-0.00291	-0.00532
8	C.00555	0.00359	-0.00133	-0.01125	0.00744	0.00059	-0.00228
9	C.00174	-0.00049	-0.00573	-0.01520	0.01034	0.00380	0.00088
10	-C.00209	-0.00429	-0.00956	-0.01861	0.01287	0.00662	0.00377
11	-0.00716	-0.00946	-0.01471	-0.00827	0.00607	0.01034	0.00750
12	-C.00928	-0.00750	-0.00580	-0.00246	0.00212	0.00487	0.00492
13	-C.00328	-0.00306	-0.00239	-0.00094	0.00081	0.00182	0.00215
14	-C.00165	-0.00150	-0.00111	-0.00043	0.00035	0.00086	0.00106

PRESSURES, FORCES, AND MOMENTS ON ISOLATED BODY

CD = 0.00069 CL = 0.00174 CM = -0.00454

BODY PRESSURE COEFFICIENTS(CP)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000

X							
0.	0.09514	0.09705	0.10048	0.10494	0.10961	0.11312	0.11491
0.7243	0.09514	0.09705	0.10048	0.10494	0.10961	0.11312	0.11491
1.4485	0.08457	0.08842	0.09533	0.10428	0.11359	0.12054	0.12408
2.1728	0.03212	0.03515	0.04068	0.04779	0.05517	0.06068	0.06349
2.8970	0.03244	0.03547	0.04088	0.04786	0.05509	0.06046	0.06320
3.6213	0.02039	0.02306	0.02784	0.03402	0.04042	0.04519	0.04762
4.3456	0.02152	0.02401	0.02846	0.03421	0.04017	0.04459	0.04685
5.0698	0.01501	0.01718	0.02107	0.02610	0.03133	0.03524	0.03723
5.7941	0.01573	0.01767	0.02115	0.02565	0.03033	0.03383	0.03561
6.5184	0.00982	0.01175	0.01521	0.01968	0.02434	0.02781	0.02958
7.2426	0.00820	0.01023	0.01385	0.01854	0.02340	0.02702	0.02886
7.9669	0.00510	0.00679	0.00984	0.01379	0.01792	0.02101	0.02259
8.6911	0.00124	0.00335	0.00712	0.01198	0.01702	0.02076	0.02266
9.4154	-0.00138	0.00048	0.00381	0.00813	0.01263	0.01599	0.01771
10.1397	-0.00405	-0.00213	0.00132	0.00577	0.01041	0.01387	0.01563
10.8639	-0.00785	-0.00589	-0.00237	0.00218	0.00692	0.01046	0.01226
11.5882	-0.00980	-0.00813	-0.00512	-0.00120	0.00292	0.00601	0.00760
12.3124	-0.01342	-0.01171	-0.00862	-0.00459	-0.00037	0.00280	0.00443
13.0367	-0.01860	-0.01686	-0.01373	-0.00965	-0.00537	-0.00214	-0.00048
13.7610	-0.02086	-0.01917	-0.01611	-0.01211	-0.00790	-0.00471	-0.00308
14.5023	-0.02856	-0.02660	-0.02307	-0.01846	-0.01363	-0.00999	-0.00812
15.2436	-0.03374	-0.03196	-0.02875	-0.02452	-0.02003	-0.01661	-0.01485
15.9849	-0.04651	-0.04524	-0.04290	-0.03970	-0.03615	-0.03335	-0.03187
16.7262	-0.03757	-0.03700	-0.03587	-0.03419	-0.03216	-0.03044	-0.02950
17.4675	-0.03057	-0.03050	-0.03025	-0.02967	-0.02876	-0.02787	-0.02733
18.2088	-0.02522	-0.02548	-0.02581	-0.02599	-0.02587	-0.02555	-0.02532
18.9501	-0.02120	-0.02165	-0.02233	-0.02298	-0.02336	-0.02343	-0.02340
19.6914	-0.01821	-0.01874	-0.01959	-0.02049	-0.02116	-0.02147	-0.02157
20.4327	-0.01599	-0.01654	-0.01744	-0.01843	-0.01923	-0.01966	-0.01983
21.1740	-0.01434	-0.01487	-0.01574	-0.01671	-0.01753	-0.01801	-0.01820
21.9153	-0.01311	-0.01359	-0.01437	-0.01527	-0.01604	-0.01651	-0.01671
22.6566	-0.01218	-0.01259	-0.01327	-0.01406	-0.01475	-0.01517	-0.01535
23.3979	-0.01146	-0.01180	-0.01238	-0.01304	-0.01362	-0.01398	-0.01414
24.1392	-0.01088	-0.01116	-0.01163	-0.01217	-0.01264	-0.01294	-0.01307
24.8805	-0.01041	-0.01063	-0.01100	-0.01142	-0.01180	-0.01204	-0.01214
25.6218	-0.00999	-0.01017	-0.01046	-0.01079	-0.01108	-0.01126	-0.01134
26.3631	-0.00963	-0.00977	-0.00999	-0.01024	-0.01046	-0.01059	-0.01065
27.1044	-0.00931	-0.00941	-0.00958	-0.00977	-0.00992	-0.01002	-0.01005
27.8457	-0.00901	-0.00908	-0.00921	-0.00935	-0.00946	-0.00953	-0.00955
28.5870	-0.00873	-0.00879	-0.00889	-0.00899	-0.00907	-0.00911	-0.00912
29.3283	-0.00847	-0.00852	-0.00859	-0.00867	-0.00872	-0.00875	-0.00875
30.0696	-0.01075	-0.01078	-0.01083	-0.01085	-0.01085	-0.01082	-0.01080
30.8109	-0.01854	-0.01857	-0.01859	-0.01858	-0.01850	-0.01839	-0.01832
31.5522	-0.02599	-0.02613	-0.02633	-0.02650	-0.02655	-0.02651	-0.02646
32.2935	-0.03001	-0.03037	-0.03093	-0.03153	-0.03198	-0.03220	-0.03227
33.0348	-0.03267	-0.03329	-0.03431	-0.03546	-0.03644	-0.03700	-0.03724
33.7761	-0.03823	-0.03914	-0.04066	-0.04240	-0.04392	-0.04484	-0.04525
34.5174	-0.03882	-0.04011	-0.04228	-0.04483	-0.04712	-0.04859	-0.04926
35.2587	-0.02672	-0.02840	-0.03127	-0.03473	-0.03799	-0.04019	-0.04124
36.0000	-0.00791	-0.00973	-0.01292	-0.01686	-0.02074	-0.02347	-0.02481

INCREMENTAL PRESSURES, FORCES, AND MOMENTS ON BODY PANELS DUE TO WING

CD = 0.00096 CL = 0.01361 CM = -0.31352

BODY PANEL PRESSURE COEFFICIENT(CP)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.00003	-0.00010	-0.00042	-0.01489	0.01849	0.00040	0.00012
2	0.00010	-0.00055	-0.00001	-0.00249	0.06808	0.01881	0.00065
3	-0.00227	0.00216	-0.00090	-0.02776	0.06480	0.03880	0.01557
4	0.00006	-0.00046	-0.01573	-0.04539	0.05622	0.04498	0.03665
5	0.00031	-0.01197	-0.02678	-0.05615	0.04827	0.04880	0.04861
6	-0.00743	-0.01827	-0.03809	-0.06675	0.04721	0.04834	0.05280
7	-0.01660	-0.02618	-0.04609	-0.07761	0.04560	0.04910	0.05186
8	-0.02858	-0.03498	-0.05561	-0.08652	0.04710	0.04796	0.04870
9	-0.03606	-0.04516	-0.06449	-0.09134	0.04675	0.04679	0.04614
10	-0.04454	-0.05382	-0.07164	-0.09054	0.04099	0.04515	0.04595
11	-0.05903	-0.06461	-0.06089	-0.03077	0.02389	0.02939	0.04212
12	-0.05477	-0.04769	-0.03240	-0.00567	0.01334	0.02052	0.02948
13	-0.03314	-0.02870	-0.01980	-0.00637	0.00670	0.01569	0.02155
14	-0.01914	-0.01662	-0.01132	-0.00356	0.00381	0.00934	0.01207

BODY PANEL SLOPE

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.04467	-0.03189	-0.00874	0.02168	0.05407	0.07888	0.09165
2	-0.06197	-0.04920	-0.02605	0.00445	0.03686	0.06157	0.07434
3	-0.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
4	-0.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
5	-0.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
6	-0.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
7	-0.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
8	-0.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
9	-0.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
10	-0.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
11	-0.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
12	-0.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
13	-0.06816	-0.05539	-0.03224	-0.00171	0.03071	0.05539	0.06816
14	-0.08237	-0.06960	-0.04645	-0.01587	0.01657	0.04117	0.05394

PRESSURES, FORCES, AND MOMENTS ON WING PANELS IN PRESENCE OF BODY

CD = 0.00936 CL = 0.15761 CM = -4.80539

SPANWISE CD DISTRIBUTION

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
	0.00071	0.00113	0.00110	0.00110	0.00105	0.00101	0.00099	0.00099	0.00050	0.00077

SPANWISE CL DISTRIBUTION

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
	0.00732	0.01362	0.01477	0.01577	0.01654	0.01719	0.01776	0.01823	0.00930	0.02712

UPPER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.01519	-0.08125	-0.11232	-0.14101	-0.16991	-0.20337	-0.23151	-0.25822	-0.26508	-0.27718
2	-0.08306	-0.11202	-0.13926	-0.15618	-0.18528	-0.20326	-0.22167	-0.22602	-0.23172	-0.23449
3	-0.08750	-0.10969	-0.13409	-0.15130	-0.17253	-0.18836	-0.19601	-0.19369	-0.19498	-0.19816
4	-0.09262	-0.11492	-0.13876	-0.15340	-0.17119	-0.18404	-0.17647	-0.17400	-0.17742	-0.17419
5	-0.10075	-0.12253	-0.14526	-0.15934	-0.16550	-0.16484	-0.15795	-0.16757	-0.16240	-0.17610
6	-0.10757	-0.13464	-0.15101	-0.16307	-0.15539	-0.14470	-0.14845	-0.14921	-0.14992	-0.14643
7	-0.12646	-0.13909	-0.15332	-0.14492	-0.14052	-0.13588	-0.13571	-0.13747	-0.13509	-0.13331
8	-0.12798	-0.13913	-0.13905	-0.12447	-0.11776	-0.11929	-0.11535	-0.11779	-0.11772	-0.11696
9	-0.13086	-0.13256	-0.10763	-0.09673	-0.09933	-0.09394	-0.09629	-0.09779	-0.09814	-0.09891
10	-0.12032	-0.07903	-0.06117	-0.06923	-0.06008	-0.06255	-0.06236	-0.06546	-0.06608	-0.07805

LOWER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.10993	0.10444	0.11255	0.12107	0.13011	0.13086	0.13374	0.13136	0.13941	0.16147
2	0.03511	0.03331	0.02966	0.03819	0.02947	0.03019	0.02526	0.03307	0.03730	0.04333
3	0.02279	0.01646	0.01391	0.01700	0.01078	0.00570	0.00530	0.01448	0.02183	0.01743
4	0.01905	0.01125	0.00818	0.00978	0.00162	-0.00688	0.00598	0.01231	0.01343	0.01322
5	0.01575	0.01080	0.00504	0.00237	0.00087	0.00347	0.01307	0.00492	0.01191	-0.00887
6	0.02059	0.00827	0.00463	-0.00289	0.00599	0.01779	0.01445	0.01565	0.01343	0.01200
7	0.01797	0.01605	0.00673	0.01464	0.01765	0.02206	0.02203	0.02154	0.02133	0.01829
8	0.02633	0.01828	0.01428	0.02578	0.03054	0.02793	0.03177	0.02935	0.02615	0.02251
9	0.02743	0.01370	0.03115	0.03831	0.03365	0.03809	0.03567	0.03311	0.02955	0.02635
10	0.01693	0.03874	0.04939	0.03899	0.04653	0.04331	0.04376	0.03898	0.03638	0.03515

WING PANEL PRESSURE DIFFERENCE(CL)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.12730	0.19025	0.22981	0.26732	0.30548	0.34179	0.37415	0.40085	0.41434	0.44398
2	0.13054	0.15971	0.18567	0.21171	0.23566	0.25602	0.27194	0.28346	0.29335	0.30152
3	0.11910	0.13667	0.16059	0.18218	0.19970	0.21249	0.22053	0.22625	0.23429	0.23382
4	0.11695	0.13318	0.15608	0.17352	0.18543	0.19167	0.19517	0.19826	0.20305	0.19926
5	0.11853	0.13694	0.15610	0.16895	0.17423	0.17585	0.17720	0.18016	0.18097	0.17661
6	0.12574	0.14291	0.15716	0.16318	0.16298	0.16233	0.16327	0.16521	0.16389	0.15868
7	0.13850	0.14978	0.15630	0.15438	0.15235	0.15143	0.15120	0.15263	0.14983	0.14502
8	0.14354	0.14741	0.14326	0.13822	0.13545	0.13462	0.13399	0.13431	0.13121	0.12692
9	0.14225	0.13042	0.12001	0.11515	0.11349	0.11197	0.11216	0.11131	0.10830	0.10605
10	0.11317	0.08992	0.08150	0.07996	0.07768	0.07717	0.07740	0.07605	0.07422	0.08541

UPPER SURFACE WING PANEL SLOPE

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.08472	0.08472	0.08472	0.08472	0.08472	0.08472	0.08472	0.08472	0.08472	0.08472

2	-0.00293	-0.00293	-0.00293	-0.00293	-0.00293	-0.00293	-0.00293	-0.00293	-0.00293	-0.00293
3	-0.03411	-0.03411	-0.03411	-0.03411	-0.03411	-0.03411	-0.03411	-0.03411	-0.03411	-0.03411
4	-0.05802	-0.05802	-0.05802	-0.05802	-0.05802	-0.05802	-0.05802	-0.05802	-0.05802	-0.05802
5	-0.07869	-0.07869	-0.07869	-0.07869	-0.07869	-0.07869	-0.07869	-0.07869	-0.07869	-0.07869
6	-0.09961	-0.09961	-0.09961	-0.09961	-0.09961	-0.09961	-0.09961	-0.09961	-0.09961	-0.09961
7	-0.12217	-0.12217	-0.12217	-0.12217	-0.12217	-0.12217	-0.12217	-0.12217	-0.12217	-0.12217
8	-0.14015	-0.14015	-0.14015	-0.14015	-0.14015	-0.14015	-0.14015	-0.14015	-0.14015	-0.14015
9	-0.16008	-0.16008	-0.16008	-0.16008	-0.16008	-0.16008	-0.16008	-0.16008	-0.16008	-0.16008
10	-0.18566	-0.18566	-0.18566	-0.18566	-0.18566	-0.18566	-0.18566	-0.18566	-0.18566	-0.18566

LOWER SURFACE WING PANEL SLOPE

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
CHORDWISE STATION										
1	-0.13186	-0.13186	-0.13186	-0.13186	-0.13186	-0.13186	-0.13186	-0.13186	-0.13186	-0.13186
2	-0.10337	-0.10337	-0.10337	-0.10337	-0.10337	-0.10337	-0.10337	-0.10337	-0.10337	-0.10337
3	-0.08408	-0.08408	-0.08408	-0.08408	-0.08408	-0.08408	-0.08408	-0.08408	-0.08408	-0.08408
4	-0.07192	-0.07192	-0.07192	-0.07192	-0.07192	-0.07192	-0.07192	-0.07192	-0.07192	-0.07192
5	-0.06345	-0.06345	-0.06345	-0.06345	-0.06345	-0.06345	-0.06345	-0.06345	-0.06345	-0.06345
6	-0.05566	-0.05566	-0.05566	-0.05566	-0.05566	-0.05566	-0.05566	-0.05566	-0.05566	-0.05566
7	-0.05504	-0.05504	-0.05504	-0.05504	-0.05504	-0.05504	-0.05504	-0.05504	-0.05504	-0.05504
8	-0.05947	-0.05947	-0.05947	-0.05947	-0.05947	-0.05947	-0.05947	-0.05947	-0.05947	-0.05947
9	-0.06769	-0.06769	-0.06769	-0.06769	-0.06769	-0.06769	-0.06769	-0.06769	-0.06769	-0.06769
10	-0.08206	-0.08206	-0.08206	-0.08206	-0.08206	-0.08206	-0.08206	-0.08206	-0.08206	-0.08206

FORCES AND MOMENTS ON WING-BODY COMBINATION

CD = 0.01101 CL = 0.17296 CM = -5.12346

TR-805 GIVEN SHAPE

DELTA ANGLE OF ATTACK = 1.0000 DEG.

ANGLE OF ATTACK WITH RESPECT TO BODY AXIS = 5.0000 DEG.

VELOCITY COMPONENTS ON BODY DUE TO BODY LINE SOURCES AND DOUBLETS

AXIAL(U)

THETA(DEG.) X	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
0.	-0.04936	-0.05185	-0.05636	-0.06230	-0.06861	-0.07341	-0.07590
0.7243	-0.04936	-0.05185	-0.05636	-0.06230	-0.06861	-0.07341	-0.07590
1.4485	-0.04281	-0.04652	-0.05325	-0.06213	-0.07155	-0.07873	-0.08244
2.1728	-0.01429	-0.01737	-0.02295	-0.03031	-0.03812	-0.04407	-0.04715
2.8970	-0.01428	-0.01731	-0.02280	-0.03004	-0.03773	-0.04359	-0.04662
3.6213	-0.00789	-0.01064	-0.01561	-0.02217	-0.02913	-0.03443	-0.03718
4.3456	-0.00845	-0.01108	-0.01584	-0.02211	-0.02877	-0.03384	-0.03647
5.0698	-0.00511	-0.00749	-0.01181	-0.01751	-0.02355	-0.02816	-0.03054
5.7941	-0.00552	-0.00775	-0.01179	-0.01712	-0.02278	-0.02708	-0.02931
6.5184	-0.00256	-0.00471	-0.00860	-0.01374	-0.01920	-0.02336	-0.02551
7.2426	-0.00179	-0.00395	-0.00787	-0.01303	-0.01852	-0.02269	-0.02486
7.9669	-0.00031	-0.00224	-0.00575	-0.01037	-0.01528	-0.01902	-0.02095
8.6911	0.00156	-0.00053	-0.00433	-0.00933	-0.01465	-0.01870	-0.02079
9.4154	0.00276	0.00084	-0.00263	-0.00722	-0.01209	-0.01579	-0.01771
10.1397	0.00397	0.00207	-0.00137	-0.00592	-0.01074	-0.01441	-0.01632
10.8639	0.00572	0.00385	0.00048	-0.00398	-0.00871	-0.01231	-0.01418
11.5882	0.00651	0.00485	0.00183	-0.00215	-0.00638	-0.00960	-0.01127
12.3124	0.00810	0.00647	0.00353	-0.00034	-0.00446	-0.00759	-0.00921
13.0367	0.01037	0.00883	0.00603	0.00233	-0.00159	-0.00458	-0.00613
13.7610	0.01119	0.00973	0.00708	0.00359	-0.00011	-0.00293	-0.00439
14.5023	0.01453	0.01305	0.01038	0.00685	0.00311	0.00026	-0.00122
15.2436	0.01650	0.01524	0.01294	0.00990	0.00668	0.00423	0.00296
15.9849	0.02177	0.02097	0.01952	0.01762	0.01559	0.01405	0.01325
16.7262	0.01682	0.01640	0.01563	0.01461	0.01354	0.01272	0.01229
17.4675	0.01301	0.01286	0.01258	0.01221	0.01182	0.01152	0.01136
18.2088	0.01015	0.01018	0.01023	0.01030	0.01037	0.01042	0.01045
18.9501	0.00803	0.00817	0.00843	0.00876	0.00912	0.00939	0.00953
19.6914	0.00648	0.00668	0.00705	0.00753	0.00804	0.00843	0.00863
20.4327	0.00536	0.00558	0.00599	0.00652	0.00709	0.00752	0.00774
21.1740	0.00454	0.00477	0.00517	0.00570	0.00626	0.00668	0.00691
21.9153	0.00395	0.00415	0.00452	0.00501	0.00553	0.00592	0.00613
22.6566	0.00351	0.00369	0.00401	0.00444	0.00489	0.00523	0.00541
23.3979	0.00318	0.00333	0.00360	0.00396	0.00434	0.00463	0.00477
24.1392	0.00291	0.00303	0.00325	0.00355	0.00385	0.00409	0.00421
24.8805	0.00269	0.00279	0.00296	0.00319	0.00344	0.00362	0.00372
25.6218	0.00251	0.00258	0.00272	0.00289	0.00308	0.00322	0.00330
26.3631	0.00234	0.00240	0.00250	0.00263	0.00277	0.00288	0.00294
27.1044	0.00219	0.00223	0.00230	0.00240	0.00251	0.00259	0.00263
27.8457	0.00205	0.00208	0.00213	0.00220	0.00228	0.00234	0.00237
28.5870	0.00191	0.00193	0.00197	0.00202	0.00208	0.00212	0.00214
29.3283	0.00179	0.00180	0.00183	0.00187	0.00191	0.00194	0.00195
30.0696	0.00277	0.00280	0.00287	0.00295	0.00304	0.00311	0.00315
30.8109	0.00606	0.00620	0.00644	0.00677	0.00711	0.00737	0.00751

31.5522	C.00898	0.00930	0.00986	0.01061	0.01140	0.01201	0.01232
32.2935	C.01018	0.01071	0.01168	0.01296	0.01431	0.01535	0.01588
33.0348	C.01062	0.01140	0.01283	0.01471	0.01671	0.01823	0.01901
33.7761	0.01210	0.01320	0.01520	0.01784	0.02064	0.02277	0.02388
34.5174	C.01123	0.01267	0.01526	0.01869	0.02232	0.02509	0.02652
35.2587	C.00519	0.00680	0.00973	0.01359	0.01769	0.02081	0.02243
36.0000	-C.00304	-0.00149	0.00132	0.00503	0.00897	0.01197	0.01352

RADIAL(VR)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
X							
0.	C.12635	0.13429	0.14869	0.16766	0.18782	0.20316	0.21110
0.7243	C.12635	0.13429	0.14869	0.16766	0.18782	0.20316	0.21110
1.4485	C.11135	0.12210	0.14158	0.16728	0.19456	0.21533	0.22608
2.1728	C.05666	0.06821	0.08914	0.11674	0.14605	0.16836	0.17991
2.8970	C.04824	0.06057	0.08291	0.11236	0.14365	0.16746	0.17979
3.6213	C.03148	0.04414	0.06708	0.09732	0.12945	0.15390	0.16656
4.3456	C.02740	0.04032	0.06376	0.09465	0.12747	0.15245	0.16537
5.0698	C.01792	0.03089	0.05441	0.08542	0.11835	0.14342	0.15640
5.7941	C.01522	0.02821	0.05177	0.08282	0.11581	0.14092	0.15391
6.5184	C.00743	0.02057	0.04439	0.07581	0.10917	0.13457	0.14771
7.2426	0.00321	0.01661	0.04090	0.07293	0.10695	0.13284	0.14624
7.9669	-0.00211	0.01120	0.03531	0.06711	0.10088	0.12659	0.13989
8.6911	-C.00804	0.00576	0.03078	0.06378	0.09882	0.12550	0.13930
9.4154	-C.01314	0.00069	0.02575	0.05879	0.09389	0.12061	0.13444
10.1397	-0.01827	-0.00418	0.02135	0.05502	0.09078	0.11801	0.13209
10.8639	-0.02443	-0.01008	0.01593	0.05023	0.08666	0.11439	0.12874
11.5882	-C.02927	-0.01490	0.01115	0.04549	0.08197	0.10974	0.12411
12.3124	-C.03547	-0.02084	0.00569	0.04066	0.07780	0.10608	0.12071
13.0367	-0.04323	-0.02833	-0.00133	0.03428	0.07210	0.10089	0.11578
13.7610	-C.04897	-0.03381	-0.00633	0.02990	0.06839	0.09768	0.11284
14.5023	-C.05946	-0.04378	-0.01536	0.02212	0.06193	0.09224	0.10792
15.2436	-C.06877	-0.05283	-0.02393	0.01418	0.05465	0.08546	0.10140
15.9849	-C.08519	-0.06923	-0.04029	-0.00214	0.03839	0.06924	0.08520
16.7262	-C.08520	-0.06923	-0.04029	-0.00214	0.03839	0.06923	0.08520
17.4675	-C.08520	-0.06923	-0.04029	-0.00214	0.03839	0.06923	0.08520
18.2088	-0.08520	-0.06923	-0.04029	-0.00214	0.03839	0.06923	0.08520
18.9501	-C.08520	-0.06923	-0.04029	-0.00214	0.03839	0.06923	0.08520
19.6914	-C.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
20.4327	-C.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
21.1740	-C.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
21.9153	-C.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
22.6566	-C.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
23.3979	-C.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
24.1392	-C.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
24.8805	-C.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
25.6218	-C.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
26.3631	-C.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
27.1044	-C.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
27.8457	-C.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
28.5870	-C.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
29.3283	-C.08520	-0.06923	-0.04030	-0.00214	0.03838	0.06923	0.08520
30.0696	-C.08728	-0.07132	-0.04238	-0.00423	0.03630	0.06715	0.08311
30.8109	-C.09434	-0.07838	-0.04945	-0.01129	0.02923	0.06007	0.07604
31.5522	-C.10257	-0.08661	-0.05768	-0.01954	0.02097	0.05181	0.06777
32.2935	-C.10940	-0.09345	-0.06453	-0.02641	0.01408	0.04491	0.06086

33.0348	-0.11599	-0.10005	-0.07116	-0.03306	0.00741	0.03821	0.05415
33.7761	-0.12568	-0.10976	-0.08090	-0.04285	-0.00244	0.02832	0.04424
34.5174	-0.13270	-0.11680	-0.08799	-0.04999	-0.00964	0.02108	0.03698
35.2587	-0.12984	-0.11395	-0.08514	-0.04715	-0.00681	0.02390	0.03979
36.0000	-0.11909	-0.10318	-0.07434	-0.03631	0.00408	0.03483	0.05074

TANGENTIAL(VT)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
X							
0.	-0.00695	-0.01954	-0.02847	-0.03209	-0.02883	-0.01954	-0.00695
0.7243	-0.00695	-0.01954	-0.02847	-0.03209	-0.02883	-0.01954	-0.00695
1.4485	-0.00874	-0.02459	-0.03583	-0.04038	-0.03627	-0.02459	-0.00874
2.1728	-0.01111	-0.03125	-0.04554	-0.05132	-0.04611	-0.03125	-0.01111
2.8970	-0.01218	-0.03425	-0.04990	-0.05624	-0.05052	-0.03425	-0.01218
3.6213	-0.01308	-0.03678	-0.05359	-0.06040	-0.05426	-0.03678	-0.01308
4.3456	-0.01356	-0.03814	-0.05557	-0.06263	-0.05626	-0.03814	-0.01356
5.0698	-0.01397	-0.03930	-0.05726	-0.06454	-0.05797	-0.03930	-0.01397
5.7941	-0.01417	-0.03986	-0.05808	-0.06546	-0.05880	-0.03986	-0.01417
6.5184	-0.01442	-0.04055	-0.05909	-0.06660	-0.05983	-0.04055	-0.01442
7.2426	-0.01467	-0.04125	-0.06010	-0.06774	-0.06085	-0.04125	-0.01467
7.9669	-0.01489	-0.04188	-0.06102	-0.06877	-0.06178	-0.04188	-0.01489
8.6911	-0.01518	-0.04271	-0.06223	-0.07013	-0.06300	-0.04271	-0.01518
9.4154	-0.01550	-0.04358	-0.06350	-0.07157	-0.06429	-0.04358	-0.01550
10.1397	-0.01581	-0.04446	-0.06478	-0.07302	-0.06559	-0.04446	-0.01581
10.8639	-0.01617	-0.04547	-0.06626	-0.07467	-0.06708	-0.04547	-0.01617
11.5882	-0.01649	-0.04639	-0.06759	-0.07618	-0.06843	-0.04639	-0.01649
12.3124	-0.01683	-0.04734	-0.06898	-0.07774	-0.06984	-0.04734	-0.01683
13.0367	-0.01723	-0.04847	-0.07063	-0.07960	-0.07151	-0.04847	-0.01723
13.7610	-0.01765	-0.04963	-0.07232	-0.08151	-0.07322	-0.04963	-0.01765
14.5023	-0.01821	-0.05121	-0.07461	-0.08409	-0.07554	-0.05121	-0.01821
15.2436	-0.01883	-0.05296	-0.07717	-0.08697	-0.07813	-0.05296	-0.01883
15.9849	-0.01956	-0.05500	-0.08014	-0.09032	-0.08114	-0.05500	-0.01956
16.7262	-0.02000	-0.05624	-0.08195	-0.09236	-0.08297	-0.05624	-0.02000
17.4675	-0.02021	-0.05685	-0.08284	-0.09336	-0.08387	-0.05685	-0.02021
18.2088	-0.02027	-0.05701	-0.08307	-0.09362	-0.08411	-0.05701	-0.02027
18.9501	-0.02022	-0.05687	-0.08286	-0.09339	-0.08389	-0.05687	-0.02022
19.6914	-0.02010	-0.05655	-0.08239	-0.09286	-0.08342	-0.05655	-0.02010
20.4327	-0.01996	-0.05613	-0.08179	-0.09218	-0.08281	-0.05613	-0.01996
21.1740	-0.01980	-0.05569	-0.08115	-0.09146	-0.08216	-0.05569	-0.01980
21.9153	-0.01965	-0.05527	-0.08053	-0.09076	-0.08154	-0.05527	-0.01965
22.6566	-0.01951	-0.05489	-0.07997	-0.09013	-0.08097	-0.05489	-0.01951
23.3979	-0.01940	-0.05456	-0.07949	-0.08959	-0.08048	-0.05456	-0.01940
24.1392	-0.01930	-0.05428	-0.07909	-0.08914	-0.08007	-0.05428	-0.01930
24.8805	-0.01922	-0.05406	-0.07876	-0.08877	-0.07975	-0.05406	-0.01922
25.6218	-0.01916	-0.05388	-0.07851	-0.08848	-0.07949	-0.05388	-0.01916
26.3631	-0.01911	-0.05375	-0.07831	-0.08826	-0.07929	-0.05375	-0.01911
27.1044	-0.01907	-0.05364	-0.07816	-0.08809	-0.07914	-0.05364	-0.01907
27.8457	-0.01905	-0.05357	-0.07805	-0.08797	-0.07903	-0.05357	-0.01905
28.5870	-0.01903	-0.05351	-0.07797	-0.08788	-0.07895	-0.05351	-0.01903
29.3283	-0.01901	-0.05347	-0.07792	-0.08782	-0.07889	-0.05347	-0.01901
30.0696	-0.01904	-0.05355	-0.07803	-0.08794	-0.07900	-0.05355	-0.01904
30.8109	-0.01918	-0.05396	-0.07862	-0.08861	-0.07960	-0.05396	-0.01918
31.5522	-0.01942	-0.05462	-0.07958	-0.08969	-0.08057	-0.05462	-0.01942
32.2935	-0.01968	-0.05534	-0.08064	-0.09088	-0.08164	-0.05534	-0.01968
33.0348	-0.01993	-0.05605	-0.08166	-0.09204	-0.08268	-0.05605	-0.01993
33.7761	-0.02021	-0.05684	-0.08282	-0.09335	-0.08386	-0.05684	-0.02021

34.5174	-0.02044	-0.05749	-0.08376	-0.09441	-0.08481	-0.05749	-0.02044
35.2587	-0.02041	-0.05741	-0.08366	-0.09428	-0.08470	-0.05741	-0.02041
36.0000	-0.02005	-0.05640	-0.08218	-0.09262	-0.08320	-0.05640	-0.02005

VELOCITY COMPONENTS ON WING PANELS DUE TO BODY PANEL PRESSURE SINGULARITIES

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.00420	-0.00066	0.00093	0.00316	0.00193	0.00180	0.00023	-0.00004	-0.00072	-0.00309
2	-0.00128	0.00103	0.00493	0.00121	0.00272	0.00057	-0.00010	-0.00146	-0.00089	-0.00179
3	0.00157	0.00325	0.00422	0.00055	0.00053	-0.00014	-0.00078	-0.00098	-0.00261	-0.00077
4	0.00443	0.00561	0.00295	-0.00073	-0.00059	-0.00062	-0.00096	-0.00228	-0.00217	-0.00110
5	0.00815	0.00414	0.00183	-0.00040	-0.00394	-0.00096	-0.00357	0.00047	-0.00120	0.00750
6	0.00576	0.00443	0.00116	0.00003	-0.00103	-0.00435	-0.00245	-0.00131	0.00012	0.00046
7	0.00769	0.00202	0.00027	-0.00212	-0.00109	-0.00238	-0.00120	0.00027	0.00047	0.00065
8	0.00422	0.00051	-0.00072	-0.00143	-0.00284	-0.00099	-0.00126	0.00046	0.00065	0.00041
9	0.00205	-0.00132	-0.00127	-0.00350	-0.00118	-0.00139	0.00057	0.00073	0.00034	0.00024
10	-0.00083	-0.00135	-0.00533	0.00047	-0.00171	0.00078	0.00073	0.00034	0.00017	0.00018

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.00693	0.00189	-0.00097	-0.00482	-0.00334	-0.00327	-0.00088	-0.00036	0.00076	0.00451
2	0.00451	-0.00064	-0.00731	-0.00239	-0.00479	-0.00153	-0.00040	0.00182	0.00108	0.00266
3	0.00116	-0.00446	-0.00707	-0.00180	-0.00172	-0.00051	0.00066	0.00115	0.00380	0.00121
4	-0.00426	-0.00964	-0.00590	-0.00029	-0.00015	0.00022	0.00099	0.00328	0.00327	0.00175
5	-0.01307	-0.00904	-0.00504	-0.00101	0.00482	0.00080	0.00508	-0.00069	0.00188	-0.01108
6	-0.01385	-0.01130	-0.00455	-0.00176	0.00058	0.00604	0.00361	0.00204	-0.00002	-0.00059
7	-0.02080	-0.00931	-0.00353	0.00154	0.00076	0.00337	0.00184	-0.00024	-0.00059	-0.00088
8	-0.02032	-0.00821	-0.00209	0.00063	0.00381	0.00147	0.00205	-0.00058	-0.00088	-0.00055
9	-0.02090	-0.00611	-0.00113	0.00422	0.00159	0.00221	-0.00072	-0.00100	-0.00046	-0.00032
10	-0.01956	-0.00591	0.00551	-0.00122	0.00258	-0.00103	-0.00099	-0.00046	-0.00021	-0.00023

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.01527	0.00773	0.00492	0.00382	0.00383	0.00377	0.00369	0.00363	0.00337	0.00302
2	0.03974	0.01216	0.00627	0.00519	0.00499	0.00463	0.00434	0.00390	0.00377	0.00337
3	0.06697	0.01682	0.00884	0.00688	0.00601	0.00542	0.00471	0.00439	0.00394	0.00370
4	0.09629	0.02163	0.01185	0.00842	0.00708	0.00598	0.00531	0.00461	0.00438	0.00378
5	0.12577	0.02844	0.01453	0.01020	0.00778	0.00670	0.00552	0.00518	0.00447	0.00434
6	0.15825	0.03530	0.01741	0.01197	0.00891	0.00705	0.00621	0.00518	0.00473	0.00417
7	0.19465	0.04224	0.02019	0.01281	0.00992	0.00775	0.00646	0.00549	0.00479	0.00429
8	0.23041	0.04905	0.02194	0.01436	0.01035	0.00839	0.00657	0.00556	0.00486	0.00428
9	0.26626	0.05357	0.02433	0.01491	0.01128	0.00848	0.00690	0.00564	0.00471	0.00434
10	0.29289	0.05899	0.02537	0.01665	0.01148	0.00886	0.00701	0.00547	0.00470	0.00442

VELOCITY COMPONENTS ON WING PANELS DUE TO WING PANEL PRESSURE SINGULARITIES

AL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.04428	0.06407	0.07627	0.08756	0.09889	0.10954	0.11892	0.12658	0.13044	0.13887
2	0.04087	0.04965	0.05739	0.06507	0.07204	0.07790	0.08244	0.08571	0.08851	0.09081
3	0.03695	0.04188	0.04881	0.05506	0.06004	0.06362	0.06586	0.06746	0.06972	0.06957
4	0.03570	0.04009	0.04658	0.05150	0.05477	0.05644	0.05738	0.05821	0.05955	0.05846
5	0.03554	0.04053	0.04584	0.04931	0.05065	0.05101	0.05133	0.05213	0.05234	0.05110
6	0.03708	0.04165	0.04543	0.04689	0.04668	0.04641	0.04662	0.04714	0.04675	0.04528
7	0.04016	0.04296	0.04445	0.04367	0.04297	0.04262	0.04252	0.04290	0.04210	0.04074
8	0.04116	0.04176	0.04023	0.03862	0.03773	0.03743	0.03722	0.03730	0.03642	0.03520
9	0.04028	0.03648	0.03328	0.03176	0.03122	0.03075	0.03078	0.03053	0.02968	0.02901
10	0.03165	0.02482	0.02230	0.02177	0.02109	0.02092	0.02098	0.02059	0.02007	0.02297

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.12164	-0.17602	-0.20954	-0.24056	-0.27169	-0.30094	-0.32671	-0.34777	-0.35835	-0.38151
2	-0.11228	-0.13641	-0.15766	-0.17876	-0.19792	-0.21402	-0.22649	-0.23546	-0.24318	-0.24949
3	-0.10151	-0.11506	-0.13411	-0.15126	-0.16495	-0.17479	-0.18094	-0.18533	-0.19156	-0.19113
4	-0.09808	-0.11014	-0.12797	-0.14150	-0.15046	-0.15505	-0.15764	-0.15994	-0.16361	-0.16061
5	-0.09765	-0.11134	-0.12593	-0.13546	-0.13914	-0.14014	-0.14103	-0.14322	-0.14380	-0.14038
6	-0.10188	-0.11443	-0.12480	-0.12882	-0.12824	-0.12750	-0.12807	-0.12950	-0.12844	-0.12439
7	-0.11033	-0.11801	-0.12213	-0.11997	-0.11804	-0.11710	-0.11681	-0.11787	-0.11567	-0.11192
8	-0.11308	-0.11473	-0.11052	-0.10609	-0.10364	-0.10283	-0.10226	-0.10247	-0.10005	-0.09669
9	-0.11066	-0.10021	-0.09144	-0.08726	-0.08575	-0.08446	-0.08456	-0.08388	-0.08153	-0.07968
10	-0.08694	-0.06819	-0.06125	-0.05981	-0.05793	-0.05748	-0.05764	-0.05657	-0.05513	-0.06310

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.10984	-0.07760	-0.06156	-0.05443	-0.05120	-0.04924	-0.04797	-0.04714	-0.04646	-0.04558
2	-0.16378	-0.11190	-0.09223	-0.08507	-0.08171	-0.07951	-0.07809	-0.07692	-0.07639	-0.07553
3	-0.20307	-0.12239	-0.10037	-0.09241	-0.08847	-0.08613	-0.08434	-0.08331	-0.08248	-0.08182
4	-0.23865	-0.13268	-0.10886	-0.09956	-0.09525	-0.09245	-0.09074	-0.08937	-0.08876	-0.08780
5	-0.27398	-0.14507	-0.11728	-0.10721	-0.10191	-0.09920	-0.09701	-0.09601	-0.09494	-0.09448
6	-0.31242	-0.15795	-0.12640	-0.11536	-0.10950	-0.10605	-0.10423	-0.10255	-0.10175	-0.10089
7	-0.35905	-0.17539	-0.13989	-0.12703	-0.12141	-0.11768	-0.11543	-0.11382	-0.11277	-0.11204
8	-0.40530	-0.19299	-0.15263	-0.13967	-0.13298	-0.12949	-0.12672	-0.12508	-0.12407	-0.12332
9	-0.45461	-0.21126	-0.16895	-0.15423	-0.14794	-0.14363	-0.14111	-0.13928	-0.13805	-0.13751
10	-0.50068	-0.23639	-0.18984	-0.17587	-0.16808	-0.16396	-0.16120	-0.15913	-0.15800	-0.15759

VELOCITY COMPONENTS ON WING PANELS DUE TO BODY LINE SOURCES AND DOUBLETS

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.00718	0.01205	0.00798	0.00569	0.00421	0.00320	0.00252	0.00202	0.00173	0.00135
2	0.01477	0.00983	0.00685	0.00484	0.00366	0.00280	0.00222	0.00180	0.00156	0.00125
3	0.01180	0.00835	0.00577	0.00418	0.00315	0.00247	0.00197	0.00162	0.00141	0.00116
4	0.00997	0.00712	0.00491	0.00363	0.00276	0.00219	0.00177	0.00146	0.00128	0.00108
5	0.00846	0.00610	0.00424	0.00316	0.00244	0.00195	0.00159	0.00132	0.00117	0.00101
6	0.00723	0.00514	0.00369	0.00277	0.00218	0.00175	0.00144	0.00121	0.00107	0.00095

7	0.00605	0.00443	0.00322	0.00246	0.00194	0.00158	0.00131	0.00111	0.00099	0.00147
8	0.00519	0.00385	0.00283	0.00219	0.00175	0.00144	0.00120	0.00113	0.00206	0.00305
9	0.00448	0.00338	0.00251	0.00196	0.00158	0.00131	0.00110	0.00226	0.00378	0.00445
10	0.00390	0.00295	0.00224	0.00177	0.00144	0.00120	0.00154	0.00404	0.00511	0.00512

TRANSVERSE(V)

SPANWISE STATION CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.03370	0.00214	-0.00184	-0.00247	-0.00230	-0.00194	-0.00164	-0.00136	-0.00119	-0.00096
2	0.01903	0.00325	-0.00126	-0.00183	-0.00184	-0.00157	-0.00136	-0.00115	-0.00102	-0.00085
3	0.01991	0.00364	-0.00058	-0.00135	-0.00141	-0.00129	-0.00112	-0.00098	-0.00088	-0.00075
4	0.01981	0.00398	-0.00005	-0.00098	-0.00109	-0.00106	-0.00094	-0.00083	-0.00076	-0.00067
5	0.01972	0.00426	0.00031	-0.00064	-0.00085	-0.00085	-0.00079	-0.00071	-0.00066	-0.00060
6	0.01964	0.00467	0.00060	-0.00038	-0.00065	-0.00069	-0.00067	-0.00061	-0.00057	-0.00053
7	0.01979	0.00488	0.00084	-0.00017	-0.00048	-0.00056	-0.00056	-0.00053	-0.00050	-0.00134
8	0.01976	0.00503	0.00105	-0.00001	-0.00034	-0.00045	-0.00047	-0.00062	-0.00217	-0.00380
9	0.01971	0.00516	0.00121	0.00013	-0.00023	-0.00036	-0.00040	-0.00239	-0.00491	-0.00606
10	0.01968	0.00530	0.00133	0.00025	-0.00014	-0.00028	-0.00113	-0.00524	-0.00713	-0.00720

VERTICAL(W)

SPANWISE STATION CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.05356	0.02885	0.01562	0.00959	0.00635	0.00445	0.00326	0.00249	0.00207	0.00155
2	0.05844	0.02914	0.01537	0.00928	0.00612	0.00429	0.00316	0.00242	0.00203	0.00156
3	0.05955	0.02902	0.01499	0.00898	0.00591	0.00416	0.00308	0.00237	0.00199	0.00157
4	0.05993	0.02862	0.01459	0.00871	0.00574	0.00405	0.00301	0.00233	0.00196	0.00159
5	0.05968	0.02811	0.01422	0.00848	0.00561	0.00397	0.00297	0.00230	0.00194	0.00161
6	0.05909	0.02756	0.01390	0.00830	0.00550	0.00391	0.00293	0.00228	0.00193	0.00164
7	0.05834	0.02708	0.01364	0.00816	0.00542	0.00387	0.00290	0.00226	0.00191	0.00169
8	0.05763	0.02668	0.01343	0.00805	0.00537	0.00384	0.00288	0.00225	0.00195	0.00178
9	0.05701	0.02635	0.01328	0.00797	0.00532	0.00381	0.00287	0.00230	0.00200	0.00183
10	0.05648	0.02609	0.01316	0.00792	0.00529	0.00379	0.00288	0.00235	0.00199	0.00186

VELOCITY COMPONENTS ON WING PANELS DUE TO WING SOURCES

AXIAL(U)

SPANWISE STATION CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.03054	-0.02433	-0.01880	-0.01746	-0.01326	-0.00857	-0.00441	-0.00076	-0.00148	-0.00331
2	-0.00716	-0.00013	0.00337	0.00784	0.01289	0.01731	0.02137	0.02116	0.02006	0.01976
3	-0.00189	0.00375	0.00866	0.01475	0.01964	0.02390	0.02566	0.02337	0.02313	0.02314
4	-0.00060	0.00495	0.01303	0.01903	0.02368	0.02789	0.02409	0.02362	0.02363	0.02249
5	-0.00045	0.00850	0.01654	0.02209	0.02667	0.02357	0.02284	0.02278	0.02177	0.02115
6	0.00270	0.01161	0.01874	0.02411	0.02187	0.02069	0.02053	0.01941	0.01883	0.01855
7	0.00547	0.01297	0.01968	0.01900	0.01691	0.01658	0.01565	0.01475	0.01439	0.01431
8	0.00716	0.01395	0.01652	0.01237	0.01172	0.01114	0.00986	0.00933	0.00908	0.00913
9	0.00900	0.01548	0.00706	0.00593	0.00571	0.00393	0.00330	0.00293	0.00274	0.00307
10	0.01097	-0.00233	-0.00442	-0.00488	-0.00666	-0.00745	-0.00788	-0.00816	-0.00831	-0.00557

TRANSVERSE(V)

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10

CHORDWISE STATION										
1	0.02346	0.03708	0.03491	0.03503	0.02960	0.02265	0.01605	0.01001	0.01062	0.01346
2	-0.02849	-0.02351	-0.02451	-0.02979	-0.03719	-0.04422	-0.05097	-0.05142	-0.05010	-0.04997
3	-0.03063	-0.02966	-0.03436	-0.04316	-0.05102	-0.05823	-0.06183	-0.05896	-0.05886	-0.05917
4	-0.02439	-0.02819	-0.03974	-0.04951	-0.05761	-0.06514	-0.06024	-0.05998	-0.06022	-0.05877
5	-0.01502	-0.02915	-0.04278	-0.05283	-0.06136	-0.05790	-0.05738	-0.05766	-0.05636	-0.05561
6	-0.00956	-0.02668	-0.04278	-0.05337	-0.05185	-0.05088	-0.05112	-0.04977	-0.04905	-0.04882
7	-0.00345	-0.02458	-0.03956	-0.04162	-0.03967	-0.03983	-0.03886	-0.03773	-0.03731	-0.03742
8	0.00316	-0.02012	-0.03011	-0.02594	-0.02587	-0.02555	-0.02394	-0.02333	-0.02304	-0.02340
9	0.00622	-0.01726	-0.00930	-0.00904	-0.00947	-0.00721	-0.00649	-0.00607	-0.00586	-0.00692
10	0.01231	0.02030	0.02082	0.02039	0.02246	0.02335	0.02383	0.02413	0.02430	0.01669

VERTICAL(W)

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
CHORDWISE STATION										
1	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829
2	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022
3	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498
4	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695
5	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762
6	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197
7	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357
8	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034
9	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620
10	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180

VELOCITY COMPONENTS ON BODY PANELS DUE TO BODY PANEL PRESSURE SINGULARITIES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.00002	0.00007	0.00029	0.00331	-0.00316	-0.00028	-0.00008
2	-0.00007	0.00041	0.00175	0.00596	-0.01799	-0.00120	-0.00043
3	-0.00075	0.00162	0.00374	0.01165	-0.01958	-0.00805	0.00144
4	0.00540	0.00136	0.00600	0.01604	-0.01826	-0.00703	-0.00156
5	0.00414	0.00917	0.01301	0.02239	-0.01094	-0.00937	-0.00866
6	0.00424	0.00893	0.01529	0.02318	-0.01324	-0.01163	-0.01258
7	0.00589	0.00900	0.01484	0.02361	-0.01478	-0.01457	-0.01492
8	0.00783	0.00985	0.01494	0.02352	-0.01753	-0.01629	-0.01607
9	0.00879	0.01077	0.01559	0.02252	-0.01942	-0.01785	-0.01702
10	0.00884	0.01151	0.01568	0.02090	-0.01969	-0.01900	-0.01876
11	0.01182	0.01267	0.01188	0.00745	-0.01442	-0.01753	-0.02013
12	0.01419	0.01211	0.00602	-0.00350	-0.00994	-0.01286	-0.01572
13	0.01062	0.00867	0.00529	0.00061	-0.00414	-0.00758	-0.01012
14	0.00581	0.00495	0.00306	0.00037	-0.00211	-0.00413	-0.00514

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.00002	-0.00010	-0.00027	-0.00831	0.00668	0.00044	0.00009

2	-0.00042	-0.00039	-0.00116	-0.02402	0.04286	0.00088	0.00022
3	0.00086	-0.00135	-0.00265	-0.05905	0.07566	0.00417	0.00249
4	0.00231	0.00055	-0.00753	-0.10042	0.10674	0.00330	-0.00339
5	-0.00184	-0.00581	-0.02112	-0.15025	0.12628	0.00776	-0.00145
6	-0.00126	-0.00800	-0.03018	-0.19336	0.15515	0.01618	-0.00043
7	-0.00129	-0.00726	-0.03778	-0.23816	0.18512	0.02407	0.00244
8	-0.00027	-0.00809	-0.04792	-0.28545	0.21752	0.03308	0.00496
9	-0.00098	-0.01125	-0.05912	-0.33274	0.25057	0.04102	0.00676
10	-0.00079	-0.01335	-0.07054	-0.37709	0.28283	0.04992	0.00925
11	-0.00157	-0.01877	-0.09408	-0.41385	0.32006	0.06893	0.01348
12	-0.00414	-0.02956	-0.10614	-0.40678	0.33921	0.08108	0.01869
13	-0.01101	-0.04378	-0.11848	-0.40820	0.34575	0.08554	0.02071
14	-0.01466	-0.05306	-0.12696	-0.39831	0.32201	0.09194	0.02400

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.00001	0.00008	0.00053	-0.00098	-0.00417	0.00034	0.00002
2	0.00009	0.00030	-0.00063	-0.00538	-0.02292	0.00392	0.00005
3	0.00218	-0.00149	-0.00318	-0.00828	-0.04361	-0.00020	0.00796
4	-0.00558	-0.00133	-0.00344	-0.01355	-0.06235	-0.00549	0.01244
5	-0.00593	-0.00719	-0.01052	-0.01710	-0.08013	-0.01313	0.00733
6	-0.00390	-0.00671	-0.01328	-0.02503	-0.09844	-0.02137	-0.00024
7	-0.00313	-0.00835	-0.01863	-0.02786	-0.11666	-0.03129	-0.00918
8	-0.00285	-0.01025	-0.02304	-0.03318	-0.13676	-0.04149	-0.01868
9	-0.00177	-0.01110	-0.02775	-0.03870	-0.15860	-0.05377	-0.02879
10	-0.00097	-0.01352	-0.03311	-0.04362	-0.17958	-0.06586	-0.03936
11	-0.00539	-0.02245	-0.04488	-0.04904	-0.20256	-0.08883	-0.06217
12	-0.02059	-0.03750	-0.05441	-0.05121	-0.21182	-0.10348	-0.08000
13	-0.03932	-0.04872	-0.05797	-0.05069	-0.21617	-0.11197	-0.09096
14	-0.05382	-0.05936	-0.06540	-0.10456	-0.26210	-0.12472	-0.10475

VELOCITY COMPONENTS ON BODY PANELS DUE TO WING PANEL PRESSURE SINGULARITIES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.	0.	0.	0.01087	-0.00750	0.	0.
2	0.	0.	0.00706	0.01514	-0.01043	-0.00475	0.
3	0.00230	0.00563	0.00988	0.01652	-0.01137	-0.00657	-0.00479
4	0.00707	0.00861	0.01323	0.01700	-0.01214	-0.00930	-0.00698
5	0.00948	0.01187	0.01449	0.01719	-0.01234	-0.01039	-0.00880
6	0.01202	0.01273	0.01506	0.01774	-0.01276	-0.01106	-0.01017
7	0.01287	0.01398	0.01612	0.01908	-0.01370	-0.01171	-0.01055
8	0.01348	0.01465	0.01763	0.02003	-0.01449	-0.01265	-0.01106
9	0.01454	0.01619	0.01852	0.02017	-0.01462	-0.01343	-0.01210
10	0.01609	0.01710	0.01893	0.01807	-0.01323	-0.01389	-0.01311
11	0.01698	0.01712	0.01222	0.00497	-0.00425	-0.00953	-0.01282
12	0.01018	0.00904	0.00698	0.00266	-0.00232	-0.00537	-0.00645
13	0.00635	0.00573	0.00409	0.00152	-0.00135	-0.00335	-0.00410
14	0.00426	0.00368	0.00261	0.00098	-0.00082	-0.00209	-0.00269

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.	0.	0.	0.00891	-0.01359	0.	0.
2	0.	0.	0.00460	0.04374	-0.04699	-0.00673	0.
3	0.00217	0.00554	0.01373	0.08402	-0.08183	-0.01568	-0.00403
4	0.00252	0.00921	0.02602	0.12400	-0.11627	-0.02433	-0.00574
5	0.00404	0.01509	0.03760	0.16204	-0.14930	-0.03379	-0.00891
6	0.00577	0.02023	0.04921	0.19999	-0.18255	-0.04261	-0.01064
7	0.00700	0.02438	0.06059	0.23947	-0.21727	-0.05166	-0.01305
8	0.00857	0.02986	0.07379	0.28161	-0.25447	-0.06189	-0.01567
9	0.01057	0.03671	0.08730	0.32448	-0.29189	-0.07260	-0.01891
10	0.01247	0.04302	0.10077	0.36506	-0.32628	-0.08292	-0.02161
11	0.01554	0.05375	0.12085	0.40116	-0.35147	-0.09524	-0.02637
12	0.01668	0.05676	0.12699	0.40506	-0.35425	-0.09839	-0.02633
13	0.01778	0.05978	0.13049	0.40671	-0.35550	-0.10027	-0.02724
14	0.01841	0.06123	0.13212	0.39727	-0.32546	-0.09884	-0.02722

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.	-0.	-0.	-0.02172	-0.00996	-0.	-0.
2	-0.	-0.	-0.01182	-0.04304	-0.01053	-0.00545	-0.
3	-0.00285	-0.00826	-0.01899	-0.05346	-0.00337	-0.00605	-0.00354
4	-0.00757	-0.01189	-0.02519	-0.05680	0.00882	-0.00481	-0.00221
5	-0.00879	-0.01452	-0.02677	-0.05772	0.02331	0.00001	0.00089
6	-0.00910	-0.01374	-0.02676	-0.05864	0.03816	0.00613	0.00589
7	-0.00707	-0.01240	-0.02647	-0.06034	0.05345	0.01303	0.01219
8	-0.00452	-0.01040	-0.02615	-0.06084	0.07057	0.02018	0.01874
9	-0.00214	-0.00878	-0.02479	-0.05959	0.08941	0.02836	0.02546
10	0.00040	-0.00613	-0.02233	-0.05327	0.11080	0.03752	0.03315
11	0.00931	0.00349	-0.00260	-0.00954	0.15781	0.06340	0.05218
12	0.02748	0.02493	0.01888	0.01595	0.18209	0.08439	0.07336
13	0.04107	0.03908	0.03486	0.03337	0.19892	0.09955	0.08742
14	0.05461	0.05422	0.05495	0.10166	0.25671	0.11902	0.10301

VELOCITY COMPONENTS ON BODY PANELS DUE TO WING SOURCES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.	0.	0.	-0.00275	-0.00157	0.	0.
2	0.	0.	-0.00645	-0.01262	-0.01134	-0.00520	0.
3	0.	-0.00686	-0.00902	-0.00734	-0.00732	-0.00785	-0.00585
4	-0.01113	-0.00699	-0.00646	-0.00369	-0.00407	-0.00956	-0.01252
5	-0.01124	-0.01126	-0.00895	-0.00581	-0.00707	-0.00877	-0.01019
6	-0.00891	-0.00826	-0.00613	-0.00292	-0.00375	-0.00602	-0.00737
7	-0.00613	-0.00531	-0.00300	0.00021	-0.00054	-0.00297	-0.00441
8	-0.00328	-0.00224	0.00013	0.00300	0.00234	0.00008	-0.00138

9	-C.00025	0.00076	0.00290	0.00529	0.00473	0.00290	0.00167
10	0.00267	0.00351	0.00541	0.00753	0.00705	0.00542	0.00439
11	C.00632	0.00717	0.00881	0.00308	0.00388	0.00862	0.00773
12	C.00740	0.00597	0.00467	0.00365	0.00410	0.00548	0.00453
13	C.00208	0.00191	0.00159	0.00128	0.00131	0.00154	0.00169
14	C.00092	0.00084	0.00070	0.00059	0.00061	0.00071	0.00078

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	C.	0.	C.	-0.00015	-0.00106	0.	0.
2	C.	0.	-0.00215	-0.01927	-0.02232	-0.00479	0.
3	C.	-0.00460	-0.00837	-0.02398	-0.02413	-0.01141	-0.00699
4	-C.00260	-0.00750	-0.01101	-0.02218	-0.02128	-0.00715	-0.00056
5	-0.00052	0.00001	-0.00376	-0.01012	-0.00750	-0.00338	-0.00047
6	-0.00028	-0.00024	-0.00255	-0.00459	-0.00255	-0.00168	-0.00015
7	-C.00016	-0.00001	-0.00119	0.00100	0.00295	-0.00000	0.00020
8	-C.00004	0.00046	0.00043	0.00642	0.00817	0.00181	0.00071
9	C.00015	0.00109	0.00215	0.01104	0.01251	0.00342	0.00112
10	0.00032	0.00154	0.00361	0.01487	0.01607	0.00468	0.00134
11	C.00066	0.00205	0.00560	0.01433	0.01246	0.00665	0.00165
12	-0.00173	-0.00105	0.00067	0.00264	0.00152	-0.00122	-0.00010
13	C.00013	0.00054	0.00127	0.00194	0.00169	0.00090	0.00026
14	C.00015	0.00048	0.00086	0.00112	0.00099	0.00059	0.00019

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-C.	-0.	-0.	0.00415	-0.00212	-0.	-0.
2	-C.	-0.	0.00997	0.03008	-0.02056	-0.00664	-0.
3	-C.	0.01007	0.01697	0.02118	-0.01490	-0.01133	-0.00630
4	0.01365	0.01149	0.01425	0.01298	-0.00936	-0.01133	-0.01238
5	0.01435	0.01457	0.01285	0.00678	-0.00548	-0.00945	-0.01063
6	0.01206	0.01125	0.00839	0.00032	-0.00087	-0.00633	-0.00813
7	0.00897	0.00761	0.00357	-0.00594	0.00360	-0.00291	-0.00532
8	C.00555	0.00359	-0.00133	-0.01125	0.00744	0.00059	-0.00228
9	C.00174	-0.00049	-0.00573	-0.01520	0.01034	0.00380	0.00088
10	-C.00209	-0.00429	-0.00956	-0.01861	0.01287	0.00662	0.00377
11	-0.00716	-0.00946	-0.01471	-0.00827	0.00607	0.01034	0.00750
12	-C.00928	-0.00750	-0.00580	-0.00246	0.00212	0.00487	0.00492
13	-0.00328	-0.00306	-0.00239	-0.00094	0.00081	0.00182	0.00215
14	-0.00165	-0.00150	-0.00111	-0.00043	0.00035	0.00086	0.00106

PRESSURES, FORCES, AND MOMENTS ON ISOLATED BODY

CD = 0.00069 CL = 0.00174 CM = -0.00454

BODY PRESSURE COEFFICIENTS(CP)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000

0.	C.08817	0.09131	0.09691	0.10415	0.11166	0.11725	0.12010
0.7243	C.08817	0.09131	0.09691	0.10415	0.11166	0.11725	0.12010
1.4485	C.07725	0.08238	0.09153	0.10329	0.11541	0.12437	0.12892
2.1728	C.02570	0.02978	0.03705	0.04641	0.05604	0.06317	0.06679
2.8970	C.02653	0.03044	0.03740	0.04632	0.05547	0.06222	0.06564
3.6213	C.01477	0.01823	0.02440	0.03232	0.04046	0.04648	0.04953
4.3456	C.01613	0.01935	0.02508	0.03243	0.03998	0.04555	0.04838
5.0698	C.00977	0.01262	0.01770	0.02424	0.03098	0.03598	0.03852
5.7941	C.01068	0.01325	0.01784	0.02375	0.02985	0.03437	0.03666
6.5184	C.00486	0.00740	0.01191	0.01773	0.02373	0.02819	0.03045
7.2426	C.00336	0.00596	0.01059	0.01654	0.02267	0.02720	0.02949
7.9669	C.00038	0.00261	0.00660	0.01175	0.01709	0.02107	0.02310
8.6911	-C.00341	-0.00079	0.00388	0.00988	0.01605	0.02060	0.02291
9.4154	-C.00592	-0.00359	0.00059	0.00597	0.01155	0.01570	0.01781
10.1397	-C.00350	-0.00613	-0.00190	0.00355	0.00920	0.01339	0.01553
10.8639	-C.01222	-0.00985	-0.00560	-0.00011	0.00558	0.00981	0.01197
11.5882	-C.01405	-0.01201	-0.00834	-0.00355	0.00145	0.00521	0.00714
12.3124	-C.01758	-0.01553	-0.01183	-0.00702	-0.00198	0.00181	0.00376
13.0367	-C.02267	-0.02063	-0.01696	-0.01216	-0.00712	-0.00332	-0.00136
13.7610	-C.02481	-0.02285	-0.01932	-0.01470	-0.00982	-0.00612	-0.00422
14.5023	-C.03245	-0.03027	-0.02632	-0.02116	-0.01573	-0.01164	-0.00953
15.2436	-C.03748	-0.03555	-0.03202	-0.02735	-0.02235	-0.01852	-0.01653
15.9849	-C.05011	-0.04877	-0.04624	-0.04270	-0.03870	-0.03548	-0.03375
16.7262	-C.04067	-0.04015	-0.03905	-0.03729	-0.03502	-0.03303	-0.03191
17.4675	-C.03331	-0.03337	-0.03329	-0.03281	-0.03183	-0.03077	-0.03011
18.2088	-C.02774	-0.02817	-0.02875	-0.02912	-0.02904	-0.02864	-0.02832
18.9501	-C.02359	-0.02422	-0.02519	-0.02608	-0.02657	-0.02661	-0.02653
19.6914	-C.02054	-0.02126	-0.02240	-0.02356	-0.02436	-0.02468	-0.02475
20.4327	-C.01831	-0.01904	-0.02021	-0.02145	-0.02240	-0.02286	-0.02301
21.1740	-C.01669	-0.01738	-0.01848	-0.01969	-0.02065	-0.02116	-0.02136
21.9153	-C.01551	-0.01612	-0.01711	-0.01821	-0.01911	-0.01961	-0.01981
22.6566	-C.01463	-0.01515	-0.01601	-0.01696	-0.01776	-0.01821	-0.01840
23.3979	-C.01396	-0.01440	-0.01511	-0.01591	-0.01658	-0.01697	-0.01713
24.1392	-C.01343	-0.01378	-0.01436	-0.01501	-0.01556	-0.01588	-0.01602
24.8805	-C.01300	-0.01328	-0.01374	-0.01425	-0.01468	-0.01494	-0.01504
25.6218	-C.01262	-0.01284	-0.01320	-0.01360	-0.01393	-0.01412	-0.01420
26.3631	-C.01229	-0.01246	-0.01274	-0.01304	-0.01329	-0.01342	-0.01348
27.1044	-C.01199	-0.01212	-0.01233	-0.01256	-0.01274	-0.01283	-0.01286
27.8457	-C.01170	-0.01180	-0.01197	-0.01214	-0.01226	-0.01232	-0.01234
28.5870	-C.01144	-0.01152	-0.01164	-0.01177	-0.01186	-0.01189	-0.01190
29.3283	-C.01119	-0.01125	-0.01135	-0.01145	-0.01151	-0.01152	-0.01152
30.0696	-C.01350	-0.01354	-0.01360	-0.01364	-0.01363	-0.01358	-0.01355
30.8109	-C.02131	-0.02136	-0.02142	-0.02141	-0.02129	-0.02114	-0.02103
31.5522	-C.02868	-0.02888	-0.02917	-0.02939	-0.02944	-0.02936	-0.02927
32.2935	-C.03248	-0.03296	-0.03372	-0.03449	-0.03503	-0.03524	-0.03529
33.0348	-C.03483	-0.03567	-0.03702	-0.03850	-0.03968	-0.04031	-0.04055
33.7761	-C.04007	-0.04129	-0.04329	-0.04552	-0.04736	-0.04842	-0.04884
34.5174	-C.04021	-0.04192	-0.04476	-0.04800	-0.05081	-0.05252	-0.05326
35.2587	-C.02759	-0.02978	-0.03349	-0.03788	-0.04190	-0.04453	-0.04573
36.0000	-C.00848	-0.01084	-0.01492	-0.01990	-0.02470	-0.02801	-0.02961

INCREMENTAL PRESSURES, FORCES, AND MOMENTS ON BODY PANELS DUE TO WING

CD = 0.00144 CL = 0.01633 CM = -0.37392

BODY PANEL PRESSURE COEFFICIENT(CP)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.00004	-0.00014	-0.00058	-0.02290	0.02448	0.00055	0.00017
2	0.00013	-0.00081	-0.00472	-0.01715	0.07945	0.02241	0.00086
3	-0.00312	-0.00078	-0.00919	-0.04233	0.07506	0.04525	0.01851
4	-0.00270	-0.00596	-0.02542	-0.06006	0.06669	0.05203	0.04302
5	-0.00475	-0.01947	-0.03708	-0.06961	0.05796	0.05752	0.05690
6	-0.01460	-0.02664	-0.04840	-0.07972	0.05684	0.05800	0.06214
7	-0.02494	-0.03510	-0.05636	-0.09055	0.05553	0.05921	0.06165
8	-0.03541	-0.04422	-0.06626	-0.09933	0.05705	0.05843	0.05873
9	-0.04505	-0.05483	-0.07528	-0.10370	0.05626	0.05730	0.05645
10	-0.05365	-0.06348	-0.08217	-0.10150	0.04937	0.05535	0.05654
11	-0.06771	-0.07340	-0.06831	-0.03494	0.02823	0.03703	0.05173
12	-0.06141	-0.05368	-0.03680	-0.00701	0.01552	0.02532	0.03592
13	-0.03734	-0.03247	-0.02249	-0.00712	0.00807	0.01867	0.02539
14	-0.02173	-0.01884	-0.01282	-0.00389	0.00463	0.01104	0.01421

BODY PANEL SLOPE

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.06171	-0.04574	-0.01680	0.02125	0.06174	0.09272	0.10869
2	-0.07901	-0.06305	-0.03411	0.00402	0.04453	0.07542	0.09138
3	-0.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
4	-0.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
5	-0.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
6	-0.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
7	-0.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
8	-0.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
9	-0.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
10	-0.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
11	-0.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
12	-0.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
13	-0.08520	-0.06923	-0.04030	-0.00214	0.03839	0.06923	0.08520
14	-0.09941	-0.08345	-0.05451	-0.01629	0.02425	0.05502	0.07098

PRESSURES, FORCES, AND MOMENTS ON WING PANELS IN PRESENCE OF BODY

CD = 0.01362 CL = 0.18653 CM = -5.66621

SPANWISE CD DISTRIBUTION

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
	0.00096	0.00156	0.00155	0.00156	0.00152	0.00148	0.00146	0.00146	0.00074	0.00133

SPANWISE CL DISTRIBUTION

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
	0.00876	0.01621	0.01751	0.01865	0.01952	0.02024	0.02089	0.02141	0.01092	0.03242

UPPER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

SPANWISE STATION CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.04062	-0.11914	-0.15889	-0.19374	-0.23058	-0.27167	-0.30633	-0.33879	-0.34785	-0.36491
2	-0.10357	-0.13776	-0.17008	-0.18964	-0.22279	-0.24410	-0.26510	-0.27100	-0.27814	-0.28195
3	-0.10660	-0.13096	-0.15842	-0.17831	-0.20184	-0.21940	-0.22798	-0.22645	-0.22841	-0.23186
4	-0.11063	-0.13449	-0.16058	-0.17746	-0.19620	-0.20983	-0.20253	-0.20032	-0.20410	-0.20055
5	-0.11791	-0.14100	-0.16573	-0.18131	-0.18863	-0.18683	-0.17965	-0.19040	-0.18469	-0.19928
6	-0.12416	-0.15294	-0.17049	-0.18307	-0.17467	-0.16832	-0.16725	-0.16834	-0.16897	-0.16505
7	-0.14351	-0.15716	-0.17152	-0.16227	-0.15748	-0.15243	-0.15229	-0.15420	-0.15162	-0.14948
8	-0.14502	-0.15602	-0.15487	-0.13962	-0.13220	-0.13371	-0.12945	-0.13221	-0.13195	-0.13078
9	-0.14689	-0.14704	-0.12093	-0.10936	-0.11188	-0.10610	-0.10865	-0.11016	-0.11021	-0.11069
10	-0.13350	-0.09050	-0.07146	-0.08032	-0.07022	-0.07288	-0.07276	-0.07571	-0.07613	-0.08842

LOWER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

SPANWISE STATION CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.12965	0.12678	0.13518	0.14549	0.15352	0.15223	0.15333	0.14840	0.15667	0.17907
2	0.04340	0.04189	0.03750	0.04835	0.03876	0.03889	0.03311	0.04113	0.04530	0.05147
3	0.02974	0.02315	0.02102	0.02466	0.01813	0.01248	0.01196	0.02122	0.02914	0.02409
4	0.02546	0.01723	0.01473	0.01614	0.00790	-0.00126	0.01189	0.01836	0.01969	0.01924
5	0.02182	0.01700	0.01104	0.00764	0.00722	0.00868	0.01879	0.00935	0.01719	-0.00577
6	0.02732	0.01412	0.00997	0.00154	0.01078	0.02315	0.01942	0.02041	0.01801	0.01632
7	0.02445	0.02143	0.01130	0.01908	0.02179	0.02627	0.02601	0.02545	0.02505	0.02173
8	0.03244	0.02307	0.01825	0.02922	0.03404	0.03104	0.03509	0.03225	0.02880	0.02496
9	0.03288	0.01740	0.03400	0.04071	0.03554	0.04009	0.03737	0.03462	0.03094	0.02759
10	0.02045	0.04030	0.05057	0.03863	0.04679	0.04319	0.04358	0.03871	0.03604	0.03488

WING PANEL PRESSURE DIFFERENCE(CL)

SPANWISE STATION CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.17710	0.25627	0.30508	0.35024	0.39556	0.43316	0.47567	0.50634	0.52174	0.55546
2	0.16347	0.19861	0.22955	0.26027	0.28817	0.31160	0.32975	0.34282	0.35405	0.36325
3	0.14779	0.16752	0.19525	0.22022	0.24016	0.25448	0.26343	0.26983	0.27890	0.27828
4	0.14280	0.16036	0.18632	0.20602	0.21907	0.22575	0.22951	0.23286	0.23821	0.23385
5	0.14218	0.16211	0.18335	0.19723	0.20258	0.20404	0.20533	0.20853	0.20937	0.20439
6	0.14834	0.16661	0.18170	0.18756	0.18671	0.18564	0.18647	0.18855	0.18701	0.18112
7	0.16063	0.17182	0.17781	0.17467	0.17186	0.17050	0.17007	0.17161	0.16841	0.16296
8	0.16464	0.16704	0.16091	0.15447	0.15090	0.14972	0.14889	0.14920	0.14567	0.14078
9	0.16111	0.14590	0.13313	0.12705	0.12486	0.12298	0.12313	0.12213	0.11871	0.11603
10	0.12658	0.09929	0.08918	0.08709	0.08435	0.08369	0.08393	0.08237	0.08027	0.09189

UPPER SURFACE WING PANEL SLOPE

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10

CHORDWISE STATION										
1	0.06727	0.06727	0.06727	0.06727	0.06727	0.06727	0.06727	0.06727	0.06727	0.06727
2	-0.02038	-0.02038	-0.02038	-0.02038	-0.02038	-0.02038	-0.02038	-0.02038	-0.02038	-0.02038
3	-0.05157	-0.05157	-0.05157	-0.05157	-0.05157	-0.05157	-0.05157	-0.05157	-0.05157	-0.05157
4	-0.07548	-0.07548	-0.07548	-0.07548	-0.07548	-0.07548	-0.07548	-0.07548	-0.07548	-0.07548
5	-0.09614	-0.09614	-0.09614	-0.09614	-0.09614	-0.09614	-0.09614	-0.09614	-0.09614	-0.09614
6	-0.11706	-0.11706	-0.11706	-0.11706	-0.11706	-0.11706	-0.11706	-0.11706	-0.11706	-0.11706
7	-0.13963	-0.13963	-0.13963	-0.13963	-0.13963	-0.13963	-0.13963	-0.13963	-0.13963	-0.13963
8	-0.15761	-0.15761	-0.15761	-0.15761	-0.15761	-0.15761	-0.15761	-0.15761	-0.15761	-0.15761
9	-0.17754	-0.17754	-0.17754	-0.17754	-0.17754	-0.17754	-0.17754	-0.17754	-0.17754	-0.17754
10	-0.20311	-0.20311	-0.20311	-0.20311	-0.20311	-0.20311	-0.20311	-0.20311	-0.20311	-0.20311

LOWER SURFACE WING PANEL SLOPE

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
CHORDWISE STATION										
1	-0.14931	-0.14931	-0.14931	-0.14931	-0.14931	-0.14931	-0.14931	-0.14931	-0.14931	-0.14931
2	-0.12082	-0.12082	-0.12082	-0.12082	-0.12082	-0.12082	-0.12082	-0.12082	-0.12082	-0.12082
3	-0.10153	-0.10153	-0.10153	-0.10153	-0.10153	-0.10153	-0.10153	-0.10153	-0.10153	-0.10153
4	-0.08937	-0.08937	-0.08937	-0.08937	-0.08937	-0.08937	-0.08937	-0.08937	-0.08937	-0.08937
5	-0.08090	-0.08090	-0.08090	-0.08090	-0.08090	-0.08090	-0.08090	-0.08090	-0.08090	-0.08090
6	-0.07312	-0.07312	-0.07312	-0.07312	-0.07312	-0.07312	-0.07312	-0.07312	-0.07312	-0.07312
7	-0.07250	-0.07250	-0.07250	-0.07250	-0.07250	-0.07250	-0.07250	-0.07250	-0.07250	-0.07250
8	-0.07692	-0.07692	-0.07692	-0.07692	-0.07692	-0.07692	-0.07692	-0.07692	-0.07692	-0.07692
9	-0.08514	-0.08514	-0.08514	-0.08514	-0.08514	-0.08514	-0.08514	-0.08514	-0.08514	-0.08514
10	-0.09951	-0.09951	-0.09951	-0.09951	-0.09951	-0.09951	-0.09951	-0.09951	-0.09951	-0.09951

FORCES AND MOMENTS ON WING-BODY COMBINATION

CD = 0.01572 CL = 0.20496 CM = -6.04443

TR-805 MIN DRAG CL BAR = .1

DESCRIPTION OF CASE REQUESTED

SYMMETRICAL CONFIGURATION - PANELS LOCATED ON BOTH SIDES OF X-Z PLANE(SYM = 1.)

CASE = 3. OPTIMIZE WING SHAPE

CPCALC = 0. LINEAR CP

POLAR = 0. POLARS NOT REQUESTED

THICK = 1. WING THICKNESS PRESSURES TO BE ADDED

VOUT = 1. VELOCITY COMPONENTS TO BE PRINTED

WING REFERENCE AREA = 89.3750

POINT ABOUT WHICH THE MOMENTS ARE TO BE COMPUTED
X-COORDINATE = 0.
Z-COORDINATE = 0.

MACH NUMBER = 1.8000

TR-805 BODY CAMBER 6A

INCLINATION OF BODY AXIS WITH RESPECT TO DEFINING AXIS = 0. DEG.

ANGLE OF ATTACK WITH RESPECT TO BODY AXIS = -0. DEG.

HEIGHT OF WING PLANE ABOVE BODY AXIS = -0.2500

WING OPTIMIZED FOR CL BAR = 0.1000

VELOCITY COMPONENTS ON BODY DUE TO BODY LINE SOURCES AND DOUBLETS

AXIAL(U)

THETA(DEG.) X	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
0.	-0.07462	-0.07237	-0.06830	-0.06293	-0.05723	-0.05289	-0.05065
0.7243	-0.07462	-0.07237	-0.06830	-0.06293	-0.05723	-0.05289	-0.05065
1.4485	-0.06806	-0.06705	-0.06520	-0.06276	-0.06018	-0.05821	-0.05719
2.1728	-0.03481	-0.03405	-0.03266	-0.03082	-0.02887	-0.02739	-0.02662
2.8970	-0.03290	-0.03244	-0.03161	-0.03051	-0.02934	-0.02845	-0.02799
3.6213	-0.02474	-0.02432	-0.02358	-0.02259	-0.02154	-0.02075	-0.02033
4.3456	-0.02436	-0.02400	-0.02336	-0.02251	-0.02160	-0.02092	-0.02056
5.0698	-0.01999	-0.01958	-0.01885	-0.01788	-0.01685	-0.01607	-0.01566
5.7941	-0.01970	-0.01927	-0.01850	-0.01748	-0.01639	-0.01556	-0.01514
6.5184	-0.01594	-0.01558	-0.01493	-0.01408	-0.01317	-0.01248	-0.01212
7.2426	-0.01454	-0.01431	-0.01390	-0.01335	-0.01278	-0.01233	-0.01211

7.9669	-0.01235	-0.01203	-0.01144	-0.01067	-0.00985	-0.00923	-0.00891
8.6911	-0.00986	-0.00981	-0.00973	-0.00962	-0.00951	-0.00942	-0.00938
9.4154	-0.00793	-0.00789	-0.00772	-0.00749	-0.00724	-0.00706	-0.00696
10.1397	-0.00613	-0.00614	-0.00615	-0.00617	-0.00619	-0.00620	-0.00621
10.8639	-0.00371	-0.00380	-0.00398	-0.00422	-0.00446	-0.00465	-0.00475
11.5882	-0.00220	-0.00224	-0.00229	-0.00237	-0.00246	-0.00252	-0.00255
12.3124	0.00013	-0.00000	-0.00023	-0.00054	-0.00087	-0.00111	-0.00124
13.0367	0.00326	0.00305	0.00266	0.00215	0.00161	0.00120	0.00098
13.7610	0.00492	0.00464	0.00412	0.00344	0.00271	0.00216	0.00187
14.5023	0.00929	0.00880	0.00790	0.00672	0.00547	0.00451	0.00402
15.2436	0.01245	0.01194	0.01102	0.00980	0.00851	0.00752	0.00701
15.9849	0.01952	0.01914	0.01846	0.01756	0.01660	0.01588	0.01550
16.7262	0.01593	0.01568	0.01521	0.01459	0.01394	0.01344	0.01318
17.4675	0.01305	0.01289	0.01260	0.01221	0.01180	0.01149	0.01132
18.2088	0.01078	0.01069	0.01053	0.01031	0.01008	0.00991	0.00982
18.9501	0.00898	0.00894	0.00888	0.00879	0.00869	0.00862	0.00858
19.6914	0.00757	0.00756	0.00756	0.00756	0.00755	0.00755	0.00754
20.4327	0.00645	0.00647	0.00650	0.00655	0.00660	0.00664	0.00666
21.1740	0.00555	0.00559	0.00564	0.00572	0.00580	0.00586	0.00590
21.9153	0.00484	0.00488	0.00494	0.00503	0.00513	0.00520	0.00523
22.6566	0.00426	0.00430	0.00437	0.00446	0.00455	0.00462	0.00466
23.3979	0.00379	0.00383	0.00389	0.00397	0.00406	0.00412	0.00416
24.1392	0.00340	0.00343	0.00349	0.00356	0.00363	0.00369	0.00372
24.8805	0.00308	0.00310	0.00315	0.00320	0.00327	0.00331	0.00334
25.6218	0.00280	0.00282	0.00285	0.00290	0.00295	0.00299	0.00301
26.3631	0.00256	0.00257	0.00260	0.00264	0.00267	0.00270	0.00272
27.1044	0.00235	0.00236	0.00238	0.00241	0.00243	0.00246	0.00247
27.8457	0.00216	0.00217	0.00218	0.00220	0.00223	0.00224	0.00225
28.5870	0.00200	0.00200	0.00201	0.00203	0.00204	0.00205	0.00206
29.3283	0.00185	0.00185	0.00186	0.00187	0.00188	0.00189	0.00189
30.0696	0.00295	0.00295	0.00295	0.00296	0.00296	0.00297	0.00297
30.8109	0.00678	0.00678	0.00678	0.00678	0.00679	0.00679	0.00679
31.5522	0.01065	0.01065	0.01065	0.01065	0.01065	0.01065	0.01066
32.2935	0.01303	0.01303	0.01303	0.01303	0.01303	0.01303	0.01303
33.0348	0.01481	0.01481	0.01481	0.01481	0.01482	0.01482	0.01482
33.7761	0.01799	0.01799	0.01799	0.01799	0.01799	0.01799	0.01799
34.5174	0.01888	0.01888	0.01888	0.01888	0.01888	0.01888	0.01888
35.2587	0.01381	0.01381	0.01381	0.01381	0.01381	0.01381	0.01381
36.0000	0.00524	0.00524	0.00524	0.00524	0.00524	0.00524	0.00524

RADIAL(VR)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
X							
0.	0.20701	0.19983	0.18683	0.16969	0.15148	0.13762	0.13044
0.7243	0.20701	0.19983	0.18683	0.16969	0.15148	0.13762	0.13044
1.4485	0.19200	0.18764	0.17973	0.16930	0.15823	0.14979	0.14543
2.1728	0.13935	0.13541	0.12825	0.11882	0.10879	0.10117	0.09722
2.8970	0.13125	0.12802	0.12217	0.11445	0.10625	0.10001	0.09678
3.6213	0.11497	0.11198	0.10656	0.09942	0.09183	0.08605	0.08307
4.3456	0.11102	0.10828	0.10331	0.09675	0.08979	0.08449	0.08174
5.0698	0.10179	0.09905	0.09408	0.08753	0.08056	0.07526	0.07252
5.7941	0.09920	0.09646	0.09149	0.08493	0.07797	0.07268	0.06993
6.5184	0.09157	0.08895	0.08419	0.07792	0.07126	0.06619	0.06356
7.2426	0.08744	0.08505	0.08074	0.07504	0.06900	0.06439	0.06201
7.9669	0.08225	0.07975	0.07521	0.06923	0.06287	0.05803	0.05553
8.6911	0.07640	0.07438	0.07072	0.06590	0.06078	0.05688	0.05486

9.4154	C.07140	0.06939	0.06573	0.06092	0.05580	0.05191	0.04990
10.1397	C.06635	0.06458	0.06138	0.05715	0.05266	0.04924	0.04747
10.8639	C.06027	0.05875	0.05599	0.05236	0.C4850	0.04556	0.04404
11.5882	C.05552	0.05400	0.05125	0.04762	0.04377	0.04084	0.03932
12.3124	C.04939	0.04812	0.04582	0.04279	0.03957	0.03712	0.03585
13.0367	C.04171	0.04069	0.03885	0.03641	0.03383	0.03186	0.03084
13.7610	C.03603	0.03526	0.03387	0.03204	0.03009	0.02861	0.02784
14.5023	C.02561	0.02535	0.02488	0.02426	0.02360	0.02310	0.02285
15.2436	C.01636	0.01635	0.01633	0.01631	0.01629	0.01628	0.01627
15.9849	C.00000	0.0000C	0.00000	0.00000	0.00000	0.00000	0.00000
16.7262	C.00000	0.0000C	0.00000	0.00000	0.00000	0.00000	0.00000
17.4675	C.00000	0.0000C	0.00000	0.00000	0.00000	0.00000	0.00000
18.2088	C.00000	0.0000C	0.00000	0.00000	0.00000	0.00000	0.00000
18.9501	C.00000	0.0000C	0.00000	0.00000	0.00000	0.00000	0.00000
19.6914	C.00000	0.0000C	0.00000	0.00000	0.00000	0.00000	0.00000
20.4327	C.00000	0.0000C	0.00000	0.00000	0.00000	0.00000	0.00000
21.1740	C.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
21.9153	C.00000	0.0000C	0.00000	0.00000	0.00000	0.00000	0.00000
22.6566	C.00000	0.0000C	0.00000	0.00000	0.00000	0.00000	0.00000
23.3979	-C.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
24.1392	C.00000	0.0000C	0.00000	0.00000	0.00000	0.00000	0.00000
24.8805	C.00000	0.0000C	0.00000	0.00000	0.00000	0.00000	0.00000
25.6218	-C.00000	-0.0000C	0.00000	0.00000	0.00000	0.00000	0.00000
26.3631	C.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
27.1044	C.00000	0.0000C	0.00000	0.00000	0.00000	0.00000	0.00000
27.8457	-C.00000	0.0000C	-0.00000	-0.00000	-0.00000	0.00000	0.00000
28.5870	-C.00000	-0.0000C	-0.00000	-0.00000	0.00000	0.00000	0.00000
29.3283	-C.00000	-0.0000C	-0.00000	-0.00000	-0.00000	-0.00000	-0.00000
30.0696	-C.00209	-0.00209	-0.00209	-0.00209	-0.00209	-0.00209	-0.00209
30.8109	-C.00915	-0.00915	-0.00915	-0.00915	-0.00915	-0.00915	-0.00915
31.5522	-C.01740	-0.0174C	-0.01740	-0.01740	-0.01740	-0.01740	-0.01740
32.2935	-C.02427	-0.02427	-0.02427	-0.02427	-0.02427	-0.02427	-0.02427
33.0348	-C.03092	-0.03092	-0.03092	-0.03092	-0.03092	-0.03092	-0.03092
33.7761	-C.04072	-0.04072	-0.04072	-0.04072	-0.04072	-0.04072	-0.04072
34.5174	-C.04786	-0.04786	-0.04786	-0.04786	-0.04786	-0.04786	-0.04786
35.2587	-C.04502	-0.04502	-0.04502	-0.04502	-0.04502	-0.04502	-0.04502
36.0000	-C.03418	-0.03418	-0.03418	-0.03418	-0.03418	-0.03418	-0.03418

TANGENTIAL(VT)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
X							
0.	C.00628	0.01766	0.02572	0.02899	0.02605	0.01766	0.00628
0.7243	C.00628	0.01766	0.02572	0.02899	0.02605	0.01766	0.00628
1.4485	C.00448	0.01261	0.01837	0.02070	0.01860	0.01261	0.00448
2.1728	C.00425	0.01194	0.01740	0.01961	0.01762	0.01194	0.00425
2.8970	0.00374	0.01051	0.01531	0.01725	0.01550	0.01051	0.00374
3.6213	C.00342	0.00961	0.01400	0.01578	0.01417	0.00961	0.00342
4.3456	C.00314	0.00884	0.01289	0.01452	0.01305	0.00884	0.00314
5.0698	C.00303	0.00852	0.01241	0.01399	0.01256	0.00852	0.00303
5.7941	0.00298	0.00839	0.01222	0.01378	0.01237	0.00839	0.00298
6.5184	C.00295	0.0083C	0.01209	0.01363	0.01225	0.00830	0.00295
7.2426	C.00285	0.00801	0.01167	0.01315	0.01181	0.00801	0.00285
7.9669	C.00281	0.00789	0.01150	0.01296	0.01164	0.00789	0.00281
8.6911	C.00265	0.00746	0.01087	0.01225	0.01101	0.00746	0.00265
9.4154	C.00250	0.00704	0.01026	0.01156	0.01039	0.00704	0.00250
10.1397	C.00234	0.00657	0.00957	0.01079	0.00969	0.00657	0.00234

10.8639	C.00213	0.00600	0.00874	0.00985	0.00885	0.00600	[illegible]
11.5852	C.00197	0.00554	0.00807	0.00909	0.00817	0.00554	0.00197
12.3124	C.00179	0.00505	0.00735	0.00829	0.00745	0.00505	0.00179
13.0367	C.00159	0.00447	0.00651	0.00733	0.00659	0.00447	0.00159
13.7610	C.00135	0.00379	0.00553	0.00623	0.00560	0.00379	0.00135
14.5023	C.00101	0.00285	0.00415	0.00467	0.00420	0.00285	0.00101
15.2436	C.00064	0.00180	0.00262	0.00296	0.00266	0.00180	0.00064
15.9849	C.00033	0.00094	0.00137	0.00154	0.00138	0.00094	0.00033
16.7262	C.00011	0.00030	0.00044	0.00050	0.00045	0.00030	0.00011
17.4675	-C.00004	-0.00012	-0.00018	-0.00020	-0.00018	-0.00012	-0.00004
18.2088	-C.00014	-0.00038	-0.00056	-0.00063	-0.00057	-0.00038	-0.00014
18.9501	-C.00018	-0.00052	-0.00076	-0.00085	-0.00077	-0.00052	-0.00018
19.6914	-C.00020	-0.00057	-0.00082	-0.00093	-0.00084	-0.00057	-0.00020
20.4327	-C.00020	-0.00055	-0.00081	-0.00091	-0.00082	-0.00055	-0.00020
21.1740	-C.00018	-0.00051	-0.00074	-0.00083	-0.00075	-0.00051	-0.00018
21.9153	-C.00016	-0.00044	-0.00064	-0.00072	-0.00065	-0.00044	-0.00016
22.6566	-C.00013	-0.00037	-0.00054	-0.00061	-0.00054	-0.00037	-0.00013
23.3979	-C.00011	-0.00030	-0.00044	-0.00049	-0.00044	-0.00030	-0.00011
24.1392	-C.00008	-0.00024	-0.00034	-0.00039	-0.00035	-0.00024	-0.00008
24.8805	-C.00006	-0.00018	-0.00026	-0.00030	-0.00027	-0.00018	-0.00006
25.6218	-C.00005	-C.00014	-0.00020	-0.00022	-0.00020	-0.00014	-0.00005
26.3631	-C.00004	-0.00010	-0.00015	-0.00016	-0.00015	-0.00010	-0.00004
27.1044	-C.00003	-0.00007	-C.00011	-0.00012	-0.00011	-0.00007	-0.00003
27.8457	-C.00002	-0.00005	-0.00008	-0.00009	-0.00008	-0.00005	-0.00002
28.5870	-C.00001	-0.00004	-0.00005	-0.00006	-0.00005	-0.00004	-0.00001
29.3283	-C.00001	-0.00003	-0.00004	-0.00004	-0.00004	-0.00003	-0.00001
30.0696	-C.00001	-0.00002	-0.00003	-0.00003	-0.00003	-0.00002	-0.00001
30.8109	-C.00001	-0.00002	-0.00002	-0.00003	-0.00002	-0.00002	-0.00001
31.5522	-C.00000	-0.00001	-0.00002	-0.00002	-0.00002	-0.00001	-0.00000
32.2935	-C.00000	-0.00001	-0.00002	-0.00002	-0.00002	-0.00001	-0.00000
33.0348	-C.00000	-0.00001	-0.00002	-0.00002	-0.00002	-0.00001	-0.00000
33.7761	-C.00000	-0.00001	-0.00002	-0.00002	-0.00002	-0.00001	-0.00000
34.5174	[illegible]000	-0.00001	-0.00002	-0.00002	-0.00002	-0.00001	-0.00000
35.2587	[illegible]00000	-0.00001	-0.00002	-0.00002	-0.00002	-0.00001	-0.00000
36.0000	-C.00000	-0.00001	-0.00002	-0.00002	-0.00002	-0.00001	-0.00000

VELOCITY COMPONENTS ON WING PANELS DUE TO BODY PANEL PRESSURE SINGULARITIES

AXIAL(U)

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
CHORDWISE STATION										
1	-C.00626	-0.00149	0.00016	0.00443	0.00279	0.00219	0.00025	-0.00049	-0.00095	-0.00284
2	-C.00276	0.00046	0.00380	0.00186	0.00277	0.00054	-0.00041	-0.00144	-0.00098	-0.00163
3	C.00051	0.00310	0.00464	0.00038	0.00019	-0.00035	-0.00109	-0.00084	-0.00242	-0.00065
4	C.00367	0.00584	0.00408	-0.00119	-0.00116	-0.00094	-0.00109	-0.00186	-0.00176	-0.00077
5	C.00817	0.00510	0.00269	-0.00104	-0.00383	-0.00114	-0.00329	0.00053	-0.00080	0.00609
6	C.00673	0.00507	0.00135	-0.00060	-0.00118	-0.00388	-0.00207	-0.00089	0.00036	0.00049
7	C.00342	0.00205	-0.00005	-0.00207	-0.00113	-0.00216	-0.00078	0.00052	0.00050	0.00052
8	C.00435	0.00019	-0.00124	-0.00126	-0.00254	-0.00042	-0.00075	0.00050	0.00048	0.00019
9	C.00179	-0.00162	-0.00160	-0.00283	-0.00096	-0.00097	0.00060	0.00051	0.00017	0.00014
10	-C.00111	-0.00164	-0.00505	0.00060	-0.00123	0.00078	0.00059	0.00017	0.00010	0.00013

TRANSVERSE(V)

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10

CHORDWISE STATION										
1	0.00984	0.00545	[illegible]	-0.00538	-0.00445	-0.00379	0.00068	0.00035	0.00114	0.00415
2	0.00507	0.00141	-0.00[illegible]	-0.00305	-0.00473	-0.00144	0.00010	0.00183	0.00125	0.00244
3	[illegible]	-0.00274	-0.00705	-0.00131	-0.00113	-0.00015	0.00118	0.00100	0.00354	0.00103
4	0.00020	-0.00829	-0.00700	0.00059	0.00078	0.00078	0.00125	0.00269	0.00267	0.00124
5	-0.00578	-0.00875	-0.00583	0.00014	0.00477	0.00116	0.00472	-0.00077	0.00128	-0.00901
6	-0.01065	-0.01053	-0.00437	-0.00061	0.00094	0.00542	0.00307	0.00142	-0.00042	-0.00066
7	-0.01725	-0.00764	-0.00258	0.00171	0.00098	0.00311	0.00123	-0.00064	-0.00067	-0.00073
8	-0.01508	-0.00628	-0.00080	0.00064	0.00347	0.00064	0.00126	-0.00067	-0.00066	-0.00025
9	-0.01527	-0.00416	-0.00009	0.00346	0.00134	0.00157	-0.00080	-0.00071	-0.00022	-0.00018
10	-0.01490	-0.00398	0.00563	-0.00124	0.00189	-0.00107	-0.00082	-0.00022	-0.00012	-0.00016

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.00059	0.00237	0.00216	0.00229	0.00249	0.00236	0.00231	0.00235	0.00219	0.00200
2	0.00867	0.00491	0.00334	0.00328	0.00316	0.00293	0.00278	0.00254	0.00248	0.00224
3	0.02315	0.00833	0.00545	0.00433	0.00376	0.00347	0.00305	0.00290	0.00261	0.00244
4	0.04325	0.01252	0.00770	0.00521	0.00449	0.00386	0.00348	0.00308	0.00291	0.00247
5	0.06737	0.01792	0.00935	0.00650	0.00500	0.00436	0.00366	0.00346	0.00294	0.00284
6	0.09493	0.02282	0.01134	0.00780	0.00580	0.00463	0.00413	0.00339	0.00306	0.00265
7	0.12467	0.02778	0.01324	0.00833	0.00654	0.00513	0.00425	0.00355	0.00305	0.00270
8	0.15250	0.03249	0.01433	0.00944	0.00685	0.00554	0.00424	0.00353	0.00306	0.00267
9	0.17914	0.03537	0.01591	0.00986	0.00741	0.00551	0.00440	0.00355	0.00294	0.00270
10	0.19696	0.03884	0.01669	0.01104	0.00747	0.00567	0.00443	0.00342	0.00292	0.00275

VELOCITY COMPONENTS ON WING PANELS DUE TO WING PANEL PRESSURE SINGULARITIES

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.00827	0.03055	0.01831	0.01469	0.01458	0.01460	0.01470	0.01622	-0.02928	0.02997
2	0.01575	0.04020	0.02562	0.02265	0.02263	0.02288	0.02305	0.02574	-0.02353	0.04219
3	0.02160	0.04196	0.02928	0.02794	0.02812	0.02856	0.02866	0.03498	-0.00426	0.04434
4	0.02552	0.04002	0.03132	0.03159	0.03206	0.03250	0.03345	0.04119	0.01476	0.04050
5	0.02887	0.03618	0.03261	0.03410	0.03467	0.03527	0.03744	0.04163	0.03040	0.03402
6	0.03192	0.03121	0.03345	0.03539	0.03602	0.03710	0.03854	0.03789	0.04122	0.02696
7	0.03421	0.02590	0.03336	0.03514	0.03628	0.03659	0.03596	0.03192	0.04802	0.02005
8	0.03436	0.02041	0.03140	0.03353	0.03421	0.03297	0.03038	0.02485	0.05156	0.01301
9	0.03146	0.01446	0.02719	0.02923	0.02859	0.02649	0.02302	0.01726	0.05107	0.00506
10	0.02291	0.00784	0.01905	0.01991	0.01851	0.01689	0.01408	0.00917	0.04161	-0.00763

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.02271	-0.08394	-0.05029	-0.04036	-0.04005	-0.04012	-0.04038	-0.04456	0.08044	-0.08235
2	-0.04327	-0.11045	-0.07038	-0.06222	-0.06218	-0.06286	-0.06334	-0.07072	0.06465	-0.11592
3	-0.05934	-0.11529	-0.08045	-0.07675	-0.07724	-0.07848	-0.07875	-0.09610	0.01171	-0.12181
4	-0.07013	-0.10994	-0.08604	-0.08680	-0.08809	-0.08929	-0.09189	-0.11316	-0.04056	-0.11126
5	-0.07932	-0.09941	-0.08959	-0.09369	-0.09524	-0.09690	-0.10285	-0.11437	-0.08353	-0.09346
6	-0.08771	-0.08574	-0.09190	-0.09723	-0.09896	-0.10192	-0.10589	-0.10410	-0.11324	-0.07408
7	-0.09398	-0.07116	-0.09165	-0.09655	-0.09967	-0.10053	-0.09879	-0.08770	-0.13192	-0.05507

8	-C.09439	-0.05606	-0.08628	-0.09213	-0.09398	-0.09058	-0.08346	-0.06827	-0.14165	-0.03575
9	-C.08644	-0.03973	-0.07471	-0.08032	-0.07855	-0.07278	-0.06323	-0.04742	-0.14032	-0.01390
10	-C.06294	-0.02153	-0.05234	-0.05470	-0.05085	-0.04639	-0.03868	-0.02518	-0.11432	0.02099

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-C.01254	-0.02109	0.03394	0.03997	0.04106	0.04493	0.04903	0.05209	0.17539	0.04483
2	-C.03899	-0.07475	0.01681	0.02511	0.02697	0.03151	0.03591	0.03893	0.30268	-0.00283
3	-C.06270	-0.11620	0.00075	0.00920	0.01258	0.01735	0.02358	0.00792	0.39447	-0.04207
4	-C.08554	-0.14787	-0.01537	-0.00700	-0.00239	0.00291	0.00940	-0.03867	0.44569	-0.06977
5	-C.11208	-0.17011	-0.03210	-0.02376	-0.01792	-0.01253	-0.00941	-0.07972	0.45241	-0.08710
6	-C.14564	-0.18225	-0.05065	-0.04127	-0.03438	-0.03099	-0.02935	-0.10636	0.42123	-0.09753
7	-C.18757	-0.18645	-0.07122	-0.05912	-0.05314	-0.05097	-0.04760	-0.12080	0.35854	-0.10376
8	-C.23366	-0.18487	-0.09222	-0.07874	-0.07404	-0.06938	-0.06415	-0.12578	0.27011	-0.10619
9	-C.27970	-0.17994	-0.11328	-0.10060	-0.09464	-0.08582	-0.07867	-0.12370	0.16246	-0.10570
10	-C.31971	-0.17616	-0.13401	-0.12355	-0.11364	-0.10146	-0.09154	-0.11507	0.04334	-0.10873

VELOCITY COMPONENTS ON WING PANELS DUE TO BODY LINE SOURCES AND DOUBLETS

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	C.00778	0.01212	0.00792	0.00562	0.00416	0.00317	0.00250	0.00200	0.00172	0.00134
2	C.01504	0.00980	0.00676	0.00478	0.00361	0.00277	0.00221	0.00179	0.00155	0.00124
3	C.01185	0.00825	0.00568	0.00412	0.00312	0.00244	0.00196	0.00161	0.00140	0.00116
4	C.00988	0.00699	0.00483	0.00358	0.00273	0.00217	0.00176	0.00145	0.00127	0.00108
5	C.00830	0.00597	0.00417	0.00312	0.00242	0.00193	0.00158	0.00132	0.00116	0.00101
6	C.00703	0.00503	0.00363	0.00274	0.00216	0.00174	0.00144	0.00121	0.00107	0.00095
7	C.00588	0.00434	0.00318	0.00243	0.00193	0.00157	0.00131	0.00111	0.00098	0.00146
8	C.00503	0.00378	0.00279	0.00217	0.00174	0.00143	0.00120	0.00113	0.00205	0.00304
9	C.00436	0.00332	0.00248	0.00195	0.00158	0.00131	0.00110	0.00225	0.00377	0.00443
10	C.00380	0.00291	0.00222	0.00176	0.00144	0.00120	0.00154	0.00403	0.00509	0.00510

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	C.01302	-0.00416	-0.00400	-0.00342	-0.00278	-0.00222	-0.00182	-0.00148	-0.00129	-0.00102
2	-C.00154	-0.00296	-0.00337	-0.00276	-0.00232	-0.00186	-0.00154	-0.00128	-0.00112	-0.00091
3	-C.00063	-0.00249	-0.00266	-0.00228	-0.00189	-0.00158	-0.00131	-0.00111	-0.00098	-0.00082
4	-C.00066	-0.00208	-0.00212	-0.00190	-0.00159	-0.00135	-0.00113	-0.00096	-0.00086	-0.00074
5	-C.00064	-0.00175	-0.00174	-0.00157	-0.00135	-0.00115	-0.00098	-0.00084	-0.00076	-0.00067
6	-C.00060	-0.00131	-0.00146	-0.00131	-0.00115	-0.00099	-0.00086	-0.00075	-0.00068	-0.00061
7	-C.00036	-0.00108	-0.00121	-0.00111	-0.00098	-0.00086	-0.00075	-0.00066	-0.00060	-0.00143
8	-C.00030	-0.00090	-0.00100	-0.00095	-0.00085	-0.00076	-0.00067	-0.00076	-0.00227	-0.00387
9	-C.00027	-0.00077	-0.00085	-0.00081	-0.00074	-0.00066	-0.00059	-0.00252	-0.00499	-0.00612
10	-C.00025	-0.00062	-0.00073	-0.00070	-0.00065	-0.00058	-0.00133	-0.00535	-0.00720	-0.00725

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10

1	-0.00442	0.00036	0.00062	0.00050	0.00034	0.00023	0.00015	0.00010	0.00007	[illegible]
2	-0.00053	0.00061	0.00064	0.00044	0.00029	0.00019	0.00012	0.00008	0.00006	[illegible]
3	0.00015	0.00075	0.00058	0.00037	0.00024	0.00015	0.00010	0.00006	0.00005	[illegible]
4	0.00062	0.00076	0.00050	0.00031	0.00019	0.00012	0.00008	0.00005	0.00004	[illegible]
5	0.00083	0.00070	0.00042	0.00025	0.00015	0.00009	0.00006	0.00004	0.00003	[illegible]
6	0.00087	0.00059	0.00034	0.00019	0.00012	0.00008	0.00005	0.00004	0.00003	[illegible]
7	0.00077	0.00048	0.00026	0.00015	0.00009	0.00006	0.00004	0.00003	0.00002	5
8	0.00065	0.00038	0.00020	0.00012	0.00007	0.00005	0.00004	0.00003	0.00002	12
9	0.00053	0.00030	0.00016	0.00009	0.00006	0.00004	0.00003	0.00009	0.00009	018
10	0.00041	0.00022	0.00012	0.00007	0.00005	0.00004	0.00006	0.00018	0.00021	021

VELOCITY COMPONENTS ON WING PANELS DUE TO WING SOURCES

AXIAL(U)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.03054	-0.02433	-0.01880	-0.01746	-0.01326	-0.00857	-0.00441	-0.00076	[illegible]	-0.00331
2	-0.00716	-0.00013	0.00337	0.00784	0.01289	0.01731	0.02137	0.02116	[illegible]	0.01976
3	-0.00189	0.00375	0.00866	0.01475	0.01964	0.02390	0.02566	0.02337	[illegible]	0.02314
4	-0.00060	0.00495	0.01303	0.01903	0.02368	0.02789	0.02409	0.02362	[illegible]	0.02249
5	-0.00045	0.00650	0.01654	0.02209	0.02667	0.02357	0.02284	0.02278	[illegible]	0.02115
6	0.00270	0.01161	0.01874	0.02411	0.02187	0.02069	0.02053	0.01941	[illegible]	0.01855
7	0.00547	0.01297	0.01968	0.01900	0.01691	0.01658	0.01565	0.01477	[illegible]	0.01431
8	0.00716	0.01399	0.01652	0.01237	0.01172	0.01114	0.00986	0.00933	[illegible]	0.00913
9	0.00900	0.01548	0.00706	0.00593	0.00571	0.00393	0.00330	0.00293	[illegible]	0.00307
10	0.01097	-0.00233	-0.00442	-0.00488	-0.00666	-0.00745	-0.00788	-0.00816	-0.00831	-0.00557

TRANSVERSE(V)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.02346	0.03708	0.03491	0.03503	0.02960	0.02265	0.01605	[illegible]	0.01062	0.01346
2	-0.02849	-0.02351	-0.02451	-0.02979	-0.03719	-0.04422	-0.05097	[illegible]	-0.05010	-0.04997
3	-0.03063	-0.02966	-0.03436	-0.04316	-0.05102	-0.05823	-0.06183	[illegible]	-0.05886	-0.05917
4	-0.02439	-0.02619	-0.03974	-0.04951	-0.05761	-0.06514	-0.06024	[illegible]	-0.06022	-0.05877
5	-0.01502	-0.02915	-0.04278	-0.05283	-0.06136	-0.05790	-0.05738	[illegible]	-0.05636	-0.05561
6	-0.00956	-0.02868	-0.04278	-0.05337	-0.05185	-0.05088	-0.05112	[illegible]	-0.04905	-0.04882
7	-0.00345	-0.02458	-0.03956	-0.04162	-0.03967	-0.03983	-0.03886	[illegible]	-0.03731	-0.03742
8	0.00316	-0.02012	-0.03011	-0.02594	-0.02597	-0.02555	-0.02394	[illegible]	-0.02304	-0.02340
9	0.00822	-0.01726	-0.00930	-0.00904	-0.00947	-0.00721	-0.00649	[illegible]	-0.00586	-0.00692
10	0.01231	0.02030	0.02082	0.02039	0.02246	0.02335	0.02383	[illegible]	0.02430	0.01669

VERTICAL(W)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829	0.10829
2	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022	0.05022
3	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498	0.02498
4	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695	0.00695
5	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762	-0.00762
6	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197	-0.02197
7	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357	-0.03357
8	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034	-0.04034

9	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620	-0.04620
10	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180	-0.05180

VELOCITY COMPONENTS ON BODY PANELS DUE TO BODY PANEL PRESSURE SINGULARITIES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.00000	0.00001	0.00005	0.00060	-0.00060	-0.00005	-0.00002
2	-0.00003	0.00002	0.00067	-0.00516	-0.00834	0.00010	-0.00013
3	-0.00009	0.00194	-0.00168	0.00158	-0.01119	-0.00318	0.00173
4	0.00677	-0.00066	0.00080	0.00867	-0.01167	-0.00311	0.00255
5	0.00287	0.00592	0.00916	0.01837	-0.00647	-0.00485	-0.00283
6	0.00059	0.00536	0.01241	0.02106	-0.00959	-0.00672	-0.00705
7	0.00119	0.00554	0.01270	0.02147	-0.01091	-0.00998	-0.01027
8	0.00346	0.00656	0.01216	0.02052	-0.01298	-0.01210	-0.01235
9	0.00544	0.00745	0.01188	0.01844	-0.01443	-0.01363	-0.01381
10	0.00644	0.00816	0.01137	0.01549	-0.01403	-0.01470	-0.01519
11	0.00920	0.00922	0.00719	0.00370	-0.00986	-0.01266	-0.01509
12	0.00892	0.00722	0.00280	-0.00388	-0.00706	-0.00784	-0.00928
13	0.00537	0.00475	0.00336	0.00066	-0.00220	-0.00380	-0.00504
14	0.00334	0.00269	0.00182	0.00026	-0.00108	-0.00223	-0.00277

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.00000	-0.00002	-0.00005	-0.00151	0.00126	0.00008	0.00002
2	-0.00005	0.00003	-0.00139	0.00907	0.01800	0.00012	0.00012
3	0.00030	-0.00111	0.00028	-0.00250	0.03283	0.00250	0.00146
4	0.00017	-0.00088	-0.00048	-0.02868	0.05151	-0.00035	-0.00300
5	-0.00278	-0.00735	-0.01163	-0.06984	0.06324	0.00069	-0.00106
6	-0.00157	-0.00752	-0.01891	-0.10748	0.08696	0.00646	-0.00164
7	-0.00080	-0.00524	-0.02504	-0.14710	0.11271	0.01256	0.00034
8	0.00051	-0.00482	-0.03248	-0.18717	0.13928	0.02060	0.00245
9	-0.00002	-0.00645	-0.04035	-0.22525	0.16492	0.02733	0.00398
10	0.00000	-0.00738	-0.04766	-0.25823	0.18770	0.03439	0.00646
11	-0.00070	-0.01085	-0.06142	-0.28005	0.21021	0.04726	0.00959
12	-0.00219	-0.01738	-0.06726	-0.27066	0.22340	0.05361	0.01267
13	-0.00670	-0.02699	-0.07652	-0.27209	0.22721	0.05439	0.01277
14	-0.00876	-0.03247	-0.08157	-0.26480	0.21112	0.05855	0.01501

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.00000	0.00001	0.00010	-0.00018	-0.00078	0.00006	0.00000
2	0.00001	-0.00003	-0.00260	-0.00198	-0.00739	0.00310	0.00003
3	0.00038	-0.00494	-0.00228	-0.00276	-0.01745	0.00344	0.00557
4	-0.01139	-0.00344	-0.00048	-0.00602	-0.02828	0.00153	0.01224
5	-0.01022	-0.00696	-0.00548	-0.00860	-0.04129	-0.00329	[illegible]

6	-0.00606	-0.00511	-0.00724	-0.01534	-0.05675	-0.00968	0.00593
7	-0.00271	-0.00568	-0.01234	-0.01754	-0.07245	-0.01822	-0.00056
8	-0.00073	-0.00546	-0.01577	-0.02199	-0.08854	-0.02672	-0.00872
9	0.00083	-0.00653	-0.01907	-0.02632	-0.10531	-0.03647	-0.01814
10	0.00128	-0.00750	-0.02230	-0.03005	-0.12045	-0.04575	-0.02765
11	-0.00206	-0.01353	-0.02903	-0.03310	-0.13421	-0.06072	-0.04511
12	-0.01096	-0.02139	-0.03332	-0.03302	-0.13892	-0.06728	-0.05347
13	-0.02098	-0.02754	-0.03520	-0.03217	-0.14141	-0.07037	-0.05589
14	-0.03025	-0.03454	-0.04026	-0.06825	-0.17183	-0.07823	-0.06430

VELOCITY COMPONENTS ON BODY PANELS DUE TO WING PANEL PRESSURE SINGULARITIES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.	-0.	-0.	0.00203	-0.00140	-0.	-0.
2	-0.	-0.	0.00132	0.00482	-0.00332	-0.00089	-0.
3	0.00043	0.00105	0.00311	0.00842	-0.00567	-0.00208	-0.00090
4	0.00172	0.00262	0.00695	0.01145	-0.00797	-0.00446	-0.00216
5	0.00377	0.00614	0.00996	0.01360	-0.00963	-0.00668	-0.00441
6	0.00729	0.00875	0.01206	0.01518	-0.01084	-0.00857	-0.00701
7	0.01000	0.01133	0.01389	0.01627	-0.01180	-0.01003	-0.00879
8	0.01153	0.01302	0.01455	0.01638	-0.01190	-0.01074	-0.00997
9	0.01282	0.01342	0.01443	0.01536	-0.01120	-0.01078	-0.01042
10	0.01271	0.01326	0.01351	0.01253	-0.00924	-0.01017	-0.01008
11	0.01036	0.01002	0.00608	0.00212	-0.00189	-0.00500	-0.00749
12	0.00357	0.00341	0.00316	0.00133	-0.00110	-0.00218	-0.00194
13	0.00341	0.00324	0.00255	0.00096	-0.00088	-0.00198	-0.00238
14	0.00272	0.00250	0.00174	0.00065	-0.00055	-0.00138	-0.00172

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.	-0.	-0.	0.00166	-0.00254	-0.	-0.
2	-0.	-0.	0.00086	0.00981	-0.01123	-0.00126	-0.
3	0.00040	0.00103	0.00339	0.02655	-0.02659	-0.00412	-0.00075
4	0.00085	0.00271	0.01033	0.05136	-0.04856	-0.00960	-0.00180
5	0.00196	0.00739	0.01975	0.08076	-0.07434	-0.01718	-0.00449
6	0.00356	0.01266	0.03033	0.11323	-0.10286	-0.02521	-0.00642
7	0.00488	0.01692	0.04087	0.14750	-0.13313	-0.03347	-0.00855
8	0.00625	0.02154	0.05083	0.18241	-0.16382	-0.04166	-0.01082
9	0.00739	0.02534	0.06029	0.21605	-0.19321	-0.04932	-0.01270
10	0.00835	0.02879	0.06877	0.24529	-0.21804	-0.05614	-0.01462
11	0.00987	0.03454	0.07899	0.25684	-0.23313	-0.06259	-0.01723
12	0.01023	0.03525	0.08201	0.26873	-0.23450	-0.06401	-0.01677
13	0.01152	0.03893	0.08602	0.27056	-0.23599	-0.06625	-0.01794
14	0.01179	0.03946	0.08670	0.26386	-0.21572	-0.06508	-0.01781

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000

ROW NO.							
1	C.	0.	0.	-0.00405	-0.00186	0.	0.
2	C.	0.	-0.00221	-0.01203	-0.00380	-0.00102	0.
3	-C.00053	-0.00154	-0.00565	-0.02135	-0.00417	-0.00210	-0.00066
4	-C.00191	-0.0037C	-0.01154	-0.02731	0.00023	-0.00333	-0.00105
5	-C.00383	-0.00765	-0.01573	-0.03049	0.00877	-0.00250	-0.00117
6	-C.00638	-0.00990	-0.01819	-0.03238	0.01996	0.00033	0.00038
7	-C.00713	-0.01089	-0.01935	-0.03356	0.03318	0.00509	0.00410
8	-C.00631	-0.01032	-0.01898	-0.03415	0.04768	0.01136	0.00929
9	-C.00424	-0.00831	-0.01764	-0.03350	0.06276	0.01843	0.01557
10	-C.00111	-0.00552	-0.01510	-0.02950	0.07844	0.02603	0.02237
11	C.00705	0.00324	-0.00078	-0.00435	0.10679	0.04372	0.03628
12	C.01884	0.01599	0.00951	0.00673	0.11724	0.05350	0.04762
13	C.02316	0.021C3	0.01689	0.01631	0.12613	0.06015	0.05265
14	C.03119	0.0303C	0.02986	0.06124	0.16411	0.07282	0.06266

VELOCITY COMPONENTS ON BODY PANELS DUE TO WING SOURCES

AXIAL(U)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	C.	0.	0.	-0.00275	-0.00157	0.	0.
2	C.	0.	-0.00645	-0.01262	-0.01134	-0.00520	0.
3	C.	-0.00686	-0.00902	-0.00734	-0.00732	-0.00785	-0.00585
4	-C.01113	-0.00699	-0.00646	-0.00369	-0.00407	-0.00956	-0.01252
5	-C.01124	-0.01126	-0.00895	-0.00581	-0.00707	-0.00877	-0.01019
6	-C.00891	-0.00826	-0.00613	-0.00292	-0.00375	-0.00602	-0.00737
7	-C.00613	-0.00531	-0.00300	0.00021	-0.00054	-0.00297	-0.00441
8	-C.00328	-0.00224	0.00013	0.00300	0.00234	0.00008	-0.00138
9	-C.00025	0.00076	0.00290	0.00529	0.00473	0.00290	0.00167
10	C.00267	0.00351	0.00541	0.00753	0.00705	0.00542	0.00439
11	C.00632	0.00717	0.00881	0.00308	0.00388	0.00862	0.00773
12	C.00740	0.00597	0.00467	0.00365	0.00410	0.00548	0.00453
13	C.00208	0.00191	0.00159	0.00128	0.00131	0.00154	0.00169
14	C.00092	0.00C84	0.00070	0.00059	0.00061	0.00071	0.00078

TRANSVERSE(V)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	C.	0.	0.	-0.00015	-0.00106	0.	0.
2	C.	0.	-0.00215	-0.01927	-0.02232	-0.00479	0.
3	C.	-0.0046C	-0.00837	-0.02398	-0.02413	-0.01141	-0.00699
4	-C.00260	-0.0075C	-0.01101	-0.02218	-0.02128	-0.00715	-0.00056
5	-C.00052	0.00001	-0.00376	-0.01012	-0.00750	-0.00338	-0.00047
6	-C.00028	-0.00024	-0.00255	-0.00459	-0.00255	-0.00168	-0.00015
7	-C.00016	-0.00001	-0.00119	0.00100	0.00295	-0.00000	0.00020
8	-C.00004	0.00046	0.00043	0.00642	0.00817	0.00181	0.00071
9	C.00015	0.00109	0.00215	0.01104	0.01251	0.00342	0.00112
10	C.00032	0.00154	0.00361	0.01487	0.01607	0.00468	0.00134
11	C.00066	0.00205	0.00560	0.01433	0.01246	0.00665	0.00165
12	-C.00173	-0.00105	0.00067	0.00264	0.00152	-0.00122	-0.00010

13	0.00013	0.00054	0.00127	0.00194	0.00169	0.00090	0.00026
14	0.00015	0.00048	0.00086	0.00112	0.00099	0.00059	0.00019

VERTICAL(W)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.	-0.	0.	0.00415	-0.00212	-0.	-0.
2	-0.	-0.	0.00997	0.03008	-0.02056	-0.00664	-0.
3	-0.	0.01007	0.01697	0.02118	-0.01490	-0.01133	-0.00630
4	0.01365	0.01149	0.01425	0.01298	-0.00936	-0.01133	-0.01238
5	0.01435	0.01457	0.01285	0.00678	-0.00548	-0.00945	-0.01063
6	0.01206	0.01125	0.00839	0.00032	-0.00087	-0.00633	-0.00813
7	0.00397	0.00761	0.00357	-0.00594	0.00360	-0.00291	-0.00532
8	0.00355	0.00359	-0.00133	-0.01125	0.00744	0.00059	-0.00228
9	0.00174	-0.00049	-0.00573	-0.01520	0.01034	0.00380	0.00088
10	-0.00209	-0.00429	-0.00956	-0.01861	0.01287	0.00662	0.00377
11	-0.00716	-0.00946	-0.01471	-0.00827	0.00607	0.01034	0.00750
12	-0.00928	-0.00750	-0.00580	-0.00246	0.00212	0.00487	0.00492
13	-0.00328	-0.00306	-0.00239	-0.00094	0.00081	0.00182	0.00215
14	-0.00165	-0.00150	-0.00111	-0.00043	0.00035	0.00086	0.00106

PRESSURES, FORCES, AND MOMENTS ON ISOLATED BODY

CD = 0.00086 CL = 0.00004 CM = -0.00385

BODY PRESSURE COEFFICIENTS(CP)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
X							
0.	0.14924	0.14475	0.13660	0.12587	0.11447	0.10579	0.10129
0.7243	0.14924	0.14475	0.13660	0.12587	0.11447	0.10579	0.10129
1.4485	0.13613	0.13409	0.13040	0.12553	0.12035	0.11642	0.11438
2.1728	0.06963	0.06809	0.06531	0.06164	0.05775	0.05478	0.05325
2.8970	0.06581	0.06469	0.06322	0.06102	0.05869	0.05691	0.05599
3.6213	0.04947	0.04865	0.04715	0.04518	0.04309	0.04149	0.04067
4.3456	0.04872	0.04801	0.04672	0.04501	0.04321	0.04183	0.04112
5.0698	0.03998	0.03917	0.03770	0.03576	0.03370	0.03214	0.03133
5.7941	0.03940	0.03854	0.03699	0.03495	0.03278	0.03113	0.03027
6.5184	0.03188	0.03117	0.02987	0.02816	0.02635	0.02496	0.02425
7.2426	0.02908	0.02862	0.02780	0.02671	0.02555	0.02467	0.02421
7.9669	0.02470	0.02405	0.02288	0.02134	0.01971	0.01846	0.01782
8.6911	0.01971	0.01962	0.01946	0.01924	0.01902	0.01884	0.01875
9.4154	0.01597	0.01578	0.01543	0.01497	0.01449	0.01412	0.01393
10.1397	0.01226	0.01228	0.01230	0.01234	0.01238	0.01241	0.01242
10.8639	0.00741	0.00761	0.00796	0.00843	0.00893	0.00930	0.00950
11.5882	0.00440	0.00447	0.00459	0.00475	0.00491	0.00504	0.00511
12.3124	-0.00026	0.00000	0.00047	0.00108	0.00173	0.00223	0.00249
13.0367	-0.00653	-0.00610	-0.00533	-0.00430	-0.00322	-0.00239	-0.00196
13.7610	-0.00985	-0.00927	-0.00824	-0.00687	-0.00542	-0.00432	-0.00375
14.5023	-0.01858	-0.01760	-0.01580	-0.01344	-0.01093	-0.00902	-0.00803
15.2436	-0.02490	-0.02389	-0.02204	-0.01960	-0.01701	-0.01504	-0.01402

15.9849	-0.03904	-0.03829	-0.03692	-0.03512	-0.03321	-0.03175	-0.03100
16.7262	-0.03187	-0.03135	-0.03042	-0.02919	-0.02788	-0.02688	-0.02636
17.4675	-0.02611	-0.02578	-0.02520	-0.02442	-0.02360	-0.02297	-0.02265
18.2088	-0.02155	-0.02138	-0.02105	-0.02062	-0.02017	-0.01982	-0.01964
18.9501	-0.01796	-0.01789	-0.01775	-0.01758	-0.01739	-0.01724	-0.01717
19.6914	-0.01513	-0.01513	-0.01512	-0.01511	-0.01510	-0.01509	-0.01509
20.4327	-0.01289	-0.01293	-0.01300	-0.01310	-0.01320	-0.01328	-0.01332
21.1740	-0.01111	-0.01117	-0.01129	-0.01144	-0.01160	-0.01173	-0.01179
21.9153	-0.00968	-0.00976	-0.00989	-0.01007	-0.01025	-0.01039	-0.01047
22.6566	-0.00853	-0.00860	-0.00874	-0.00891	-0.00910	-0.00924	-0.00932
23.3979	-0.00759	-0.00766	-0.00778	-0.00794	-0.00811	-0.00825	-0.00831
24.1392	-0.00681	-0.00687	-0.00697	-0.00712	-0.00727	-0.00738	-0.00744
24.8805	-0.00615	-0.00620	-0.00629	-0.00641	-0.00653	-0.00663	-0.00668
25.6218	-0.00559	-0.00563	-0.00570	-0.00580	-0.00590	-0.00598	-0.00602
26.3631	-0.00511	-0.00514	-0.00520	-0.00527	-0.00535	-0.00541	-0.00544
27.1044	-0.00469	-0.00472	-0.00476	-0.00481	-0.00487	-0.00491	-0.00494
27.8457	-0.00432	-0.00434	-0.00437	-0.00441	-0.00445	-0.00448	-0.00450
28.5870	-0.00399	-0.00401	-0.00403	-0.00405	-0.00408	-0.00411	-0.00412
29.3283	-0.00370	-0.00371	-0.00372	-0.00374	-0.00376	-0.00378	-0.00378
30.0696	-0.00589	-0.00590	-0.00590	-0.00592	-0.00593	-0.00594	-0.00594
30.8109	-0.01355	-0.01356	-0.01356	-0.01357	-0.01358	-0.01358	-0.01358
31.5522	-0.02129	-0.02129	-0.02130	-0.02130	-0.02131	-0.02131	-0.02131
32.2935	-0.02605	-0.02605	-0.02605	-0.02606	-0.02606	-0.02606	-0.02606
33.0348	-0.02963	-0.02963	-0.02963	-0.02963	-0.02963	-0.02963	-0.02963
33.7761	-0.03597	-0.03597	-0.03597	-0.03597	-0.03598	-0.03598	-0.03598
34.5174	-0.03776	-0.03776	-0.03776	-0.03776	-0.03776	-0.03776	-0.03776
35.2587	-0.02761	-0.02761	-0.02762	-0.02762	-0.02762	-0.02762	-0.02762
36.0000	-0.01048	-0.01048	-0.01048	-0.01048	-0.01048	-0.01048	-0.01048

INCREMENTAL PRESSURES, FORCES, AND MOMENTS ON BODY PANELS DUE TO WING

CD = 0.00001 CL = 0.01030 CM = -0.23571

BODY PANEL PRESSURE COEFFICIENT(CP)

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	-0.00001	-0.00003	-0.00011	0.00025	0.00714	0.00010	0.00003
2	0.00005	-0.00005	0.00892	0.02592	0.04600	0.01197	0.00026
3	-0.00067	0.00773	0.01517	-0.00530	0.04834	0.02622	0.01002
4	0.00527	0.01006	-0.00257	-0.03287	0.04743	0.03427	0.02426
5	0.00918	-0.00160	-0.02034	-0.05231	0.04633	0.04061	0.03487
6	0.00205	-0.01168	-0.03669	-0.06664	0.04836	0.04260	0.04287
7	-0.01011	-0.02313	-0.04718	-0.07591	0.04651	0.04597	0.04695
8	-0.02404	-0.03468	-0.05368	-0.07980	0.04509	0.04551	0.04741
9	-0.03602	-0.04326	-0.05842	-0.07819	0.04180	0.04299	0.04511
10	-0.04364	-0.04984	-0.06057	-0.07110	0.03244	0.03888	0.04177
11	-0.03175	-0.05281	-0.04414	-0.01779	0.01576	0.01310	0.02971
12	-0.03980	-0.03319	-0.02125	-0.00218	0.00811	0.00907	0.01337
13	-0.02172	-0.01988	-0.01500	-0.00580	0.00355	0.00848	0.01147
14	-0.01397	-0.01246	-0.00852	-0.00300	0.00204	0.00580	0.00742

BODY PANEL SLOPE

THETA(DEG.)	12.5000	37.5000	62.5000	88.5950	116.0950	142.5000	167.5000
ROW NO.							
1	0.02349	0.02349	0.02349	0.02339	0.02336	0.02349	0.02349
2	0.00619	0.00619	0.00619	0.00616	0.00615	0.00619	0.00619
3	0.	0.	0.	0.	-0.	-0.	-0.
4	0.	0.	0.	0.	-0.	-0.	-0.
5	0.	0.	0.	0.	-0.	-0.	-0.
6	0.	0.	0.	0.	-0.	-0.	-0.
7	0.	0.	0.	0.	-0.	-0.	-0.
8	0.	0.	0.	0.	-0.	-0.	-0.
9	0.	0.	0.	0.	-0.	-0.	-0.
10	0.	0.	0.	0.	-0.	-0.	-0.
11	0.	0.	0.	0.	-0.	-0.	-0.
12	0.	0.	0.	0.	-0.	-0.	-0.
13	0.	0.	0.	0.	-0.	-0.	-0.
14	-0.01422	-0.01422	-0.01422	-0.01415	-0.01413	-0.01422	-0.01422

PRESSURES, FORCES, AND MOMENTS ON WING PANELS IN PRESENCE OF BODY

CD = 0.00268 CL = 0.10000 CM = -3.02498

SPANWISE CD DISTRIBUTION

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
	0.00048	0.00136	0.00041	0.00035	0.00027	0.00019	0.00011	0.00053	-0.00136	0.0003

SPANWISE CL DISTRIBUTION

SPANWISE STATION	1	2	3	4	5	6	7	8	9	10
	0.00571	0.01095	0.01068	0.01078	0.01083	0.01077	0.01059	0.01065	0.00423	0.0148

UPPER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.04153	-0.03371	-0.01516	-0.01455	-0.01654	-0.02278	-0.02608	-0.03395	0.05997	-0.05033
2	-0.04175	-0.10066	-0.07910	-0.07426	-0.08379	-0.08699	-0.09244	-0.09450	0.00581	-0.12313
3	-0.06413	-0.11412	-0.09653	-0.09437	-0.10214	-0.10912	-0.11037	-0.11823	-0.03570	-0.13597
4	-0.07734	-0.11560	-0.10649	-0.10603	-0.11465	-0.12324	-0.11640	-0.12880	-0.07580	-0.12659
5	-0.08978	-0.11149	-0.11202	-0.11654	-0.11937	-0.11926	-0.11715	-0.13252	-0.10507	-0.12455
6	-0.09677	-0.10563	-0.11434	-0.12328	-0.11773	-0.11130	-0.11689	-0.11522	-0.12297	-0.09390
7	-0.10794	-0.09051	-0.11233	-0.10900	-0.10796	-0.10518	-0.10428	-0.09659	-0.12779	-0.07269
8	-0.10180	-0.07674	-0.09896	-0.09365	-0.09026	-0.09024	-0.08137	-0.07163	-0.12633	-0.05075
9	-0.09321	-0.06330	-0.07026	-0.06858	-0.06985	-0.06152	-0.05602	-0.04590	-0.11549	-0.02541
10	-0.07314	-0.01357	-0.02360	-0.03478	-0.02412	-0.02283	-0.01665	-0.01040	-0.07696	0.01595

LOWER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.07459	0.08850	0.05806	0.04421	0.04177	0.03564	0.03271	0.03093	-0.05715	0.06956
2	0.02125	0.06015	0.02337	0.01633	0.00673	0.00454	-0.00022	0.00846	-0.08832	0.04564
3	0.02226	0.05373	0.02060	0.01738	0.01032	0.00514	0.00428	0.02168	-0.05275	0.04138
4	0.02475	0.04447	0.01877	0.02034	0.01360	0.00676	0.01738	0.03595	-0.01675	0.03540
5	0.02570	0.03324	0.01842	0.01987	0.01880	0.02181	0.03260	0.03400	0.01655	0.01153
6	0.03093	0.01900	0.01947	0.01828	0.02634	0.03709	0.03728	0.03633	0.04190	0.01395
7	0.02888	0.01310	0.02111	0.03157	0.03715	0.04120	0.03956	0.03110	0.06428	0.00751
8	0.03563	0.00488	0.02665	0.04048	0.04657	0.04164	0.04015	0.02778	0.07990	0.00130
9	0.03264	-0.00544	0.03851	0.04836	0.04451	0.04444	0.03604	0.02314	0.08880	-0.00517
10	0.01850	0.01778	0.05260	0.04486	0.04992	0.04472	0.03967	0.02626	0.08948	-0.01459

WING PANEL PRESSURE DIFFERENCE

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.03306	0.12222	0.07322	0.05876	0.05831	0.05841	0.05879	0.06488	-0.11711	0.11989
2	0.06300	0.16081	0.10247	0.09060	0.09052	0.09152	0.09222	0.10296	-0.09413	0.16877
3	0.08639	0.16785	0.11713	0.11175	0.11246	0.11426	0.11465	0.13991	-0.01705	0.17734
4	0.10210	0.16007	0.12527	0.12637	0.12825	0.13000	0.13378	0.16475	0.05905	0.16199
5	0.11548	0.14473	0.13043	0.13641	0.13867	0.14107	0.14974	0.16651	0.12162	0.13608
6	0.12770	0.12483	0.13381	0.14156	0.14408	0.14839	0.15417	0.15156	0.16487	0.10785
7	0.13083	0.10361	0.13344	0.14057	0.14511	0.14637	0.14384	0.12769	0.19206	0.08019
8	0.13743	0.08162	0.12562	0.13413	0.13683	0.13188	0.12152	0.09940	0.20623	0.05205
9	0.12585	0.05786	0.10878	0.11694	0.11436	0.10597	0.09207	0.06904	0.20429	0.02024
10	0.09164	0.03135	0.07621	0.07964	0.07405	0.06755	0.05632	0.03666	0.16644	-0.03054

UPPER SURFACE WING PANEL SLOPE

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	0.09634	0.08957	0.14439	0.15055	0.15183	0.15558	0.15963	0.16273	0.28587	0.15512
2	0.01990	-0.01962	0.07037	0.07861	0.08034	0.08466	0.08890	0.09168	0.35537	0.04963
3	-0.01457	-0.08288	0.03118	0.03851	0.04132	0.04581	0.05161	0.03580	0.42206	-0.01465
4	-0.03534	-0.12841	-0.00073	0.00516	0.00905	0.01372	0.01982	-0.02865	0.45555	-0.06035
5	-0.05233	-0.15981	-0.03037	-0.02488	-0.02054	-0.01578	-0.01337	-0.08388	0.44773	-0.09188
6	-0.07268	-0.18140	-0.06127	-0.05545	-0.05055	-0.04833	-0.04719	-0.12493	0.40231	-0.11686
7	-0.09646	-0.19224	-0.09155	-0.08436	-0.08017	-0.07941	-0.07692	-0.15082	0.32802	-0.13463
8	-0.12150	-0.19272	-0.11823	-0.10964	-0.10753	-0.10418	-0.10025	-0.16259	0.23282	-0.14386
9	-0.14675	-0.19077	-0.14357	-0.13693	-0.13342	-0.12651	-0.12046	-0.16635	0.11920	-0.14920
10	-0.17455	-0.18912	-0.16912	-0.16431	-0.15707	-0.14759	-0.13891	-0.16345	-0.00553	-0.15778

LOWER SURFACE WING PANEL SLOPE

SPANWISE STATION / CHORDWISE STATION	1	2	3	4	5	6	7	8	9	10
1	-0.12024	-0.12701	-0.07219	-0.06603	-0.06475	-0.06100	-0.05695	-0.05385	0.06929	-0.06146
2	-0.08054	-0.12006	-0.03007	-0.02183	-0.02009	-0.01578	-0.01153	-0.00875	0.25494	-0.05081
3	-0.06454	-0.13285	-0.01878	-0.01146	-0.00865	-0.00416	0.00164	-0.01417	0.37210	-0.06462
4	-0.04924	-0.14230	-0.01462	-0.00874	-0.00484	-0.00018	0.00593	-0.04254	0.44166	-0.07425
5	-0.03709	-0.14457	-0.01513	-0.00964	-0.00530	-0.00054	0.00187	-0.06864	0.46297	-0.07664
6	-0.02874	-0.13745	-0.01733	-0.01150	-0.00661	-0.00438	-0.00324	-0.08099	0.44626	-0.07291
7	-0.02933	-0.12511	-0.02442	-0.01723	-0.01304	-0.01228	-0.00979	-0.08369	0.39515	-0.06750

8	-0.04032	-0.11203	-0.03754	-0.02895	-0.02634	-0.02349	-0.01956	-0.03190	0.31351	-0.06317
9	-0.05435	-0.09538	-0.05113	-0.04454	-0.04193	-0.03412	-0.02807	-0.07396	0.21159	-0.05681
10	-0.07095	-0.08553	-0.05552	-0.05071	-0.05437	-0.04399	-0.03532	-0.05985	0.09806	-0.05418

FORCES AND MOMENTS ON WING-BODY COMBINATION

CD = 0.00356 CL = 0.11034 CM = -3.26454

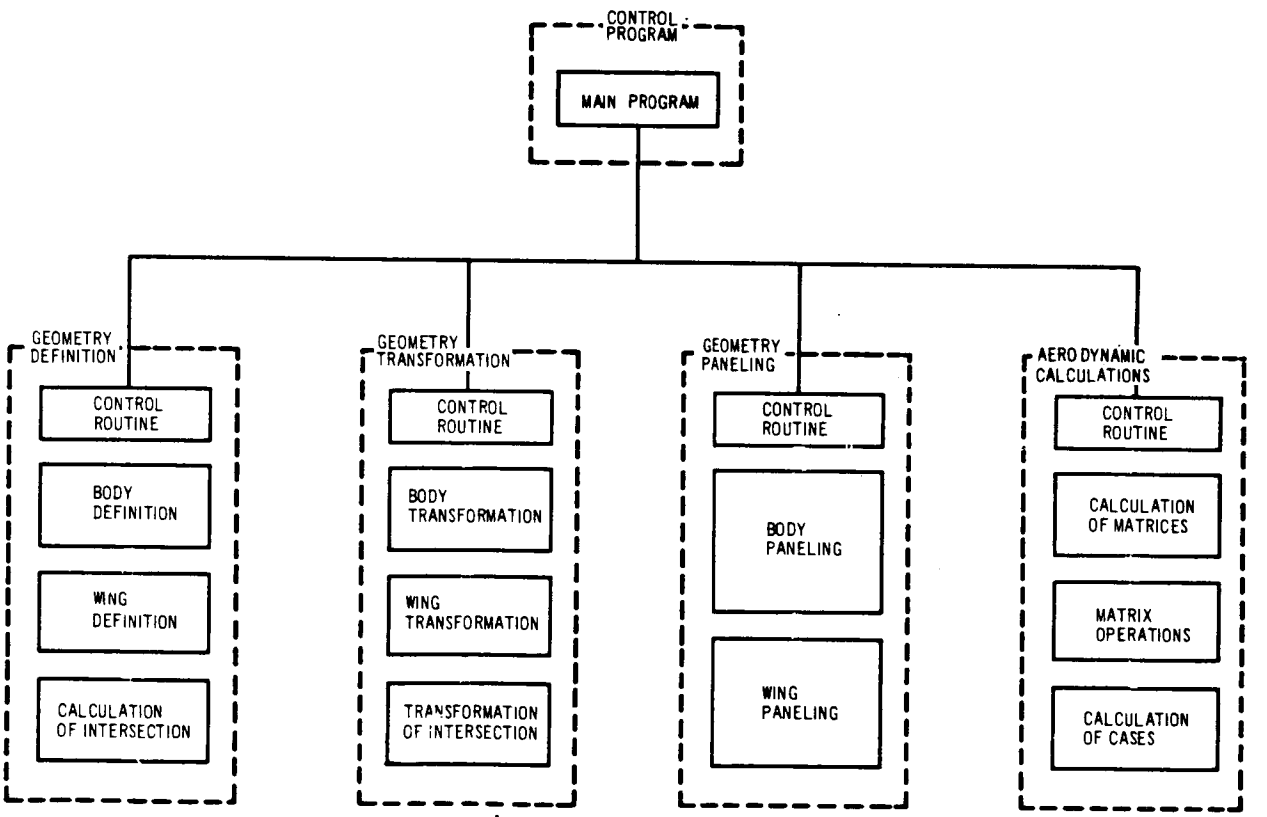
END OF DATA TIME 18.354 091365

4. PROGRAM RECORDS

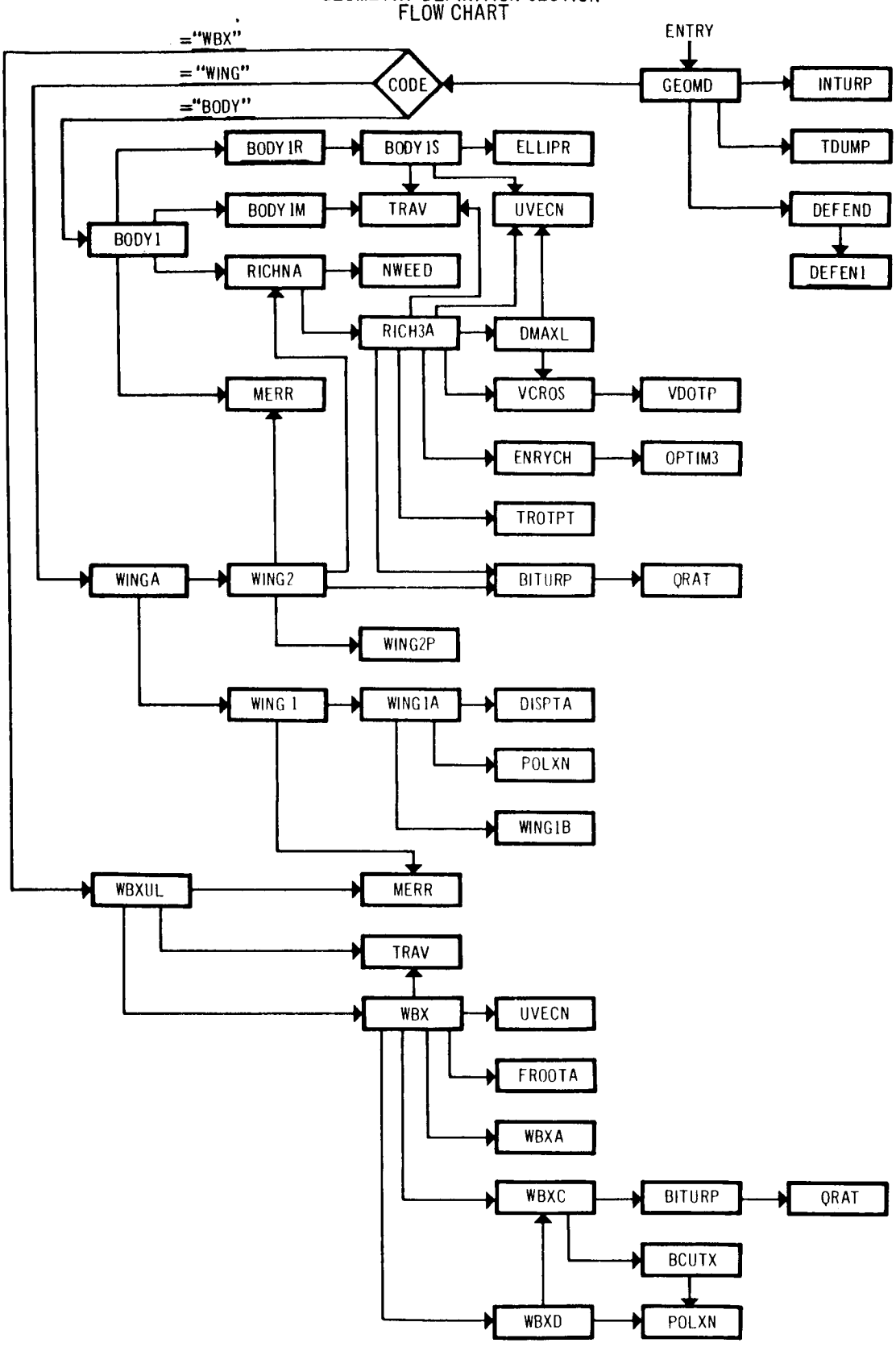
4.1 Flow Charts

A program flow chart, supplemented by flow charts of each of the four main program sections, and a program overlay structure diagram are presented in this section.

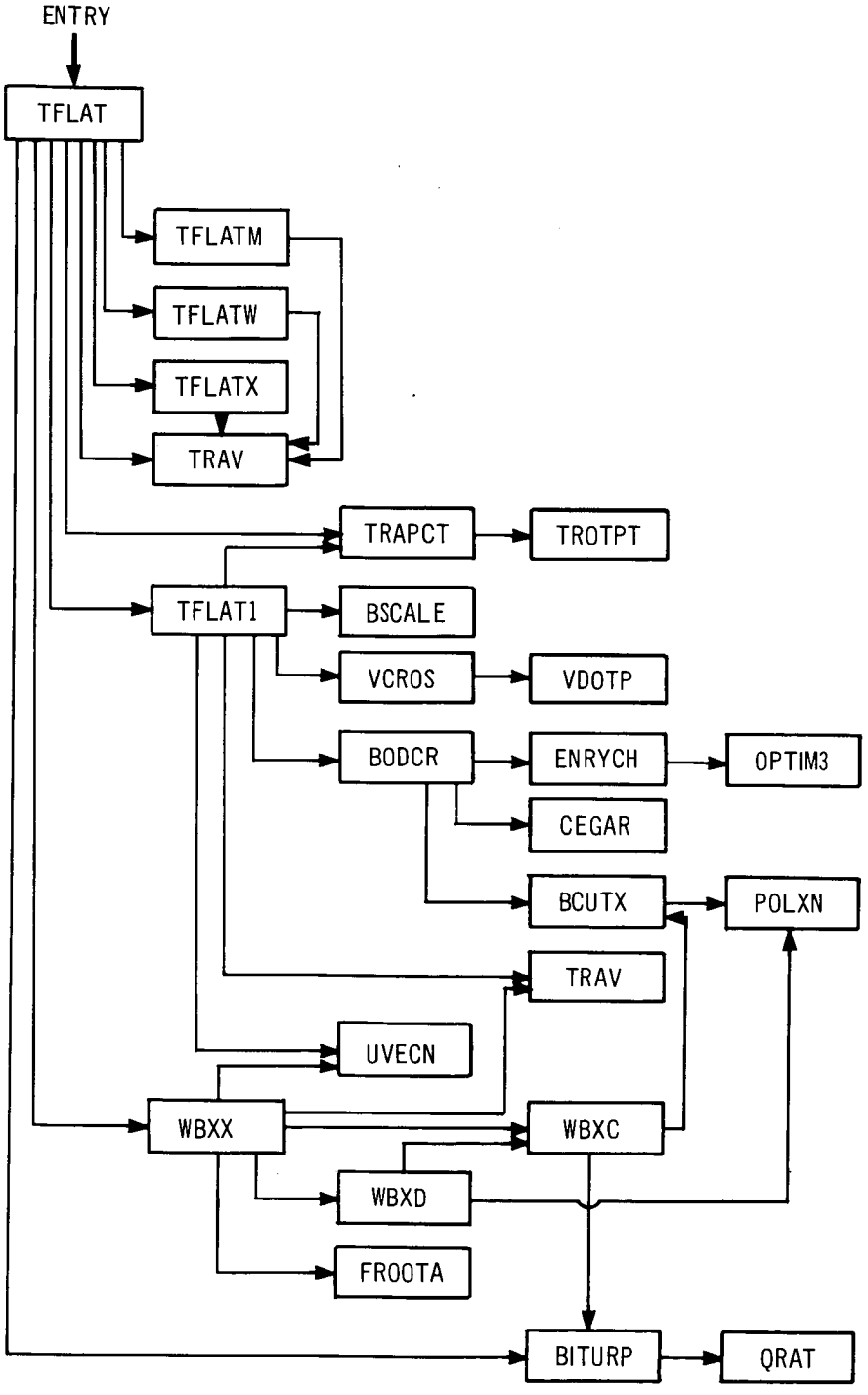
PROGRAM FLOW CHART



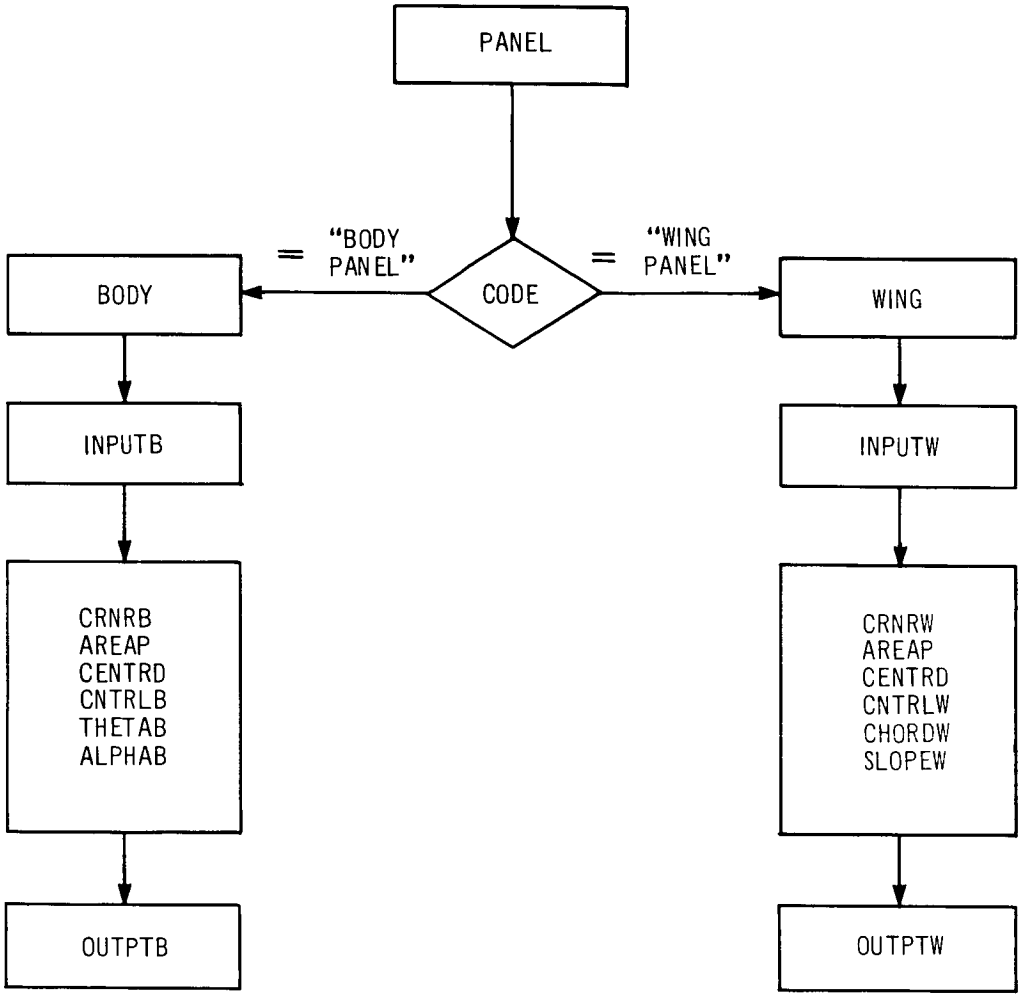
GEOMETRY DEFINITION SECTION
FLOW CHART



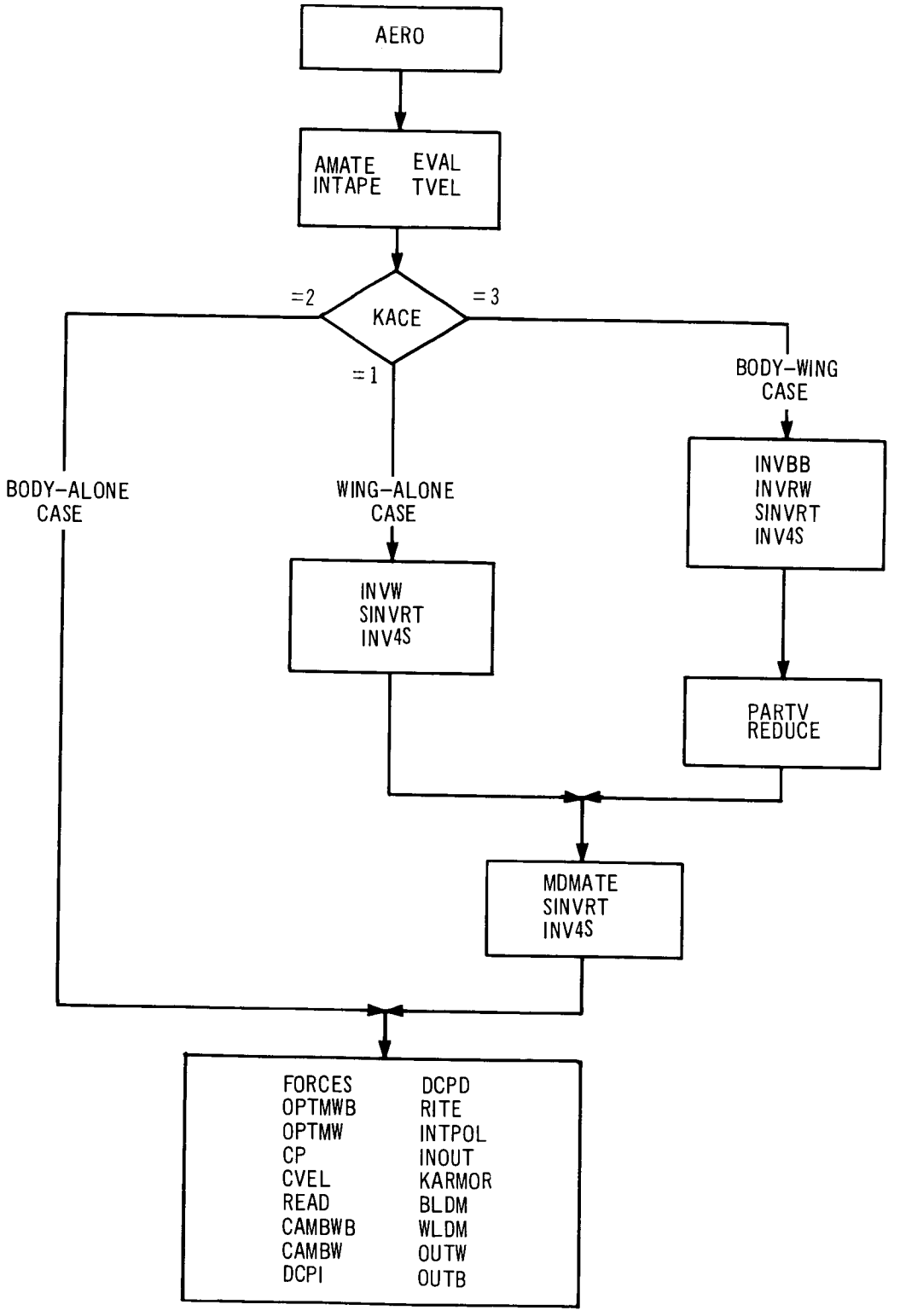
GEOMETRY TRANSFORMATION SECTION
FLOW CHART



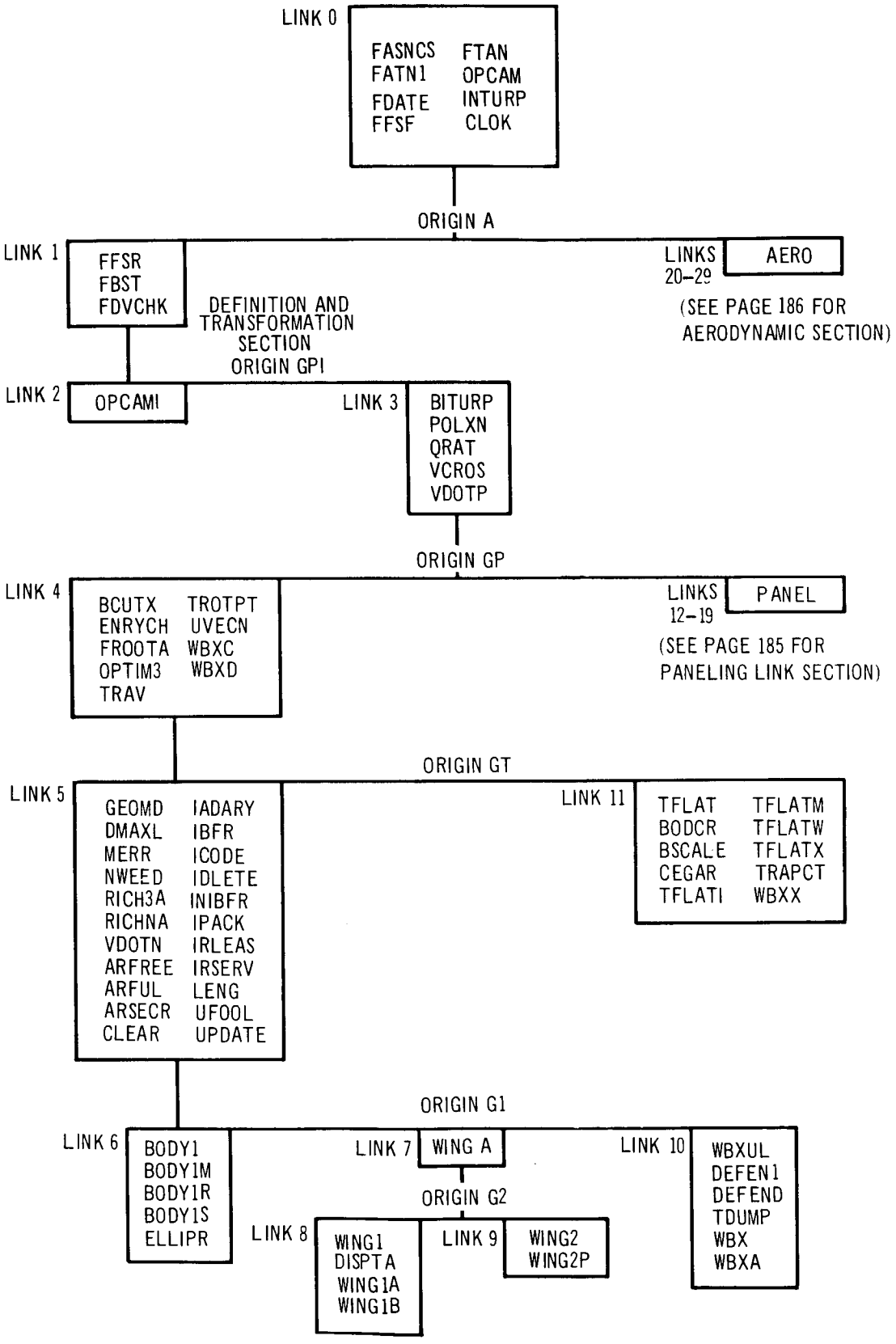
GEOMETRY PANELING SECTION
FLOW CHART

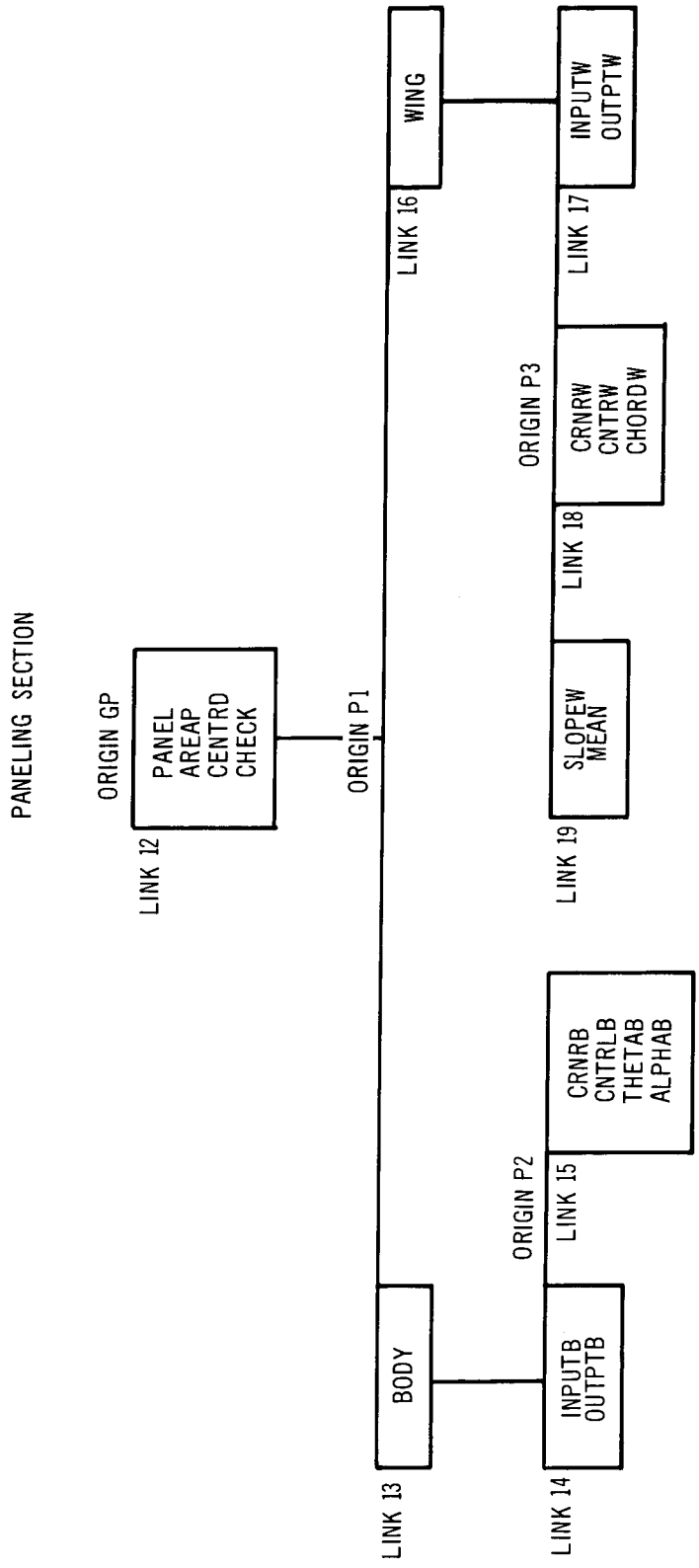


AERODYNAMIC SECTION
FLOW CHART

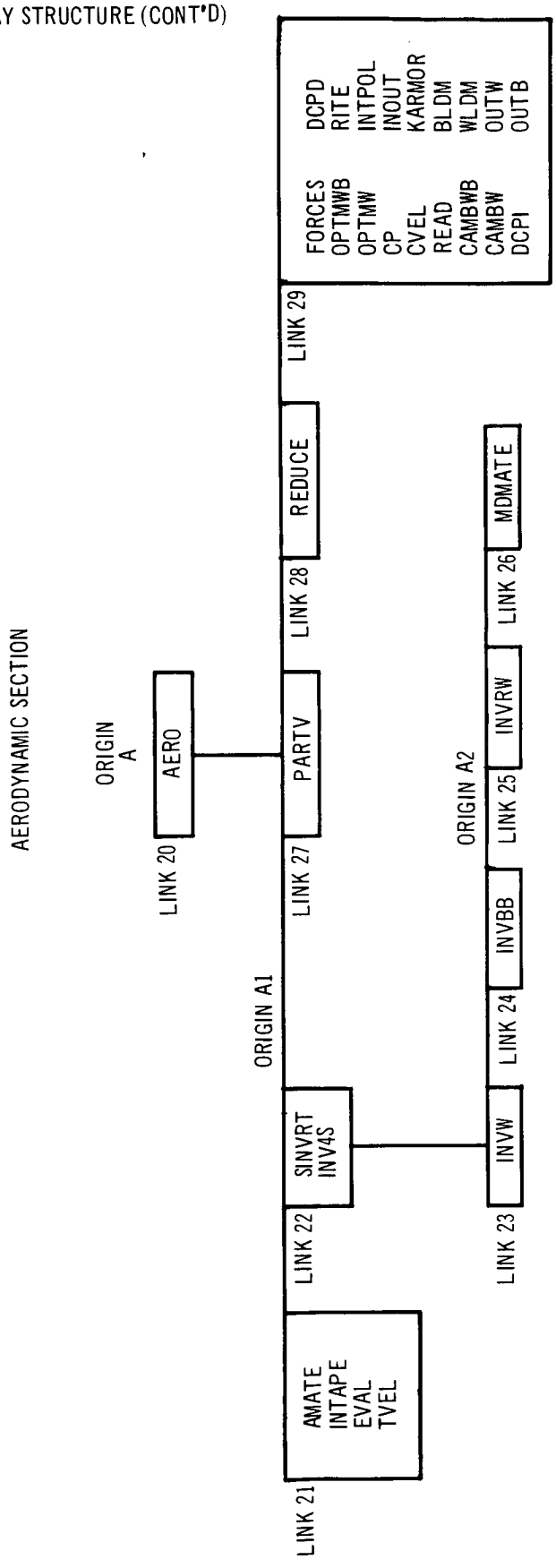


PROGRAM OVERLAY STRUCTURE




PANELING SECTION
ORIGIN GP
PANEL
AREAP
CENTRD
CHECK
LINK 12
ORIGIN P1
WING
LINK 16
INPUTW
OUTPTW
LINK 17
ORIGIN P3
CRNRW
CNTRW
CHORDW
LINK 18
SLOPEW
MEAN
LINK 19
BODY
LINK 13
ORIGIN P2
INPUTB
OUTPTB
LINK 14
LINK 15
CRNRB
CNTRLB
THETAB
ALPHAB

PROGRAM OVERLAY STRUCTURE (CONT'D)



The subroutines listed alphabetically in the accompanying index (with the exceptions noted) are discussed in this section.

Subroutine Index

Subroutine	Page No.	Subroutine	Page No.
ACOS	190	CNTRLW	235
AERO	191	CP	237
ALPHAB	193	CRNRB	239
AMATE	195	CRNRW	244
AREAP	197	CVEL	248
ARFREE	(See Appendix A)	DATE	250
ARFUL	(See Appendix A)	DCPD	251
ARSECR	(See Appendix A)	DCPI	252
ASIN	199	DEFEN1	254
ATN1	200	DEFEND	255
BCUTX	201	DISPTA	257
BITURP	202	DMAXL	258
BLDM	204	ELLIPR	259
BODCR	206	ENRYCH	260
BODY	208	EVAL	262
BODY1	211	FASNCS	265
BODY1M	214	FDATE	266
BODY1R	215	FFSF	267
BODY1S	217	FFSR	268
BSCALE	219	FORCES	270
CAMBW	220	FROOTA	273
CAMBWB	221	FSF	275
CEGAR	223	FSR	276
CENTRD	224	FTAN	277
CHECK	227	GEOMD	278
CHORDW	229	IADARY	(See Appendix A)
CLEAR	(See Appendix A)	IBER	(See Appendix A)

Subroutine	Page No.
IDLETE	(See Appendix A)
INIBFR	(See Appendix A)
INOUT	280
INPUTB	281
INPUTW	283
INTAPE	284
INTPOL	286
INTURP	288
INV4S	(See Appendix B)
INVBB	289
INVRW	291
INVW	292
IPACK	(See Appendix A)
IRLEAS	(See Appendix A)
IRSERV	(See Appendix A)
KARMOR	293
LENG	(See Appendix A)
MDMATE	298
MEAN	301
MERR	302
NWEED	303
OPCAM	304
OPCAMI	306
OPTIM3	308
OPTMW	310
OPTMWB	312
OUTB	314
OUTPTB	316
OUTPTW	318
OUTW	320
PANEL	321
PARTV	324
POLXN	326
QRAT	328

Subroutine	Page No.
READ	330
REDUCE	332
RICH3A	334
RICHNA	337
RITE	339
SINVRT	(See Appendix B)
SLOPEW	341
TAN	346
TDUMP	347
TFLAT	348
TFLAT1	354
TFLATM	356
TFLATW	357
TFLATX	358
THETAB	359
TRAPCT	361
TRAV	362
TROTPT	363
TVEL	364
UFOOL	(See Appendix A)
UPDATE	(See Appendix A)
UVECN	367
VCROS	368
VDOTN	369
VDOTP	370
WBX	371
WBXA	374
WBXC	375
WBXD	377
WBXUL	379
WBXX	383
WING	386
WING1	389
WING1A	392

Subroutine	Page No.
WING1B	395
WING2	397
WING2P	402
WINGA	403
WLDM	411

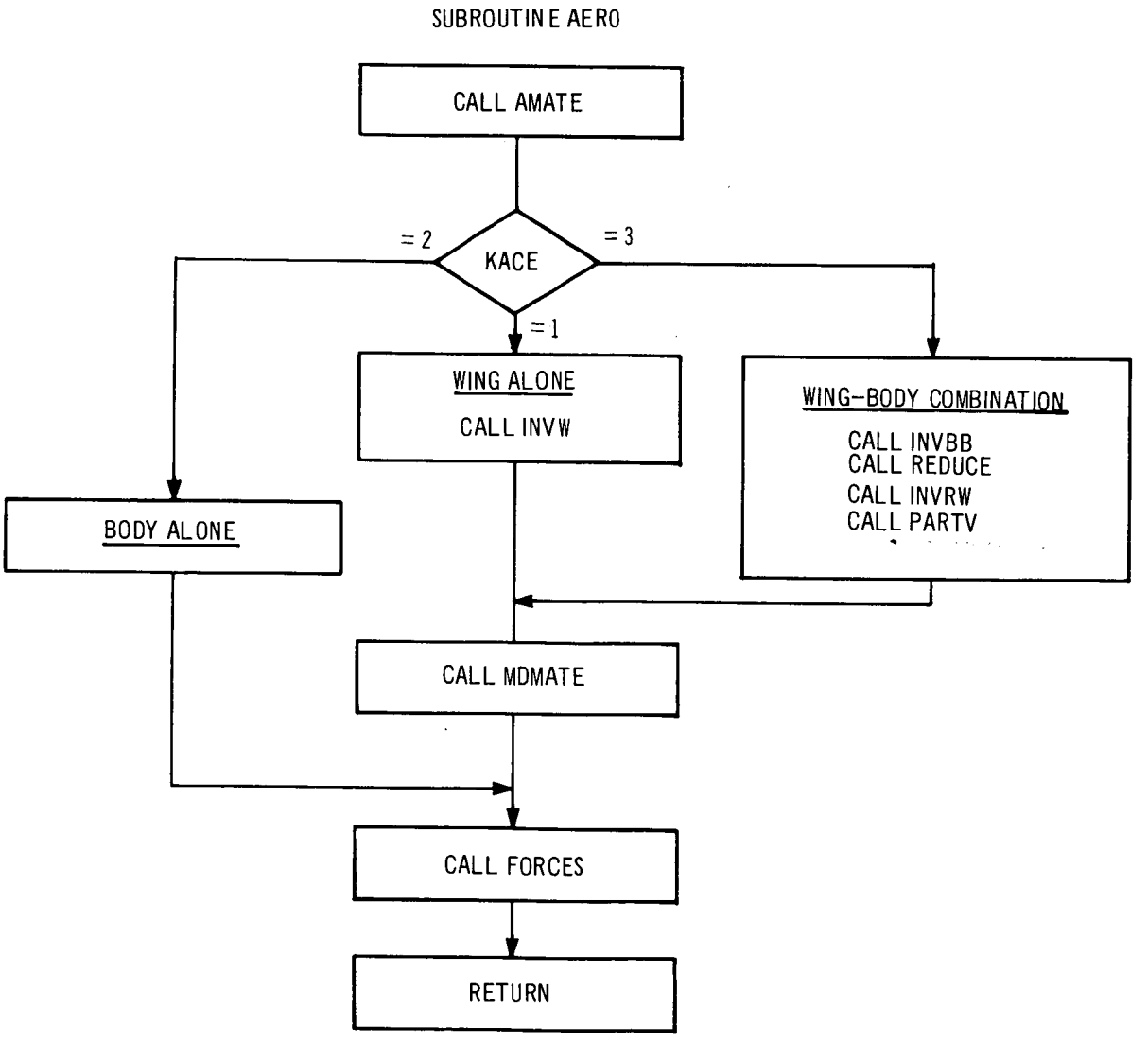
SUBJECT: FORTRAN IV Subroutine ACOS

(See subroutine FASNCS)

SUBJECT: FORTRAN IV Subroutine AERO

PURPOSE: To control the flow through the Aerodynamics section of the program.

METHOD: The flow to the main subroutines of the aerodynamics link is controlled by this subroutine for the three possible configurations (wing alone, body alone, and wing-body combinations).



USAGE: CALL AERO

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

Input: KACE = Indicates type of configuration.
= 1, wing alone
= 2, body alone
= 3, wing-body comgination

SUBPROGRAMS CALLED:

AMATE
INVW
INVBB
REDUCE
INVRW
PARTV
MDMATE
FORCES

ERROR RETURNS: None

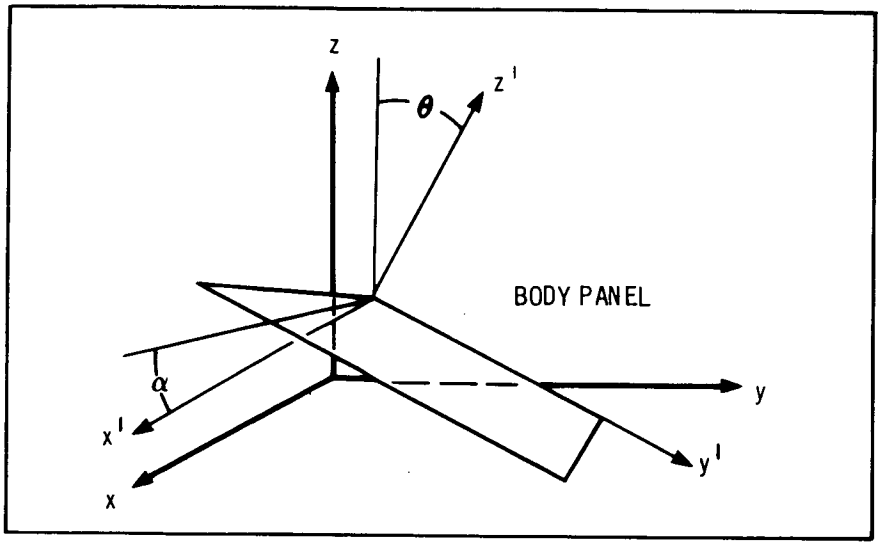
RESTRICTIONS: None

STORAGE: $54_{10} = 66_8$

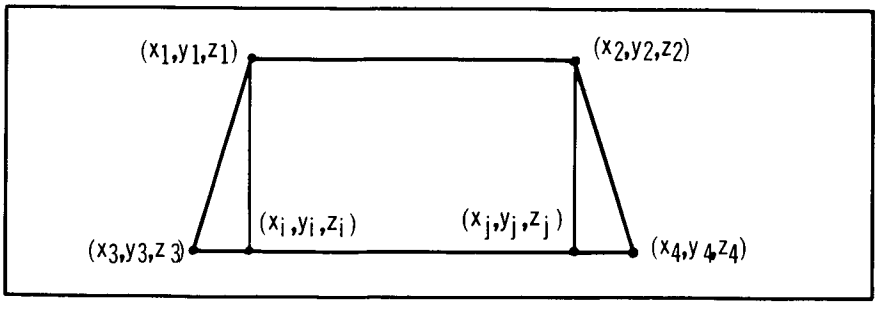
SUBJECT: FORTRAN IV Subroutine ALPHAB

PURPOSE: To calculate body panel alpha-incidence angles.

METHOD: The program calculates the panel alpha-incidence angle by use of a geometric function. For an arbitrary body panel,



the incidence angle, α, is the angle of the panel in the x', y', z' coordinate system of the panel as viewed down the y'-axis. The x'-axis of this system is parallel to the x-axis of the body coordinate system, though this is not true for the y' and z' axes for a panel of nonzero θ-inclination. For a body panel defined as follows,



the α-incidence angle is calculated by the following formula, which involves the inclination angle θ,

$$\alpha = \tan^{-1}\left[\frac{(z_1 - z_i)\cos\theta - (y_1 - y_i)\sin\theta}{(x_1 - x_i)}\right]$$

USAGE:

COMMON (See subroutine OPCAMI for unlabeled COMMON description)

COMMON /COM1/ (See subroutine PANEL)

Input: NPER1
NPLN1

COMMON /COM2/ (See subroutine BODY)

Input: KPANEL
XCOR
YCOR
ZCOR
XINT
YINT
ZINT
THETA

Output: ALPHA

CALL ALPHAB

SUBPROGRAMS CALLED:

ATAN
SIN
COS
} Built-in functions

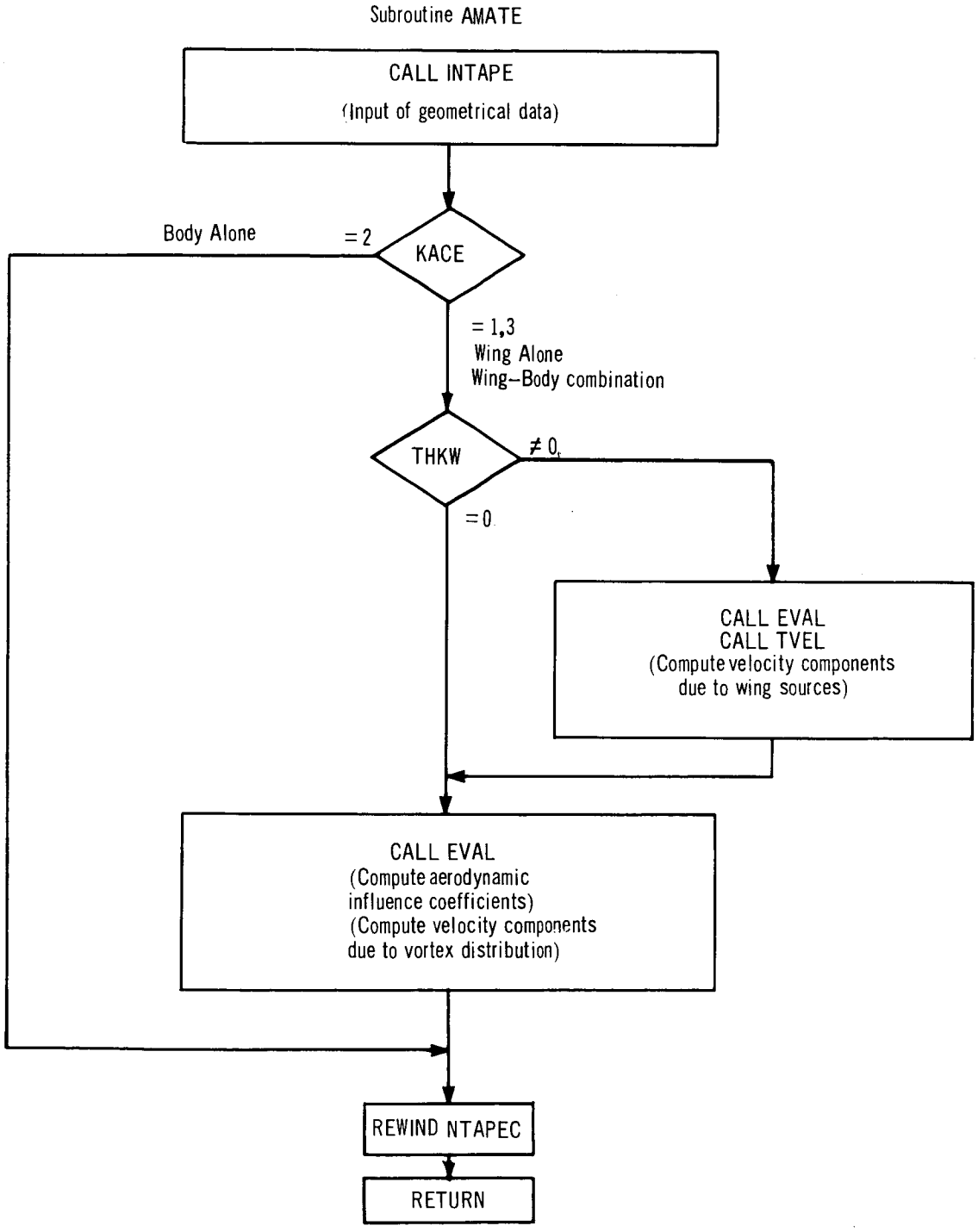
ERROR RETURNS: None

STORAGE: $362_{10} = 552_8$

SUBJECT: FORTRAN IV Subroutine AMATE

PURPOSE: A subroutine to control the flow through a section of the aerodynamic links of the program.

METHOD: The flow is illustrated in the following diagram.



USAGE: CALL AMATE

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

/BLOCK/ X(200, 3, 4), Y(200, 3, 4), Z(200, 3, 4), XBAR(20
YBAR(200), ZBAR(200), XC(200), YC(200), ZC(200), THETA(200), ALPHAS(200), AREA(200), CHORD(200), NPART(200), NPANEL, ISYM, A(200), U(200), V(200), VPM(200), VV(200), W(200), WPM(200), WW(200), VPMM(20
WPMM(200), ALPHAC(100), ALPHAT(100)

Input: KACE = 1, indicates wing-only case.

= 2, indicates body-only case.

= 3, indicates wing-body case.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

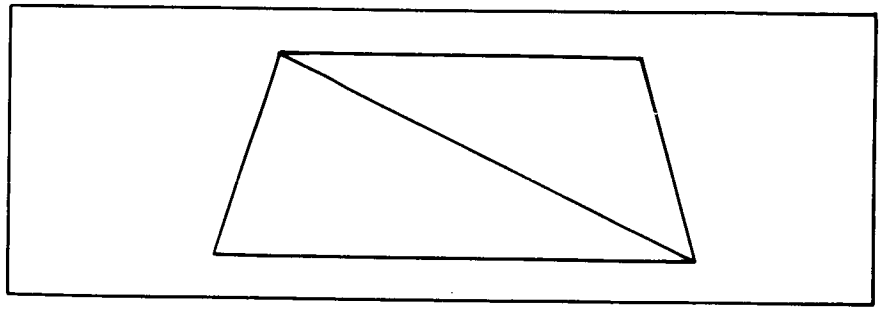
RESTRICTIONS: None

STORAGE: $11671_{10} = 26627_8$

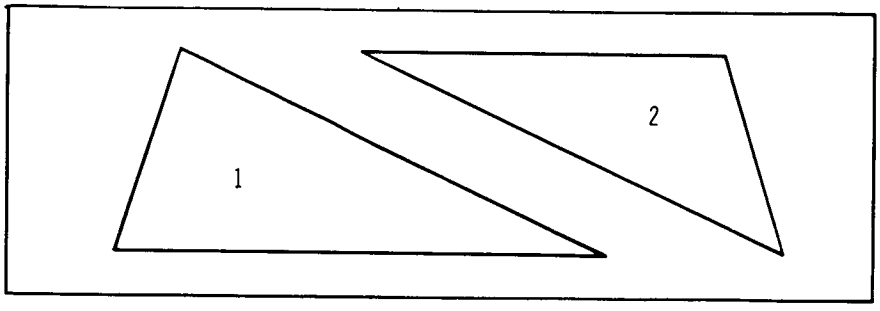
SUBJECT: FORTRAN IV Subroutine AREAP

PURPOSE: To calculate the area of a panel.

METHOD: Area S of a panel is obtained by dividing panel into two triangular subpanels, using a vector cross-product routine to calculate the area of each subpanel and summing these two results for the area of the panel. For example, a quadrilateral panel,



is divided into two subpanels, thus:



and areas S_1 and S_2 of each subpanel are calculated and summed for panel area S.

$$S_1 + S_2 = S$$

The subroutine also calculates ratio R of area S_1 of subpanel 1, to panel area S,

$$R = \frac{S_1}{S}$$

This value is used in other subroutines.

USAGE: DIMENSION X(16, 16), Y(16, 16), Z(16, 16), S(15, 15), R(15, 15)

$\left.\begin{matrix} X \\ Y \\ Z \end{matrix}\right\} =$ x, y, and z coordinates of panel corner points.

S = Panel area.

R = Ratio of subpanel area to panel area.

CALL AREAP (X, Y, Z, S, R, NC, NR)

Input: X, Y, Z (see above)

NC = Number of panel columns.

NR = Number of panel rows.

Output: S, R (see above)

SUBPROGRAM CALLED: VCROS

ERROR RETURNS: None

STORAGE: $239_{10} = 357_8$

SUBJECT: FORTRAN IV Subroutine ASIN

(See subroutine FASNCS)

SUBJECT: FORTRAN IV Function ATN1

PURPOSE: To compute the arctangent, in radians.

METHOD: This function has been replaced by the standard built-in function ATAN2, with a $NAME control card.

SUBJECT: FORTRAN IV Subroutine BCUTX

PURPOSE: To find the intersections of a set of body meridian lines with a plane normal to the x-axis.

METHOD: Subroutine POLXN is used to find the intersection of each meridian line with the plane.

USAGE:

```
DIMENSION  B(1), A(2, N)
CALL BCUTX (B, N, X, EP, NA, A)
```

Input:
- B = Array of body meridian line points, with header (see subroutine WBXUL).
- N = Number of body meridian lines.
- X = Body station. The plane will be taken as x = X.
- EP = A tolerance used by POLXN in finding intersections.

Output:
- NA = Number of points in A. NA $\leq$ N.
- A = Array of points (y_i, z_i) in section. These points are in 1-to-1 correspondence with meridian lines only if each meridian line intersects the plane.

SUBPROGRAM CALLED: POLXN

ERROR RETURNS: None

RESTRICTIONS: N must be greater than zero. If a meridian line has multiple plane intersections, the first intersection found will be used.

STORAGE: $133_{10} = 205_8$

SUBJECT: FORTRAN IV Subroutine BITURP

PURPOSE: Given a table of y versus x and a value of x, to find the corresponding value of y by either linear or biquadratic interpolation.

METHOD: Let n be the number of points (x_i, y_i) in the given table. Let X be the given value of x for which the corresponding Y is desired. If $n = 2$, the linear interpolation mode is selected (even if biquadratic interpolation was requested). If $x_1 < x_n$, the table is searched from top to bottom; otherwise the search is from bottom to top. If X is outside the table range, an error code is set and Y is returned as y_1 or y_n (depending on whether X is closer to x_1 or x_n). If $X = x_i$ then no interpolation is necessary and $Y = y_i$. Otherwise, the table interval that contains X is located. For linear interpolation, the calculation of Y is trivial.

Y may be found by biquadratic interpolation if X does not lie in the first or last table interval. For illustration of the method, assume that $x_2 < X < x_3$. Fit a quadratic $Q_2(x)$ to the points (x_i, y_i), $i = 1, 2, 3$ and another quadratic $Q_3(x)$ to the points (x_i, y_i), $i = 2, 3, 4$. Find the slopes $Q'_2(x_2)$ and $Q'_3(x_3)$. Fit a cubic to the points (x_2, y_2) and (x_3, y_3) with the corresponding slopes $Q'_2(x_2)$ $Q'_3(x_3)$. Evaluate the cubic at $x = X$ to find Y.

However, Y is actually computed by the following interpolating formula (which has a continuous first derivative), mathematically equivalent to the above method:

$$Y = \frac{(x_3 - X)Q_2(X) + (X - x_2)Q_3(X)}{x_3 - x_2}$$

$Q_2(X)$ and $Q_3(X)$ are found by subroutine QRAT.

If X lies in the first or last interval of the table, biquadratic interpolation cannot be used. Y is therefore computed by quadratic interpolation.

USAGE:

```
DIMENSION  X(1), Y(1), NU(3)
CALL BITURP (X, Y, I, N, K, XI, YI, NU)
```

Input: X = Location of the first x-coordinate in the table.

Y = Location of the first y-coordinate in the table.

I = Skip number; that is, the spacing of table values in the X and Y arrays. For example, I = 1 if X = (x_1, x_2, ...) and Y = (y_1, y_2, ...); however, if x and y are stored in a single array, say W = (x_1, y_1, x_2, y_2, ...), then I = 2 and the subroutine call would be CALL BITURP (W, W(2), 2, ...).

N = Number of points in the table.

K = Interpolation code (K = 1 if linear interpolation; K = 2 if biquadratic interpolation).

XI = The given value of x for which the corresponding value of y is desired.

NU = Error indicator array: NU(1) is not used, on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) ≤ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = NU(3) - 1. Then if NU(3) > 0 and NU(2) > 0, an error message is written.

Output: YI = Interpolated value of y (if NU(1) = 0, 1, or 2).

NU = Error indicator array: NU(1) = 0 if success; = -1 if I ≤ 0; = -2 if N < 2; = -3 if $x_1 = x_n$; = 1 or 2 if X is outside the table range; = 3 or 4 if a quadratic could not be formed (because two points near the line x = X have the same x-coordinate. NU(2) is not an output. NU(3) is reduced by one if an error was detected.

SUBPROGRAM CALLED: QRAT

RESTRICTIONS: BITURP does not check the X array to see that it is monotonic. Weird results may be obtained if it is not monotonic in the neighborhood of the interpolating interval for biquadratic interpolation.

STORAGE: $375_{10} = 567_8$

SUBJECT: FORTRAN IV Subroutine BLDM

PURPOSE: To compute the coefficients of lift, drag, and pitching moment for the region of the body represented by panels.

METHOD: The force normal to the panel is given as the product of the dynamic pressure, surface pressure coefficient, and panel area:

$$F_i = q \cdot C_{p_i} \cdot A_i$$

Resolving into the components normal and parallel to the free stream yields the lift and drag

$$L_i = - F_i \cdot \cos \theta_i$$

$$D_i = F_i \cdot n_i$$

where n_i is the component of velocity normal to the x-axis and θ_i is the angle between the plane of the panel and a panel parallel to the x-y plane.

The moment of force with respect to a point (x, 0, z) is given by

$$M_i = - L_i (\bar{x}_i - x) + D_i (\bar{z}_i - z)$$

where $\bar{x}_i$, $\bar{z}_i$ are the x and z coordinates of the panel centroid.

Finally, the coefficients of lift, drag, and moment on the body panels are given by

$$C_L = \frac{1}{q \cdot S_W} \sum_{i=1}^{NM} L_i$$

$$C_D = \frac{1}{q \cdot S_W} \sum_{i=1}^{NM} D_i$$

$$C_M = \frac{1}{q \cdot S_W} \sum_{i=1}^{NM} M_i$$

where S_W is the wing reference area, and NM is the number of body panels.

USAGE: CALL BLDM(NM, XP, ZP, RFAREA, AREA, XBAR, ZBAR, ALPHAM, THETAM, CPM, CL, CD, CM)

DIMENSION AREA(NM), XBAR(NM), ZBAR(NM), ALPHAM(NM), THETAM(NM), CPM(NM)

Input:

NM	=	Number of body panels.
XP	=	x-coordinate of the point about which the pitching moments are to be calculated.
ZP	=	z-coordinate of the point about which the pitching moments are to be calculated.
RFAREA	=	Wing reference area.
AREA	=	Array of body panel areas.
XBAR	=	Array of body panel centroid x-coordinates.
ZBAR	=	Array of body panel centroid z-coordinates.
ALPHAM	=	Array of the velocity components normal to the x-axis.
THETAM	=	Array of angles between the plane of the panel and a plane parallel to the x-y plane.
CPM	=	Array of panel surface pressure coefficients.

Output:

CL	=	Coefficient of lift.
CD	=	Coefficient of drag.
CM	=	Coefficient of pitching moment.

SUBPROGRAM CALLED: COS (Built-in function)

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $118_{10} = 166_8$

SUBJECT: FORTRAN IV Subroutine BODCR

PURPOSE: To find the area, equivalent radius, and centroid of a body section.

METHOD: It is assumed that the body is symmetrical about a plane parallel to the x-z plane, and that only one side of the body is given. The body is specified by a set of meridian lines (subroutine WBXUL). Subroutine BCUTX is used to find the y-z coordinates of the meridian lines at a given value of x. These coordinates form the two-dimensional points of a body section. A test is made to see that the y-coordinates of the first and last points are the same within a given tolerance. The average of these two values is used for the centroid y-coordinate. Subroutine ENRYCH is called and, if a given quantity CHD is greater than zero, additional points on the section are interpolated. The centroid and area of the section are found by subroutine CEGAR. The area is then doubled to make it apply to the entire body section. If the area is smaller than a given tolerance, the z-coordinate of the centroid is taken as the average of the z-coordinates of the first and last points in the section; otherwise, the value found by CEGAR is used. The z-coordinate of the centroid is set to zero if it is smaller than a given tolerance. The section radius is found by averaging the distances from the centroid to each meridian line; if the radius is less than a given tolerance, it is set to zero.

USAGE: DIMENSION B(1), EP(5), S(2, N), E(MAXE), NU(3)

CALL BODCR (B, N, X, EP, CHD, MAXE, S, E, NE, Y, Z, AREA, RAD, NU)

Input:	B	= Body meridian line array, with header (see subroutine WBXUL).
	N	= Number of meridian lines.
	X	= Body station (x-coordinate section).
	EP	= Array of tolerances.
	CHD	= Tolerance that regulates the number of points interpolated by ENRYCH (affects only AREA and Z).
	MAXE	= Length of array E.

	NU	= Error indicator array. NU(1) is not used on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) ≤ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written.
Output:	S	= Array of points (y_i, z_i) in section.
	E	= Scratch array.
	NE	= Number of elements in enriched array.
	Y	= y-coordinate of centroid.
	Z	= z-coordinate of centroid.
	AREA	= Area of entire section (twice area of the half-section).
	RAD	= Equivalent radius.
	NU	= Error indicator array. NU(1) = 0 if success.

SUBPROGRAMS CALLED

ENRYCH
CEGAR
SQRT (Built-in function)
BCUTX

ERROR RETURNS: A message is written if an error is detected. NU = 1 if the number of points in the section (found by BCUTX) is less than N or less than 2. An error of NU = 2, 4, or 5 is from ENRYCH. If an error code of 3 was returned by ENRYCH, NU is set to zero but a message is written.

RESTRICTIONS: CHD should be zero if the section is symmetrical about a horizontal plane or if there are less than eight meridian lines.

STORAGE: $486_{10} = 746_8$

SUBJECT: FORTRAN IV Subroutine BODY

PURPOSE: To serve as a control program for the body paneling subroutines.

METHOD: Program uses a series of FORTRAN IV language "CALL" statements to call the required body paneling subroutines.

The program contains the following labeled COMMON statement that occurs in all lower level body paneling subroutines,
COMMON /COM2/ NPLNB, NPLNW, JLEAD, JTRAIL, IMID, NPTS(16), X(16, 90), Y(16, 90), Z(16, 90), XCEPT(21), XCEPTB(21), XCEPTW(16), YCEPTW(16), ZCEPTW(16), CODEBW(16), KPANEL(15, 20), XCOR(16, 21), YCOR(16, 21), ZCOR(16, 21), XINT(15, 20, 2), YINT(15, 20, 2), ZINT(15, 20, 2), XCEN(15, 20), YCEN(15, 20), ZCEN(15, 20), XCON(15, 20), YCON(15, 20), ZCON(15, 20), AREA(15, 20), ARAT(15, 20), THETA(15, 20), ALPHA(15, 20), CHORD(15, 20)

NPLNB = Number of cutting planes to intersect the body fore and aft of the body-wing intersection region.

NPLNW = Number of cutting planes to intersect the body in the body-wing intersection region.

JLEAD, JTRAIL, IMID = Codes for internal program control.

NPTS = Number of points on the successive body meridian lines that define the body.

X, Y, Z = x, y, and z coordinates of points on body meridian lines.

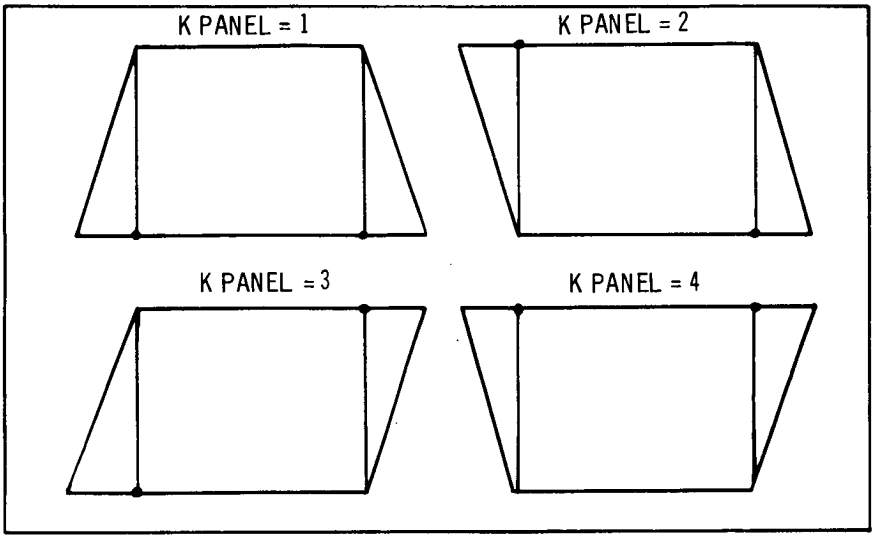
XCEPT = x intercepts of all body cutting planes.

XCEPTB = x intercepts of those body cutting planes that intersect fore and aft of the body-wing intersection region.

XCEPTW, YCEPTW, ZCEPTW = x, y, and z coordinates of the intersections with the body of those wing percent lines that define body cutting planes.

CODEBW = Codes that indicate wing percent lines that are to define body cutting planes in body-wing intersection region.

KPANEL = Code for body panel type. If KPANEL (I, J) = 1, both additional subpanel corner points are located on panel trailing edge. If KPANEL(I, J) = 2, inboard point is on leading edge, and outboard point is on trailing edge. If KPANEL (I, J) = 3, inboard point is on trailing edge and outboard point is on leading edge. If KPANEL(I, J) = 4, both points are on leading edge.



XCOR, YCOR, ZCOR = x, y, and z coordinates of body panel corner points.

XINT, YINT, ZINT = x, y, and z coordinates of additional subpanel corner points.

XCEN, YCEN, ZCEN = x, y, and z coordinates of body panel centroid.

XCON, YCON, ZCON = x, y, and z coordinates of body panel control point.

AREA = Body panel area.

ARAT = Ratio of subpanel area to panel area.

THETA = Body panel theta-inclination angle.

ALPHA = Body panel alpha-incidence angle.

CHORD = Body panel streamwise chord length.

USAGE: COMMON (See subroutine OPCAMI for description of unlabeled COMMON)

CALL BODY

SUBPROGRAMS CALLED:

INPUTB
CRNRB
AREAP
CENTRD
CNTRLB
THETAB
ALPHAB
OUTPTB

ERROR RETURNS: Error message indicates whether error occurred in calculation of body panel geometry.

STORAGE: $152_{10} = 230_8$

SUBJECT: FORTRAN IV Subroutine BODY1

PURPOSE: To read data cards that specify a body surface and generate three-dimensional points on body meridian lines.

METHOD: A body surface is specified by a set of defining sections normal to the X axis. Points on defining sections are given in polar coordinate form (ρ, θ) or in rectangular coordinates (y, z). The number of points must be the same for all sections (except for zero-area sections). For convenience, several options are available for specifying section points (see Input Data Format, 3.3, card 6D).

Three-dimensional points on the ith body meridian line are formed by selecting the ith point at each defining section. Additional points are interpolated (if requested) so that straight-line connections between adjacent points will represent a smooth curve as closely as specified.

USAGE:

```
DIMENSION   DAT(2), B(1), AXIS(2), TITLE(12), NU(3)
LOGICAL   LGDEF(3, 6)
CALL BODY1 (DAT, LI, LO, B, LSTA, NSTA, LBPL,
            MBPL, NBPL, LTHETS, AXIS, TITLE,
            LGDEF, NU)
```

Input:	DAT	= Date (alphameric).
	LI	= Input tape number.
	LO	= Output tape number.
	B	= Buffer for variable-length arrays.
	NU	= Error indicator array: NU(1) is not used, on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written.
Output:	B	= Buffer for variable-length arrays.
	LSTA	= The set of body-defining stations (x-values) starts in B(LSTA).

NSTA	=	Number of body-defining stations.
LBPL	=	The body meridian line array, with header, starts in B(LBPL). See Appendix C for the array format.
MBPL	=	Number of meridian lines.
NBPL	=	Number of cells in meridian-line array.
LTHETS	=	The standard set of θ values (in degrees) corresponding to meridian lines, starts in B(LTHETS).
AXIS	=	y and z coordinates of the body main axis.
TITLE	=	Body title (alphameric).
LGDEF	=	LGDEF(1, 3) is set to .TRUE., LGDEF(2, 3) is set to .TRUE. if the body meridian lines are successfully computed.
NU	=	Error-indicator array. NU(1) is zero if no errors were detected. NU(3) may have been changed if an error was found (see INPUT, above).

SUBPROGRAMS CALLED:

RICHNA
BODY1M
BODY1R
MERR
IDLETE
IRLEAS
IRSERV
IPACK

(See Appendix A)

ERROR RETURNS: Function MERR is used to write an error message "ERROR i, CODE j IN SUBROUTINE BODY1 DURING GEOMETRIC DEFINITION" if an error is detected.

i	j	Explanation
1	0	The number of defining sections (BNS, card 4D) or the number of meridian lines (BTHETA) is less than 2 or greater than 150.

i	j	Explanation (cont)
2	0	BNS * BTHETA is so large that the storage required for the meridian-line points (even with no enriching) is greater than the storage available in array B.
3	1	The code for the type of body section (SCODE, card 6D) is less than 0 or greater than 6.
3	2	SCODE = 3 (elliptical section), but a semi-axis is zero.
4	k	Meridian lines have been formed, but error k was detected in subroutine RICHNA.
5	0	Machine or program error of undetermined origin.

RESTRICTIONS: The storage buffer, B, for variable-length arrays must have been initialized by subroutine INIBFR (Appendix A) or its equivalent.

STORAGE: $568_{10} = 1070_8$

SUBJECT: FORTRAN IV Subroutine BODY1M

PURPOSE: To write body meridian line points on an output tape.

METHOD: This is primarily an output routine for BODY1. The meridian lines are given in a single array with the format described in Appendix C. The polar coordinates of each point are computed in a plane normal to the x axis with polar origin at the body axis. Points that lie in a body defining section (within 0.0001) are identified by an asterisk.

USAGE:

```
DIMENSION  B(1), AXIS(2), TITLE(12), DAYT(2), STA(1)
CALL BODY1M (B, NB, AXIS, TITLE, DAYT, LO, STA)
```

Input:

B	=	Array of meridian lines, with header.
NB	=	Number of meridian lines.
AXIS	=	y, z coordinates of main body axis.
TITLE	=	Title (alphameric).
DAYT	=	Date (alphameric).
LO	=	Output tape number.
STA	=	Array of defining stations X-values).

Output: Printout on tape LO.

SUBPROGRAMS CALLED:

ATN1
SQRT (Built-in function)
TRAV

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $373_{10} = 565_8$

SUBJECT: FORTRAN IV Subroutine BODY1R

PURPOSE: To read and write data that specify body defining sections, and form three-dimensional points on body meridian lines.

METHOD: The number of body-defining sections, number of meridian lines, the Y and Z coordinates of the body axis, and a standard set of angles are given. Each section lies in a station plane (normal to the X axis). A meridian line of three-dimensional points on the body surface is formed by selecting corresponding points from each station. Let NPT = number of meridian lines = number of two-dimensional points per section = number of angles in the standard set.

Subroutine BODY1S is called for each section to read data cards and compute NPT three-dimensional points on the body surface. These points are then stored in meridian-line order. Results for each section are written on an output tape.

USAGE:

```
DIMENSION  BAXIS(2), TITLE(12), DAT(2), THETS(NPT),
           THETA(NPT), RHO(NPT), PT(3, NPT),
           BPL(3, NSTA, NPT), STA(NSTA)

CALL BODY1R (LI, LO, NPT, NSTA, BAXIS, TITLE,
             DAT, THETS, THETA, RHO, PT, BPL,
             STA, NU)
```

Input:

LI	=	Input tape number.
LO	=	Output tape number.
NPT	=	Number of meridian lines.
NSTA	=	Number of defining sections (stations).
BAXIS	=	Y, Z coordinates of main body axis.
TITLE	=	Title (alphameric).
DAT	=	Date (alphameric).
THETS	=	Array of standard angles, in degrees.

Scratch: THETA, RHO, PT = Storage for θ, ρ, and point coordinates (used at each section).

Output:

BPL	=	Points on meridian lines.
STA	=	Array of defining stations (X coordinates).

NU = Error indicator, which is zero for success.

SUBPROGRAM CALLED: BODY1S

ERROR RETURNS: If NU $\neq$ 0, see BODY1S.

RESTRICTIONS: Tape LI must be properly positioned and contain NSTA sets of data for BODY1S.

STORAGE: $368_{10} = 560_8$

SUBJECT: FORTRAN IV Subroutine BODY1S

PURPOSE: To read body-defining section data cards and compute points on the body section.

METHOD: Data cards are described in section 3.3. The code for section type is checked for validity; then the indicated operation is carried out.

USAGE:

```
DIMENSION   BAXIS(2), THETS(NPT), SECAX(2),
            THETA(NPT), RHO(NPT), PT(3, NPT)

CALL BODY1S (LI, BAXIS, NPT, THETS, SECAX, SCODE,
             THETA, RHO, PT, NU)
```

Input:

- LI = Input tape number.
- BAXIS = Y, Z coordinates of the main body axis.
- NPT = Number of points in section.
- THETS = Array of angles, in degrees, which are used for SCODE = 1, 2, and 3.
- PT = See OUTPUT, below; used as input only if SCODE = 0.

Output:

- SECAX = Y, Z coordinates of the section origin (ρ, θ origin).
- SCODE = Code for type of section (read from card).

  0. This section is identical to the previous section, except for X-coordinates. Store the station value (read from card) as the X-coordinate for each point in PT.
  1. Read NPT values of ρ from cards. Use with the THETS array to compute Y, Z coordinates of points on the section.
  2. Circular section. Use the radius (read from card) as ρ, and THETS to compute section points.
  3. Section is a rectangular ellipse. Use the semi-axes (read from card) to construct ellipse. Find

ρ's corresponding to THETS. Construct points on section.

4. Circular section, but read NPT values of θ from cards instead of using THETS.

5. Read NPT values of ρ from cards, then read NPT values of θ from cards.

6. The y,z coordinates of the section points are given. Find corresponding ρ, θ coordinates; but if any ρ is zero, use corresponding θ from THETS.

THETA = Array of θ values (in degrees) corresponding to each section point (same as THETS for SCODE = 1, 2, 3).

RHO = Array of ρ values.

PT = Array of points (x, y, z coordinates) in section.

NU = Error indicator, which is zero for success.

SUBPROGRAMS CALLED:

ATN1
UVECN
ELLIPR
SIN (Built-in function)
TRAV
COS (Built-in function)

ERROR RETURNS:

NU = 1 if SCODE < 0 or SCODE > 6.
NU = 2 if SCODE = 3 and a semi-axis is zero.

RESTRICTIONS:

Input tape LI is assumed to be correctly positioned and to contain valid data. Angles are measured from the vertical, and it is assumed that $0 \leq \theta \leq 180$. The ρ-θ origin is at SECAX.

STORAGE:

$439_{10} = 667_8$

SUBJECT: FORTRAN IV Subroutine BSCALE

PURPOSE: To scale body meridian lines so that each line extends from $x = 0$ to $x = x_E$, where x_E is given. The y and z coordinates are unchanged.

METHOD: The x-coordinates of the points on each meridian line are scaled as follows:

$$x' = (x - x_1)\, x_E/(x_L - x_1)$$

where x_1 = x of first point, x_L = x of last point.

USAGE:

```
DIMENSION  B(1)
CALL BSCALE (B, NB, XEND, NU)
```

Input: B = Meridian line array (see subroutine TFLAT1).

NB = Number of meridian lines.

XEND = x_E.

Output: B = Scaled meridian line array.

NU = Error indicator. NU = 0 if success. NU = k if $x_L - x_1 = 0$ for the kth meridian line.

SUBPROGRAM CALLED: None

ERROR RETURNS: See Output

RESTRICTIONS: None

STORAGE: $125_{10} = 175_8$

SUBJECT: FORTRAN IV Subroutine CAMBW

PURPOSE: To compute the velocity components normal to the x-axis on the wing panels, given the matrix of the aerodynamic influence coefficients and the pressure difference across the panels.

METHOD: The velocity components are given by

$$n_{W_i} = \sum_{i=1}^{NW} \sum_{j=1}^{NW} A_{WW_{ij}} \cdot p_{W_j}$$

where A_{WW} is the matrix of aerodynamic influence coefficients (influence on the wing, due to the wing) and p_W is the pressure difference across each wing panel.

USAGE: CALL CAMBW (NW, NT, A, CL, ALPHAW)

DIMENSION A(NW), CL(NW), ALPHAW(NW)

Input:	NW	=	Number of wing panels
	NT	=	Logical tape number on which A_{WW} is stored
	A	=	Dummy array used by the subroutine
	CL	=	Array of pressure differences
Output:	ALPHAW	=	Array of velocity components

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $90_{10} = 132_8$

SUBJECT: FORTRAN IV Subroutine CAMBWB

PURPOSE: In the presence of a body, to compute the velocity components on the wing, normal to the x-axis.

METHOD: The velocity component on the wing panel i is given by

$$n_{W_i} = \sum_{j=1}^{NW} A_{R_{ij}} \cdot p_{W_j} + \sum_{j=1}^{NB} D_{ij} \cdot n_{B_j}$$

where A_R = The matrix of the reduced aerodynamic influence coefficients,

p_W = The pressure difference across wing panels,

D = The matrix formed by the product.

$$A_{WB} \cdot A_{BB}^{-1}$$

where A_{WB} = A partition of the aerodynamic influence coefficients matrix resulting from the influence of the body on the wing,

A_{BB} = The partition resulting from the influence of the body on the body,

n_B = The velocity component on the body normal to the x-axis.

USAGE: DIMENSION A(NW), CLW(NW), ALPHAB(NB), ALPHAW(NW)

CALL CAMBWB (NB, NW, NTAPEX, NTAPEY, A, CLW, ALPHAB, ALPHAW)

Input:

NB	=	Number of body panels.
NW	=	Number of wing panels.
NTAPEX	=	Logical tape number on which A_R is stored.
NTAPEY	=	Logical tape number on which D is stored.
A	=	Dummy array used by the subroutine
CLW	=	Array of the pressure differences across wing panels.
ALPHAB	=	Array of velocity components on the body, normal to the x-axis.

Output: ALPHAW = Array of velocity components on the wing, normal to the x-axis.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $144_{10} = 220_8$

SUBJECT: FORTRAN IV Subroutine CEGAR

PURPOSE: To find the area and the centroid (center of gravity) of a polygon.

METHOD: Given n points (x_i, y_i: $i = 1, \ldots, n$) that are the vertices of a polygon, the area of the polygon is given by the magnitude of A, where

$$A = 1/2 \sum_{i=1}^{n} (y_{i+1} + y_i)(x_{i+1} - x_i)$$

where x_{n+1}, y_{n+1} are set equal to x_1, y_1 by the subroutine to ensure that the polygon is closed. If any two or more of the points (x_i, y_i) coincide, no difficulty will result in the calculation of the centroid, and the area as long as the resulting polygon is a simple closed curve.

The coordinates of the centroid are given by,

$$\bar{x} = \sum_{i=1}^{n} (x_{i+1}^2 + x_i^2 + x_{i+1} x_i)(y_{i+1} - y_i) / (-6A), \text{ and}$$

$$\bar{y} = \sum_{i=1}^{n} (y_{i+1}^2 + y_i^2 + y_{i+1} y_i)(x_{i+1} - x_i) / (6A)$$

USAGE: DIMENSION P(2, n+1), CG(2)

CALL CEGAR (N, P, A, CG)

Input: N = the number of points, n.

P = the array of points (x_i, y_i) that are the vertices of the polygon.

Output: A = the magnitude of the area of the polygon.

CG = the centroid of the area, $(\bar{x}, \bar{y})$.

RESTRICTIONS: The polygon must be a simple closed curve, (i.e., the polygonal arc cannot cross itself).

ERROR RETURNS: If the area of the polygon is zero, then the subroutine sets the coordinates of the centroid equal to zero.

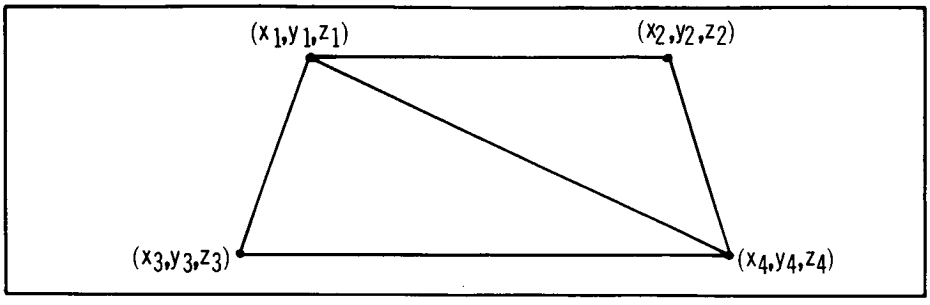
Because, in the calculation of $\bar{x}$ and $\bar{y}$, A is used as a divisor, considerable error could result in the values obtained for $\bar{x}$ and $\bar{y}$ when A is close to zero. Overflow could also occur for small values of A.

STORAGE: $229_{10} = 345_8$

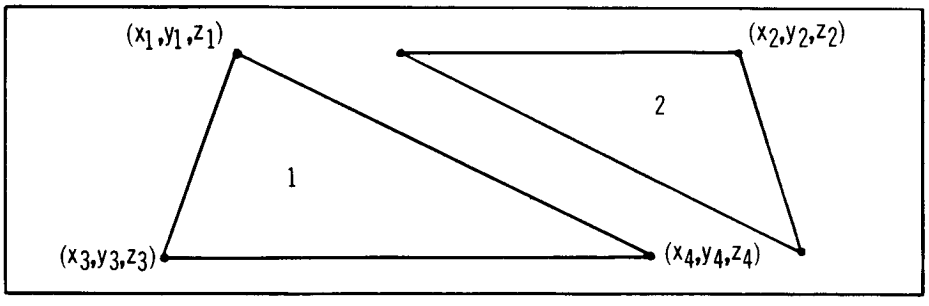
SUBJECT: FORTRAN IV Subroutine CENTRD

PURPOSE: To calculate coordinates of a panel centroid.

METHOD: Centroid coordinates of a panel are calculated by dividing panel into two subpanels, locating a centroid on each subpanel, and averaging coordinates of the two centroids with appropriate weighting factor. For example, a given quadrilateral panel:



is subdivided into two subpanels, thus:



and a centroid is calculated for each subpanel using standard formulae. For subpanel 1 of the above example,

$$\bar{x}_1 = x_1 + \frac{2}{3}\left(x_3 + \frac{1}{2}(x_4 - x_3) - x_1\right)$$

$$\bar{y}_1 = y_1 + \frac{2}{3}\left(y_3 + \frac{1}{2}(y_4 - y_3) - y_1\right)$$

$$\bar{z}_1 = z_1 + \frac{2}{3}\left(z_3 + \frac{1}{2}(z_4 - z_3) - z_1\right)$$

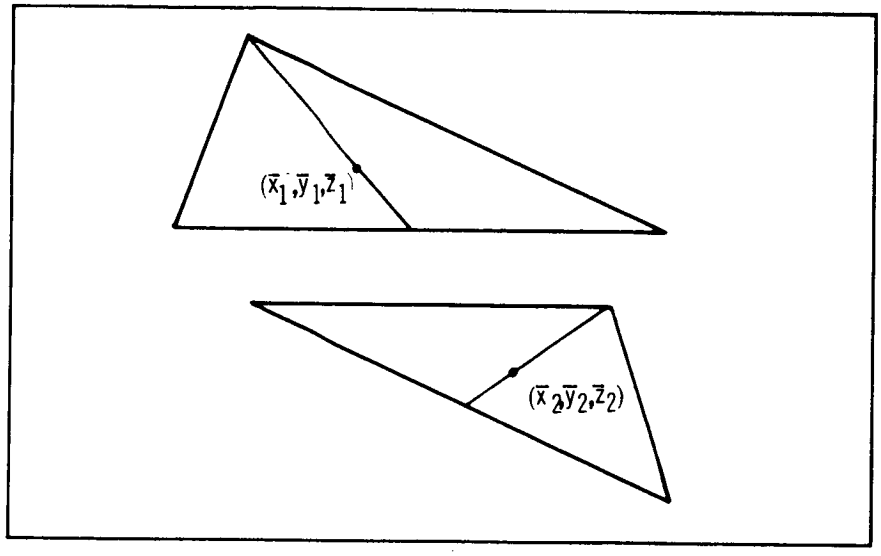
Similarly for subpanel 2,

$$\bar{x}_2 = x_2 + \frac{2}{3}\left(x_1 + \frac{1}{2}(x_4 - x_1) - x_2\right)$$

$$\bar{y}_2 = y_2 + \frac{2}{3}\left(y_1 + \frac{1}{2}(y_4 - y_1) - y_2\right)$$

$$\bar{z}_2 = z_2 + \frac{2}{3}\left(z_1 + \frac{1}{2}(z_4 - z_1) - z_2\right)$$

This is shown in the following sketch,



A weighting factor is used to average coordinates of centroids of the two subpanels and derive coordinates of panel centroid. The weighting factor is merely the ratio of the area of subpanel 1 to the panel area (see subroutine AREAP). That is,

$$\bar{x} = \bar{x}_2 + R(\bar{x}_1 - \bar{x}_2)$$

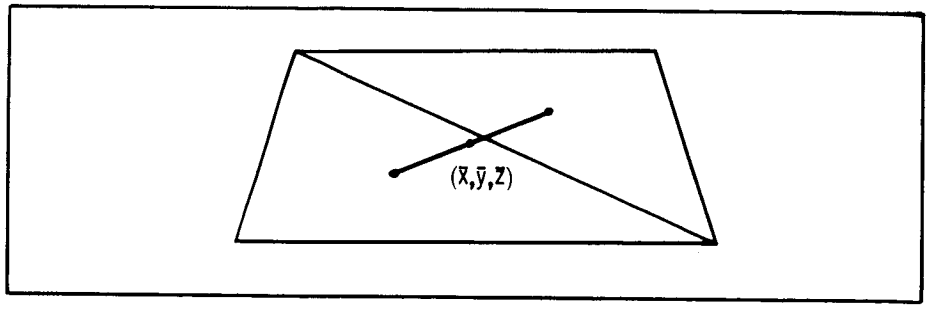
$$\bar{y} = \bar{y}_2 + R(\bar{y}_1 - \bar{y}_2)$$

$$\bar{z} = \bar{z}_2 + R(\bar{z}_1 - \bar{z}_2)$$

and,

$$R = \frac{S_1}{S}$$

This is shown below,



USAGE:

DIMENSION X(16, 16), Y(16, 16), Z(16, 16), R(15, 15), XC(15, 15), YC(15, 15), ZC(15, 15)

X, Y, Z = x, y, and z-coordinates of panel corner points.

R = Ratio of subpanel area to panel area.

XC, YC, ZC = x-, y-, and z-coordinates of panel centroid.

CALL CENTRD (X, Y, Z, R, XC, YC, ZC, NC, NR

Input: X, Y, Z, R (see above)

NC = Number of panel columns.

NR = Number of panel rows.

Output: XC, YC, ZC (see above)

SUBPROGRAM CALLED: None

ERROR RETURNS: None

STORAGE: $298_{10} = 452_8$

SUBJECT: FORTRAN IV Subroutine CHECK

PURPOSE: To check if a sequence $\{x_i\}$ of numbers is monotonic increasing (or decreasing).

METHOD: The given sequence $\{x_i\}$ can be one of four types:

1) Monotonic increasing; i.e., $x_i \leq x_{i+1}$
2) Strictly monotonic increasing; i.e., $x_i < x_{i+1}$
3) Monotonic decreasing; i.e., $x_i \geq x_{i+1}$
4) Strictly monotonic decreasing; i.e., $x_i > x_{i+1}$

The program calling sequence includes a code that indicates which type the sequence should be. The program consists of a "do-loop" involving an "if" statement that checks the sequence accordingly.

USAGE: CALL CHECK (X, N, K, L, N6)

Input:
- X = Sequence to be checked.
- N = Number of elements in sequence $\{x_i\}$.
- K = Code indicating sequence type that $\{x_i\}$ should be. If K = 1, $\{x_i\}$ should be monotonic increasing sequence. If K = 2, $\{x_i\}$ should be strictly monotonic increasing sequence. If K = -1, $\{x_i\}$ should be monotonic decreasing sequence. If K = -2, $\{x_i\}$ should be strictly monotonic decreasing sequence.
- N6 = Output tape unit.

Output:
- L = Error code. If L = 0, $\{x_i\}$ successfully checked. If L = 1, error occurred.

SUBPROGRAM CALLED: None

ERROR RETURNS: See above

RESTRICTIONS: The sequence $\{x_i\}$ must be a sequence of floating point numbers.

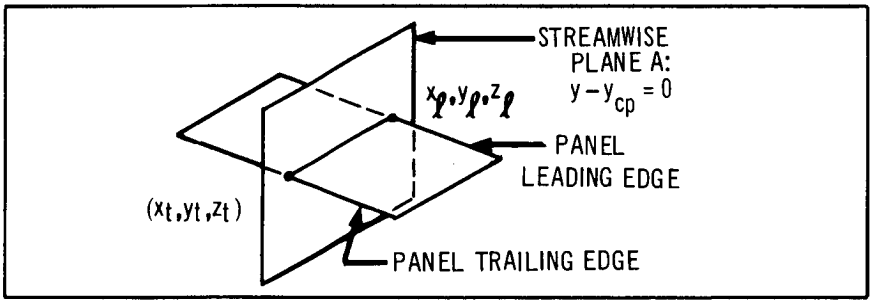
STORAGE: $237_{10} = 355_8$

SUBJECT: FORTRAN IV Subroutine CHORDW

PURPOSE: To calculate the length of a streamwise chord through the wing panel control point.

METHOD: The program calculates the length of the streamwise chord by use of a basic distance formula.

The chord is located by construction of a streamwise plane (A) normal to the plane of the panel and passing through the panel control point. This is shown in the following sketch:



The line of intersection of this plane and the original panel is the desired chord, and the chord length is calculated,

$$L = x_t - x_\ell$$

USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description)

COMMON /COM1/ (See subroutine PANEL)

Input: NPER1
NPLN1

COMMON /COM2/ (See subroutine WING)

Input: XCOR
YCOR
ZCOR
YCON

Output: CHORD

CALL CHORDW

SUBPROGRAM CALLED: POLXN

ERROR RETURNS: Error messages indicate if error occurred in use of subroutine POLXN to calculate chord intercepts.

STORAGE: $418_{10} = 642_8$

SUBJECT: FORTRAN IV Subroutine CLOK

PURPOSE: To find the time of day.

METHOD: This is a dummy subroutine which returns blanks. An appropriate change may be made at the user's installation to return the true time and date.

USAGE: CALL CLOK (TIME, DATE)

Output: TIME = Time of day (alphameric).

DATE = Date (alphameric).

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

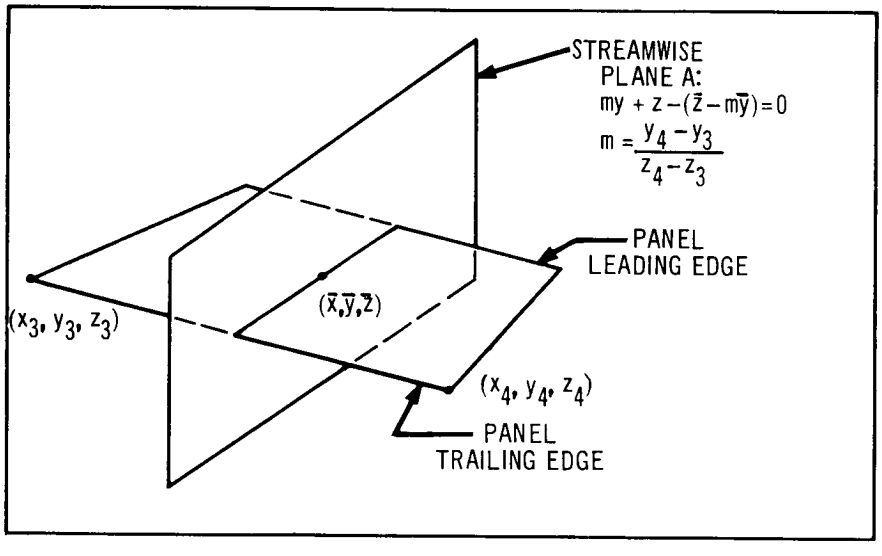
STORAGE: $23_{10} = 27_8$

SUBJECT: FORTRAN IV Subroutine CNTRLB

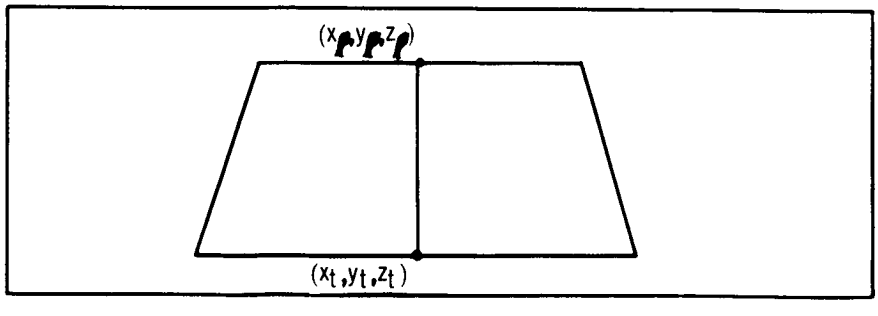
PURPOSE: To calculate the coordinates of a body panel control point and the length of a streamwise chord through the control point.

METHOD: The coordinates of a body panel control point are calculated by locating a streamwise chord through the panel centroid, determining the intercepts of this chord with the panel leading and trailing edges, and using a specified input constant to average these coordinates and locate the control point along the chord.

To find the chord that is required, the program constructs a streamwise plane (A) normal to the plane of the panel and through the panel centroid. For example,



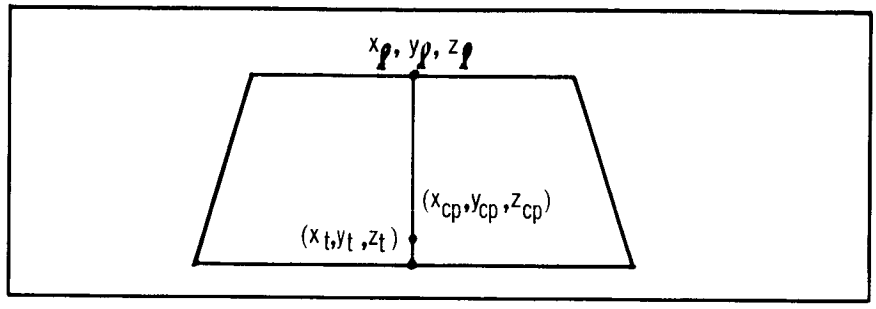
The intersection of this plane with the given panel defines the chord that is needed, and the chord length is calculated using a standard distance formula. For the above example,



and,

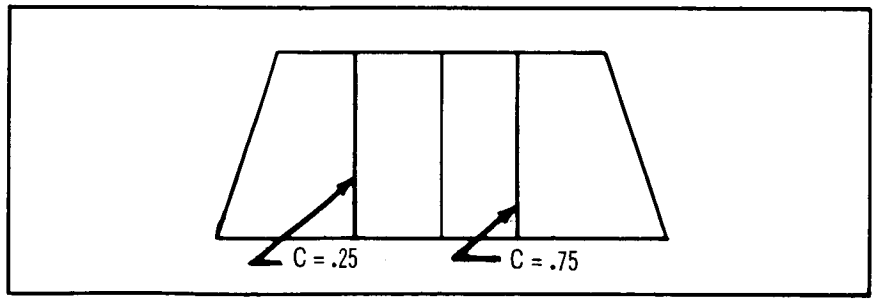
$$L = \left[(x_t - x_\ell)^2 + (y_t - y_\ell)^2 + (z_t - z_\ell)^2\right]^{1/2}$$

The control point coordinates are then calculated by using an input constant, c, and averaging the chord intercepts.



$$x_{cp} = x_\ell + c \cdot (x_t - x_\ell)$$
$$y_{cp} = y_\ell + c \cdot (y_t - y_\ell)$$
$$z_{cp} = z_\ell + c \cdot (z_t - z_\ell)$$

As noted above, the location of the panel control point along the streamwise chord is controlled by an input constant, c, (fraction). In addition and as an option, a second constant can be employed as input to change the spanwise location of the chord on which the control point is calculated. For a case in which the option is <u>not</u> used, the streamwise chord is constructed through the panel centroid; thus is located at approximately 50 percent in the spanwise direction. But for a case in which this option <u>is</u> used, this spanwise percent value can be changed to an arbitrary one, say 25 or 75 percent.



USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description).

CONTROL /COM1 / (See subroutine PANEL)

Input: KODEC
XPER
YPER
NPER1
NPLN1

Output: KODEB

COMMON /COM2/ (See subroutine BODY)

Input: XCOR
YCOR
ZCOR
YCEN
ZCEN

Output: XCON
YCON
ZCON
CHORD

CALL CNTRLB

SUBPROGRAMS CALLED: POLXN
SQRT (Built-in function)

ERROR RETURNS: Error message indicates if error has been encountered in use of subroutine POLXN to calculate chord intercepts.

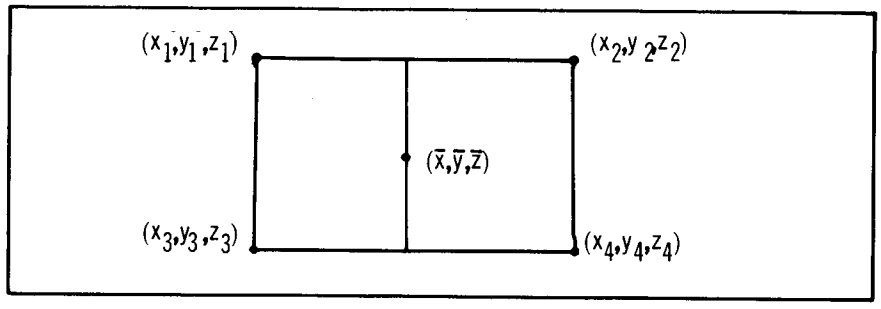
STORAGE: $\overset{>}{484}_{10} = 744_8$

SUBJECT: FORTRAN IV Subroutine CNTRLW

PURPOSE: To calculate the coordinates of a wing panel control point.

METHOD: The coordinates of a wing panel control point are calculated by defining a constant, c_1, that locates the control point in the spanwise direction, and a second constant, c_2, that locates it in the streamwise direction. The latter constant, c_2, is always given as input data, though the former constant, c_1, can be given as input or calculated by the program as a function of the centroid location.

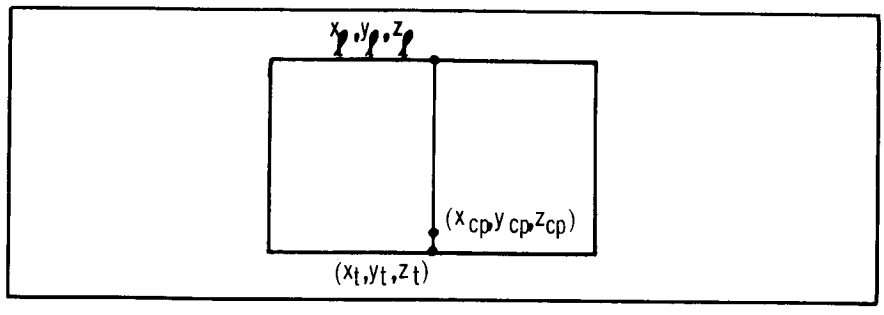
The two constants are used to average the panel corner point coordinates and derive the control point coordinates. As an example, a wing panel is defined,



The constant, c_2, is given as input and the constant, c_1, is either given as input or calculated by the program. For the latter case,

$$c_1 = \frac{\bar{y} - y_1}{y_2 - y_1}$$

These constants are then used to average the four corner point coordinates and obtain the control point coordinates,



$$x_{cp} = x_\ell + c_2 \cdot (x_t - x_\ell)$$
$$x_\ell = x_1 + c_1 \cdot (x_2 - x_1)$$
$$x_t = x_3 + c_1 \cdot (x_4 - x_3)$$
$$y_{cp} = y_\ell + c_2 \cdot (y_t - y_\ell)$$
$$y_\ell = y_1 + c_1 \cdot (y_2 - y_1)$$
$$y_t = y_3 + c_1 \cdot (y_4 - y_3)$$
$$z_{cp} = z_\ell + c_2 \cdot (z_t - z_\ell)$$
$$z_\ell = z_1 + c_1 \cdot (z_2 - z_1)$$
$$z_t = z_3 + c_1 \cdot (z_4 - z_3)$$

USAGE:

COMMON (See subroutine OPCAMI for unlabeled COMMON description)

COMMON /COM1/ (See subroutine PANEL)

Input: KODEC
XPER
YPER
KSTART
KEND
NPER1
NPLN1

COMMON /COM2/ (See subroutine WING)

Input: XCOR
YCOR
ZCOR
YCEN

Output: XCON
YCON
ZCON

CALL CNTRLW

SUBPROGRAM CALLED:

AMAX (Built-in function)

ERROR RETURNS:

Error message indicates if error occurred in calculation of panel control point.

STORAGE:

$735_{10} = 1337_8$

SUBJECT: FORTRAN IV Subroutine CP

PURPOSE: To compute the pressure coefficients by either the linear or nonlinear equation, given the velocity components in the x, y, and z directions at each of the panels.

METHOD: For nonlinear pressure coefficient

$$C_p = -2u + \beta^2 u^2 - v^2 - w^2$$

For linear pressure coefficient

$$C_p = -2u$$

where

u = Velocity component in the x-direction.

v = Velocity component in the y-direction.

w = Velocity component in the z-direction.

$\beta = M^2 - 1$

USAGE: CALL CP(NP, BETA2, CPCALC, U, V, W, CPP)

DIMENSION U(NP), V(NP), W(NP), CPP(NP)

Input: NP = Number of panels.

BETA2 = $M^2 - 1$.

CPCALC = Calculation option.

= 0., calculation of pressure coefficients to use linear equation.

$\neq$ 0., calculation of pressure coefficients to use nonlinear equation.

U = Array of velocity components in the x-direction.

V = Array of velocity components in the y-direction.

W = Array of velocity components in the z-direction.

Output: CPP = Array of pressure coefficients.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $129_{10} = 201_8$

SUBJECT: FORTRAN IV Subroutine CRNRB

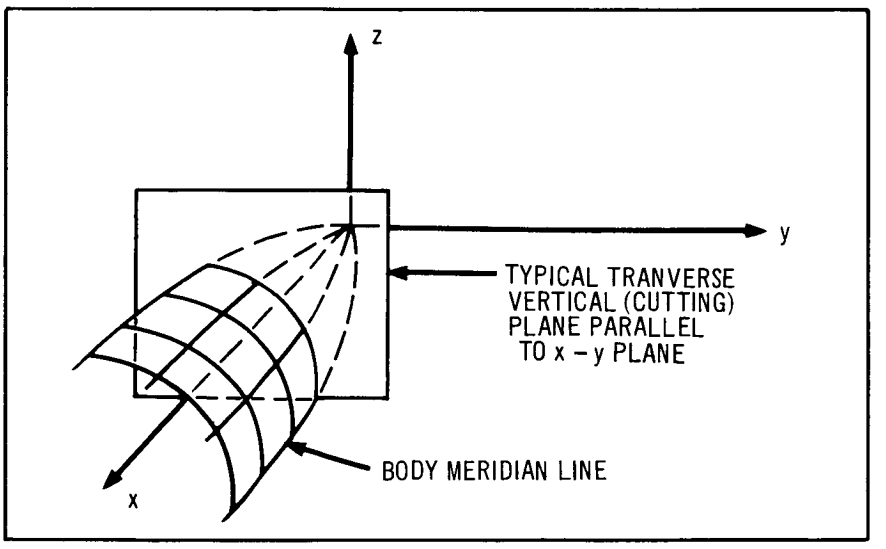
PURPOSE: To calculate body panel (and secondary panel) corner point coordinates.

METHOD: In general, a body panel is defined by four corner points. A program restriction requires either that the panel have streamwise inboard and outboard edges, or that it be subdivided so that its secondary panels have streamwise edges. To satisfy this condition, additional points are located on the panel leading and/or trailing edge.

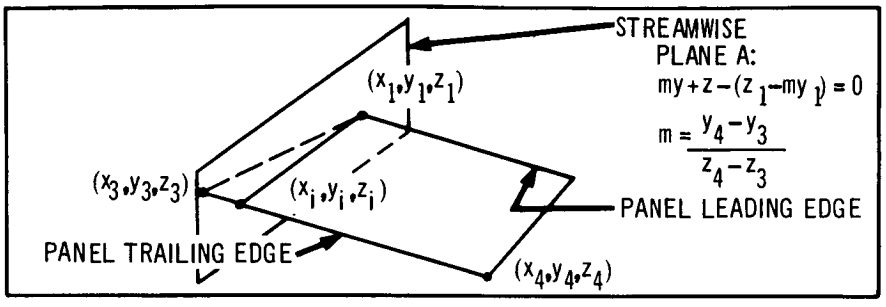
The program calculates the four panel corner points by intersecting with the body meridian lines a series of parallel transverse vertical (cutting) planes. The equations for these planes have the general form,

$$x - c_i = 0,$$

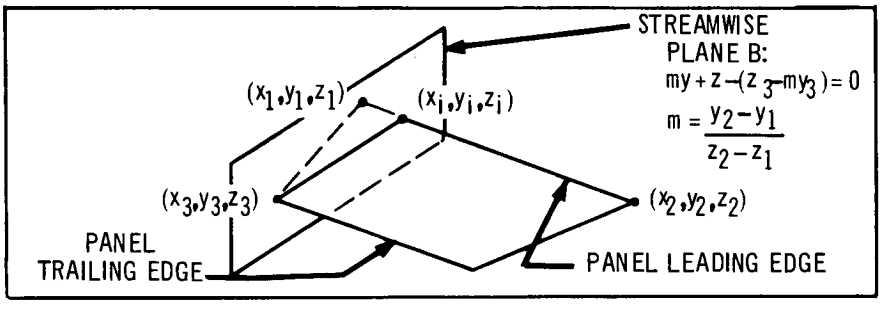
for which c_i is the ith intercept as given on input cards. This is shown in the following sketch:



The program calculates the additional panel points by constructing planes normal to the plane of the panel and through a panel corner point, and intersecting these planes with the panel leading and trailing edges. At most, two such constructions are necessary for each of the two required points. To obtain the first point, the program constructs a streamwise plane (A) normal to the panel trailing edge and through the inboard leading edge corner point and attempts to intersect this plane with the panel trailing edge:

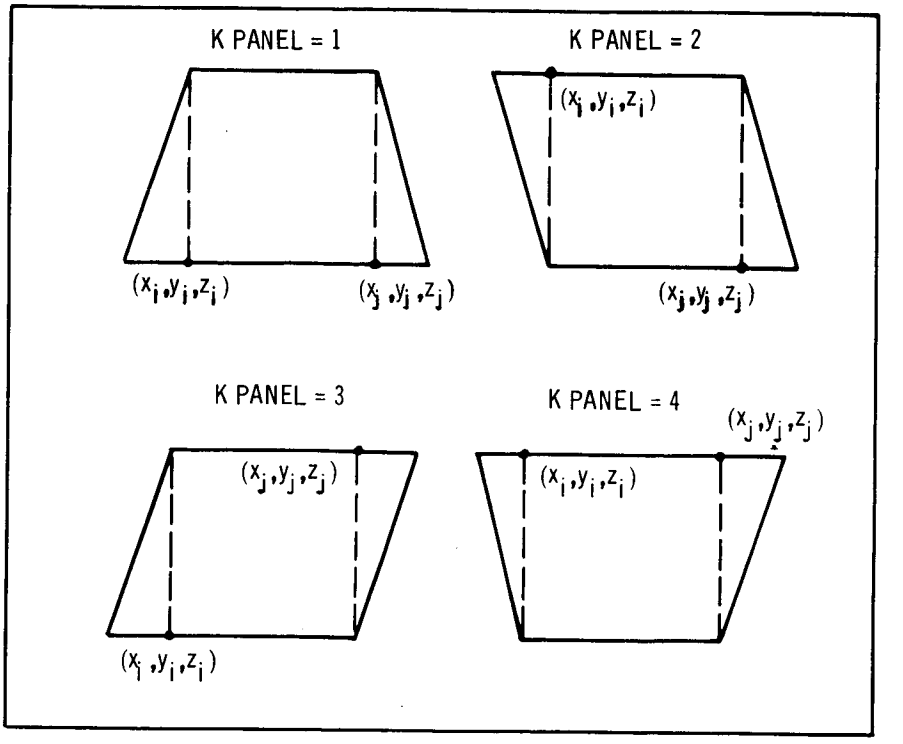


If an intersection with the trailing edge is obtained, this point is defined to be the first of the two required points. Should no intersection be calculated, the program then constructs a streamwise plane normal to the panel leading edge and through the inboard trailing edge corner point and attempts to intersect this plane with the panel leading edge:



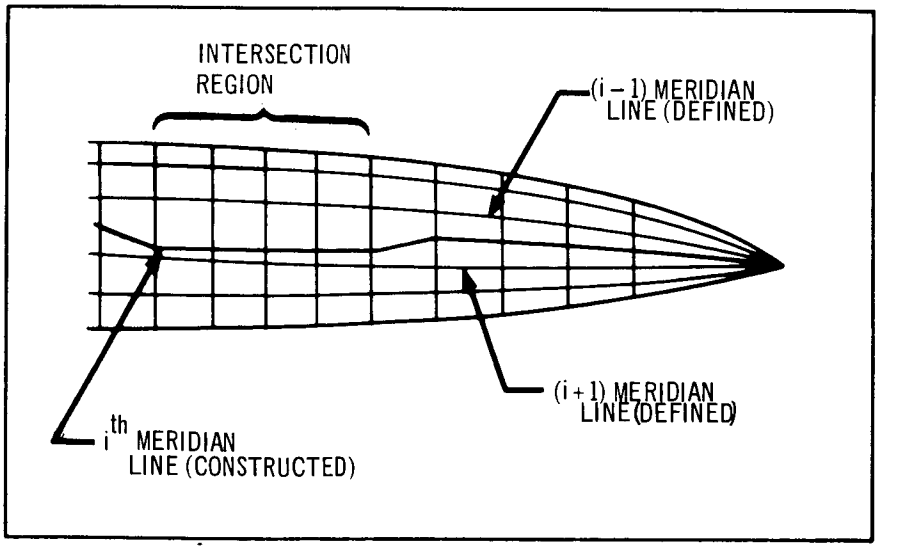
If the previous construction has failed, then this construction yields an intersection, and the point is defined to be the first of the two required points. The calculations used to obtain the second of the two points are identical to those given above except that the planes that are constructed pass through outboard rather than inboard corner points.

As both, either, or neither of these two points can occur on the panel leading (or trailing) edge, four panel configurations are possible, and a code is assigned for each panel. These codes are the following:



These codes are used in later subprograms.

In certain cases the body-wing intersection line does not coincide with a body meridian line, and an additional meridian line is constructed. In the intersection region proper, points on this additional line are defined to be the intersection points; elsewhere, fore and aft of the intersection region, points are obtained by averaging the corresponding points on the two adjacent body meridian lines. This is shown in the following sketch:



USAGE:

COMMON (See subroutine OPCAMI for unlabeled COMMON description).

Input: NWRITE

COMMON /COM1/ (See subroutine PANEL)

Input: KOPTB
KINT
XI
YI
ZI
NPER
NPER1
NPLANE
NPLN1

Output: KSTART
KEND

COMMON /COM2/ (See subroutine BODY)

Input: NPLNB
NPLNW
JLEAD
JTRAIL
NPTS
X
Y
Z
XCEPT
XCEPTW
YCEPTW
ZCEPTW

Output: IMID
KPANEL
XCOR
YCOR
ZCOR
XINT
YINT
ZINT

CALL CRNRB

SUBPROGRAM CALLED:	POLXN
ERROR RETURNS:	Error messages indicate if panel or secondary panel corner points not correctly calculated.
STORAGE:	$1282_{10} = 2402_8$

SUBJECT: FORTRAN IV Subroutine CRNRW

PURPOSE: To calculate wing panel (and secondary panel) corner point coordinates.

METHOD: In general, a wing panel is defined by four corner points. To satisfy a program restriction that either a panel have streamwise inboard and outboard edges or that it be subdivided so that its secondary panels have streamwise edges, an additional point is determined in the following two cases:

1) For panels on the inboard column if the body and wing intersect in a nonstreamwise line;

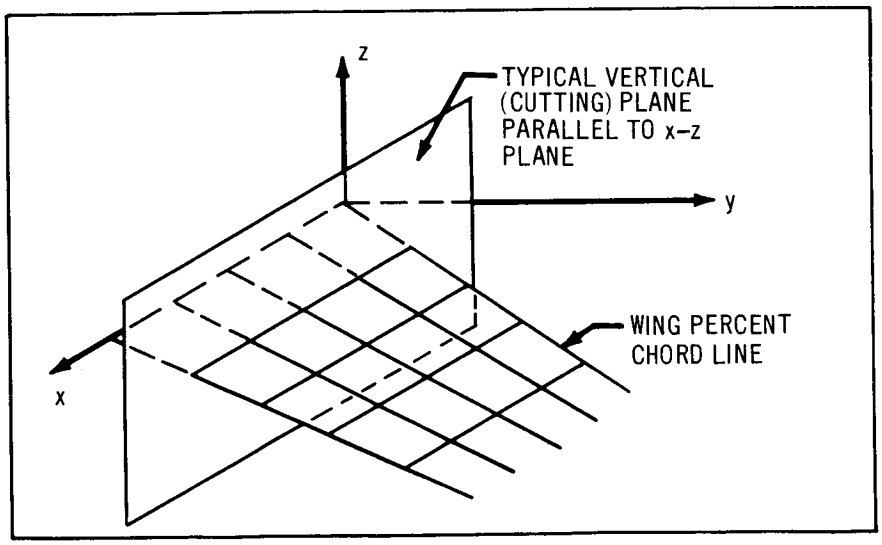
2) For panels on the outboard column if the outboard wing edge is a nonstreamwise edge.

All other panels have streamwise edges and are not subpaneled.

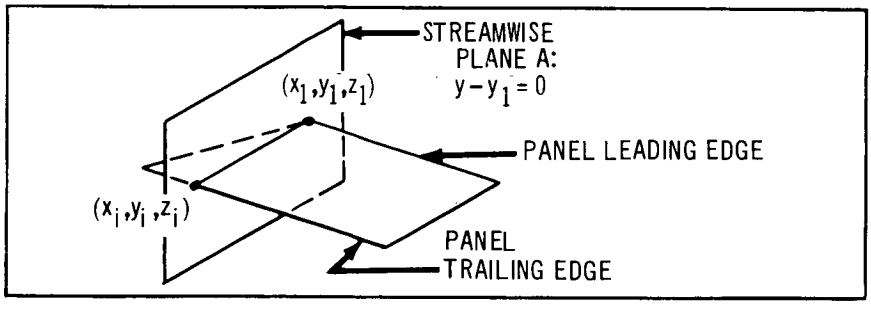
The program calculates the four panel corner points by intersecting with the wing percent chord lines a series of vertical (cutting) planes. The equations for these planes have the general form,

$$y - c_i = 0,$$

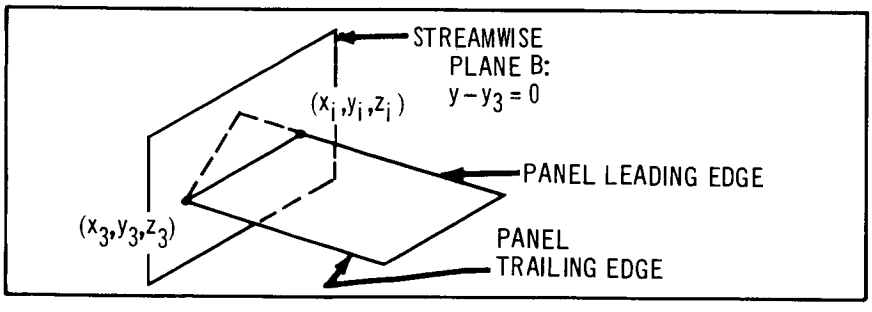
for which c_i is the ith intercept as given on input cards. This is shown in the following sketch.



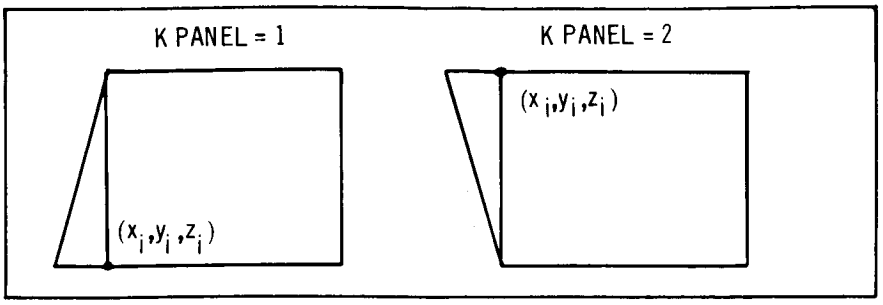
The program calculates the additional panel point, in those cases indicated above, by constructing planes normal to the plane of the panel and through a panel corner point, and by intersecting these planes with the panel leading or trailing edges. Two such constructions, at most, are required to obtain this point. The program first constructs a plane (A) normal to the panel and through a leading edge corner point and attempts to intersect this plane with the panel trailing edge.



If an intersection with the trailing edge is found, it is then defined to be that additional panel point. If no intersection is calculated, the program constructs a plane (B) normal to the panel and through a trailing edge corner point and attempts to intersect this plane with the leading edge.



As this additional point can be on either the leading or trailing edge a code is determined and assigned for any such panel.



USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description).

Input: NWRITE

COMMON /COM1 / (See subroutine PANEL)

Input: KOPTW
KINT
XI
YI
ZI
NPER
NPER1
NPLANE
NPLN1

Output: KSTART
KEND

COMMON /COM2 / (See subroutine WING)

Input: IJ
NPTS
X
Y
Z
KPNT
VALUE
YCEPT
SLOPE

Output: KPANEL
XCOR
YCOR
ZCOR
XINT
YINT
ZINT

CALL CRNRW

SUBPROGRAM CALLED:	POLXN
ERROR RETURNS:	Error messages indicate if panel (or secondary panel) corner points not calculated.
STORAGE:	$926_{10} = 1636_8$

SUBJECT: FORTRAN IV Subroutine CVEL

PURPOSE: To compute the velocity components due to a given pressure difference, given the velocity components resulting from a unit pressure difference across panels.

METHOD: The velocity components resulting from a unit pressure difference across the influencing panels are computed and stored on tape previous to calling of this subroutine. Then the velocity components resulting from a given pressure difference across the influencing panels are found by

$$u_i = \sum_{i=1}^{M} \sum_{j=1}^{N} U_{ij} \cdot p_j$$

$$v_i = \sum_{i=1}^{M} \sum_{j=1}^{N} V_{ij} \cdot p_j$$

$$w_i = \sum_{i=1}^{M} \sum_{j=1}^{N} W_{ij} \cdot p_j$$

where U_{ij}, V_{ij}, and W_{ij} are the velocity components in the x, y, and z directions respectively at panel i induced by panel j, and p_j is the given pressure differences across the influencing panel.

USAGE: CALL CVEL(M, N, NT, A, B, C, CL, U, V, W)

DIMENSION A(M), B(M), C(M), U(N), V(N), W(N)

Input:

M = Number of influenced panels.

N = Number of influencing panels.

NT = Logical tape number on which the velocity components due to unit pressure differences are stored.

A = Dummy array used by the subroutine.

B = Dummy array used by the subroutine.

C = Dummy array used by the subroutine.

CL = Array of given pressure differences.

Output: U = Array of velocity components in the x-direction.

V = Array of velocity components in the y-direction.

W = Array of velocity components in the z-direction.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $129_{10} = 201_8$

SUBJECT: FORTRAN IV Subroutine DATE

(See subroutine FDATE)

SUBJECT: FORTRAN IV Subroutine DCPD

PURPOSE: To compute the pressure difference across the wing panels.

METHOD: The pressure difference across a wing panel i is given by:

$$p_{W_i} = \sum_{j=1}^{NM} A_{R_{ij}}^{-1} \cdot n_{W_i}$$

where A_R^{-1} = The inverse of the reduced aerodynamic influence coefficients matrix.

n_W = The velocity component of the wing, normal to the x-axis.

USAGE: CALL DCPD (NM, NTAPEX, A, ALPHAM, CLM)

DIMENSION A(NM), ALPHAM(NM), CLM(NM)

Input:

NM = Number of wing panels.

NTAPEX = Logical tape number, of which A_R^{-1} is stored.

A = Dummy array used by the subroutine.

ALPHAM = Array velocity components on wing panels normal to the x-axis.

Output:

CLM = Array of the pressure difference across wing panels.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE $90_{10} = 132_8$

SUBJECT: FORTRAN IV Subroutine DCPI

PURPOSE: In the presence of a wing, to compute the pressure difference across a body panel.

METHOD: The pressure difference is given by:

$$p_{B_i} = \sum_{j=1}^{NB} A_{BB_{ij}}^{-1} \cdot n_{B_j} + \sum_{j=1}^{NW} E_{ij} \cdot p_{W_i}$$

where A_{BB}^{-1} = The inverse of a partition of the aerodynamic influence coefficients resulting from the influence of the body on the body.

n_B = The velocity component normal to the x-axis, on the body.

E = The matrix formed by the product,

$$A_{BB}^{-1} \cdot A_{BW}$$

where A_{BW} = The partition of the aerodynamic influence coefficients matrix resulting from the influence of the wing on the body.

p_W = The pressure difference across the wing panels.

USAGE: CALL DCPI (NB, NW, NTAPEX, A, ALPHAB, CLW, CLB)

DIMENSION A(NW), ALPHAB(NB), CLW(NW), CLB(NB)

Input:

NB	=	Number of body panels.
NW	=	Number of wing panels.
NTAPEX	=	Logical tape number on which A_{BB}^{-1} and E are stored.
A	=	Dummy array used by the subroutine.
ALPHAB	=	Array of velocity components on the body, normal to the x-axis.
CLW	=	Array of the pressure difference across wing panels.

Output:

CLB	=	Array of the pressure difference across the body panels.

SUBPROGRAM CALLED:	None
ERROR RETURNS:	None
RESTRICTIONS	None
STORAGE:	$143_{10} = 217_8$

SUBJECT: FORTRAN IV Subroutine DEFEN1

PURPOSE: To write records on the Geometry Definition binary tape.

METHOD: The format of the binary tape to be written is shown in Appendix C. Each time that DEFEN1 is called it writes two records. Record A consists of ten fixed-point words. Record B contains either nine words or K - J + 1 words, where K and J are given. A ten-word array, IREC, is given. If IREC(3) $\neq$ 0, then record B consists of nine words, which are zeros, and record A has the following format:

word 1	IREC(1)
word 2	0
word 3	IREC(3)
words 4-10	0

However, if IREC(3) = 0, then IREC is written as record A and the specified part of a given array is written as record B.

USAGE:

```
DIMENSION   B(1), IREC(10)
CALL DEFEN1 (B, J, K, LTAPE, IREC)
```

Input:

B	=	B(J) through B(K) is to be written on LTAPE.
J	=	Subscript.
K	=	Subscript.
LTAPE	=	Tape number.
IREC	=	Array to be written on LTAPE (see METHOD, above).

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: LTAPE is assumed to be correctly positioned.

STORAGE: $128_{10} = 200_8$

SUBJECT: FORTRAN IV Subroutine DEFEND

PURPOSE: To prepare records for the Geometry Definition tape and to monitor tape writing.

METHOD: This subroutine is used by subroutine GEOMD to transfer data from core to tape. If wing-body intersections are not requested, GEOMD calls DEFEND at the end of the case to write records 1-18 (see Appendix C). But if intersections are requested, GEOMD first calls DEFEND to write records 1-10, then finds intersections, and finally calls DEFEND again to write records 11-18.

USAGE:

```
DIMENSION  B(1), LK(3, 9)

LOGICAL  LGDEF(3, 6), LGTAPE

CALL DEFEND (B, LK, LO, LTAPE, LGDEF, LGTAPE,
INTWB)
```

Input: B = Storage buffer for variable-length arrays.

LK = Array that describes contents of B. Consider LK(i, j). The index j refers to a data array as follows:

j	Data array
1	Upper wing percent lines
2	Lower wing percent lines
3	Upper wing percent lines (part outside body)
4	Lower wing percent lines (part outside body)
5	Body meridian lines
6	Body stations
7	Wing planform (leading and trailing edge points)
8	Upper wing-body intersections
9	Lower wing-body intersections

A data array is stored starting in B(LK(1, j)). The number of elements

in the data array is LK(3, j). The number of percent lines (j = 1, 2, 3, 4) or meridian lines (j = 5) is LK(2, j). The number of airfoils is LK(2, 7). LK(2, j) is not used for j = 6, 8, 9.

	LO	= Output tape number.
	LTAPE	= Binary tape number.
	LGDEF	= Logical array of problem status (see subroutine GEOMD).
	LGTAPE	= .TRUE. if records are to be written on LTAPE, .FALSE. if not.
	INTWB	= If INTWB < 0, LTAPE is to be rewound and records 1-10 written. If INTWB > 0, records 11-18 and an EOF are to be written. If INTWB = 0, records 1-18 and an EOF are to be written.
Output:	LGDEF	= If LGDEF(1, i) and LGDEF(2, i) are both true, LGDEF(3, i) is set to .TRUE. to indicate that the corresponding data array has been written on LTAPE.

SUBPROGRAM CALLED: DEFEN1

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $373_{10} = 565_8$

SUBJECT: FORTRAN Function DISPTA

PURPOSE: To find the distance between two points in N-space.

METHOD: Standard

USAGE: D = DISPTA (P, Q, N)

DIMENSION P(N), Q(N)

Input: P = Array of coordinates of one point.

Q = Array of coordinates of other point.

N = Dimension of space.

Output: D = Distance between points P and Q.

SUBPROGRAM CALLED: SQRT (Built-in function)

ERROR RETURNS: None

RESTRICTIONS: $N > 0$

STORAGE: $50_{10} = 62_8$

SUBJECT: FORTRAN IV Subroutine DMAXL

PURPOSE: Given a set of points in 3-space, to find which point is farthest from the line through the end-points.

METHOD: Let P_1 and P_N be the first and last points, and let $\bar{u}$ be a unit vector from $\bar{P}_1$ to $\bar{P}_N$. For each intermediate point P_i, find the vector $\bar{v} = \bar{P}_i - \bar{P}_1$. Then the distance from P_i to the line $P_1 P_N$ is the magnitude of the cross-product of $\bar{u}$ and $\bar{v}$ The point P_j farthest from the line is selected. If there is more than one such point, the first is selected.

USAGE:

```
DIMENSION   P(3, N), U(3)

CALL DMAXL (P, N, DIST, J, U, NU)
```

Input: P = Array of points.
N = Number of points in P.

Output: DIST = Distance from point P_j to the line $P_1 P_N$.
J = Subscript of point P_j (that is, the desired point is P(i, J), i = 1, 3).
U = Unit vector directed from P_1 to P_N.
NU = Error indicator, which is zero for success.

SUBPROGRAMS CALLED: UVECN
VCROS

ERROR RETURNS: NU = 1 if N = 2,
= -1 if N < 2,
= -2 if P_1 and P_N are coincident.

RESTRICTIONS: None

STORAGE: $139_{10} = 213_8$

SUBJECT: FORTRAN IV Subroutine ELLIPR

PURPOSE: Given the horizontal and vertical semi-axes of a rectangular ellipse and the angle of a radius vector to find the length of the vector and the corresponding point on the ellipse.

METHOD: The sine and cosine of the angle (measured in a clockwise direction from the + y axis) are given. The required quantities are computed in a standard manner.

USAGE: CALL ELLIPR (A, B, SIN, COS, RHO, X, Y, NU)

Input:
- A = Horizontal semi-axis.
- B = Vertical semi-axis
- SIN = Sine of angle.
- COS = Cosine of angle.

Output:
- RHO = Length of radius vector.
- X = x-coordinate.
- Y = y-coordinate.
- NU = Error code, which is zero if successful.

SUBPROGRAM CALLED: SQRT (Built-in function)

ERROR RETURNS:
- NU = 1 if either of the semi-axes is zero.
- NU = 2 if the radius vector has zero length. (This indicates that SIN = COS = 0.)

RESTRICTIONS: None

STORAGE: $83_{10} = 123_8$

SUBJECT: FORTRAN IV Subroutine ENRYCH

PURPOSE: Given a set of points that lie on a smooth curve, to interpolate additional points so spaced that between each pair of points the distance from the chord to the assumed interpolation curve is less than a given tolerance.

METHOD: For each set of four consecutive points, subroutine OPTIM3 is called to interpolate additional points. Because OPTIM3 measures error between a chord and arc in a y direction, x and y are reversed for each set of four consecutive points if the range of y is greater than the range of x. Also, signs are reversed if necessary to make the equivalent x_i strictly increasing before calling OPTIM3. The inverse operations are performed after OPTIM3 does the interpolation.

USAGE:

```
DIMENSION  A(NI), B(NI), C(NO), NU(3)
CALL ENRYCH (A, B, NI, EPS, C, NO, NU)
```

Input:

- A = Array of independent variable coordinates.
- B = Array of dependent variable coordinates.
- NI = Number of pairs, A_i, B_i.
- EPS = Arc-chord tolerance.
- NO = Length of array C.
- NU(2) = Logical output tape number on which to write error comments, if any.
- NU(3) = Error message limiter and counter. If an error occurs for which a message is normally written, NU(3) is first reduced by one. The message is then written only if both NU(2) and NU(3) are greater than zero.

Output:

- C = Array of A_i, B_i pairs interspersed with interpolated pairs. If NI < 4 or EPS $\leq$ 0, no interpolation takes place; then C contains A_i, B_i, NO = 2*NI, and NU = 0.
- NO = Number of elements in the C array.
- NU = 0 — success.
  = 2 — at least one set of four consecutive points (A_i, B_i) is not strictly monotonic in either x or y.

= 3 — more than 199 interpolated points are required between two consecutive points (A_i, B_i) in order to satisfy the given value of EPS. This may be due to (1) a very small value of EPS, or (2) a "wild" interpolating cubic between two points (points too "rough" or too far apart in that region).

= 4 — C array is too small.

= 5 — the input value of NO is less than 2*NI.

SUBPROGRAM CALLED: OPTIM3

ERROR RETURNS: If an error is encountered in any interpolating interval (NU = 2 or 3), an error message is written if NU(2) and NU(3) > 0, no points are interpolated in that interval, and the routine continues. The value of NU returned is that of the last error encountered (if any).

RESTRICTIONS: For every set of four consecutive points, either the independent variable coordinates or the dependent variable coordinates must be strictly monotonic. Since OPTIM3 uses cubic or quadratic interpolation, some input arrays may cause rather extreme swings in the output array. This may usually be corrected by supplying more input points in the swing region. In general, the output array of points will lie on a curve that maintains first derivative continuity. However, this may not be true if an x,y reversal (see METHOD, above) is required for some but not all sets of four points; appropriate scaling of one variable before entry (preferably by a power of 2) will ensure that either no set or all sets of four consecutive points will be reversed.

STORAGE: $975_{10} = 1717_8$

SUBJECT: FORTRAN IV Subroutine EVAL

PURPOSE: To compute the matrix of the aerodynamic influence coefficients, the velocity components due to a surface distribution of vorticity, and the velocity components due to a surface distribution of sources in the reference plane of the wing.

METHOD: The equations in this subroutine are given in detail in the Appendix B of Part I.

USAGE:

```
CALL EVAL (NI,XCPT, THKW)

COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,
NTAPED,NTAPEE,NTAPEF,NTAPEI,NTAPEO,NBODY,
NWING,XMACH,SYM,KACE

/BLOCK/ X(200,3,4),Y(299,3,4,),Z(200,3,4), XBAR(200),
YBAR(200), ZBAR(200),XC(200),YC(200),ZC(200),THETA
(200),ALPHAS(200),AREA(200),CHORD(200),NPART
(200),NPANEL,ISYM,A(200),U(200),V(200),VPM(200),
VV(200),W(200),WPM(200),WW(200),VPMM(200),WPMM
(200),ALPHAC(100),ALPHAT(100)

DIMENSION BBETAM(4),R(4),Q(4),CPM(4),RPM(4),
CPMM(4),RPMM(4),XCPT(200)
```

Input:

NI	=	Index to indicate the start of the influencing panels.
XCPT	=	Array of panel control point x-coordinates.
THKW	=	0., indicates the aerodynamic influence coefficients and velocity components due to a vortex distribution are to be computed.
	$\neq$	0., indicates the velocity components due to wing sources are to be computed.
NPANEL	=	Total number of panels.
ISYM	=	1, for unsymmetric configurations.
	=	2, for symmetric configurations.

NTAPEA = Logical number of tape on which the aerodynamic influence coefficients or normal velocity components on the body due to wing sources are written.

NTAPEB = Logical number of tape on which the velocity components are written.

XMACH = Mach number.

X = Array of x-coordinates of the panel corner points.

Y = Array of y-coordinates, of the panel corner points.

Z = Array of z-coordinates, of the panel corner points.

THETA = Array of θ angles (angle between the plane of the panel and a plane parallel to the x-y plane).

ALPHAS = Array of the panel slopes, $\frac{dz}{dx}$

Output: IF THKW = 0.

A = Matrix of the aerodynamic influence coefficients.

U = Velocity components in the x-direction due to vortex distribution.

V = Velocity components in the y-direction due to vortex distribution.

W = Velocity components in the z-direction due to vortex distribution.

IF THKW $\neq$ 0.

A = Normal velocity components on the body due to wing sources.

U = Velocity components in the x-direction due to wing sources

V = Velocity components in the y-direction due to wing sources.

W = Velocity components in the z-direction due to wing sources.

SUBROUTINES CALLED: ACOS, ALOG, SIN, TAN, SQRT (Built-in functions)

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $1872_{10} = 3520_8$

SUBJECT: FORTRAN IV Subroutine FASNCS

PURPOSE: To compute the arc sine and/or the arc cosine of a normalized floating point argument.

METHOD: The arc sine and arc cosine are calculated as follows:

1) for $0 \leq x \leq 1$, $\arccos x = (1 - x)^{1/2} a(x)$

2) for $-1 \leq x < 0$, $\arccos x = \pi - (1 - |x|)^{1/2} a(|x|)$

3) for $0 \leq x \leq 1$, $\arcsin x = \pi/2 - (1 - x)^{1/2} a(x)$

4) for $-1 \leq x < 0$, $\arcsin x = -\pi/2 + (1 - |x|)^{1/2} a(|x|)$

where a(x) is approximated by an eighth degree polynomial. For x from 1 to -1, the subroutine gives the arc sine in radians from $\pi/2$ to $-\pi/2$, and the arc cosine in radians from 0 to π.

USAGE: The subroutine FASNCS has two entry points, ASIN and ACOS; the function names ASIN (X) and ACOS (X) are used in FORTRAN arithmetic expressions.

SUBPROGRAM CALLED: SQRT (Built-in functions)

ERROR RETURNS: None

RESTRICTIONS: $|x| \leq 1$. If $|x| > 1$, then an error message is printed. The subroutine is accurate to 8 decimal places, with the following exceptions in the arc sine:

$\lvert x \rvert < 0.0015$	6 decimal figures
$0.0015 \leq \lvert x \rvert < 0.002$	7 decimal figures

STORAGE: $109_{10} = 155_8$

SUBJECT: FORTRAN IV Subroutine FDATE

PURPOSE: To make the date of the run available for printout.

METHOD: The FORTRAN monitor places the current date in core as a BCD word. The first two characters are the month, the second two characters are the day, and the last two characters are the year. This date is printed at the top of each page of a listing.

USAGE: CALL DATE(A, B)

The date, in BCD, is stored and arranged as follows:

Location A = XXXbYY

Location B = ,b19ZZ

where XXX represents a three-character abbreviation for the month, b represents a blank, and YY and ZZ represent the day and year, respectively.

DATE may be used as a function. If so used, the BCD word XXYYZZ is left in the accumulator upon exit from the subroutine,

C = DATE(A, B).

SUBPROGRAM CALLED: None

ERROR RETURNS: None

STORAGE: $41_{10} = 51_8$

SUBJECT: FORTRAN IV Subroutine FFSF

PURPOSE: To enable the FORTRAN program to move past end-of-file marks on FORTRAN written binary tapes, and to skip entire files.

METHOD: The first two words of each record on the designated logical tape unit are read using the IOCS read routine. If the second word is the Hollerith literal "TAPEND," the error code is set equal to the number of files remaining in the file count. Whenever end-of-file is encountered by the IOCS read subroutine, the file count is reduced by one. When the file count reaches zero, the error code is set to zero, and the subroutine returns to the main program. The tape is then positioned just beyond the end-of-file mark, ready to read the first record in the next file. A file count of zero or one will move past one end-of-file mark to the beginning of the next file on the tape. A file count of two will pass two end-of-file marks, thereby skipping the remainder (if any) of the current file and the next file in its entirety.

USAGE: CALL FSF (NFILE, LTAPE, LERROR)

Input:

NFILE = The file count described above; the number of files to space forward.

LTAPE = The logical tape unit to be used.

LERROR = Error code. If LERROR = 0, subroutine successful. If LERROR = n, n > 0, n indicates the number of unspaced files.

SUBPROGRAMS CALLED: IOCS
FVIO

ERROR RETURNS: None

RESTRICTIONS: If TAPEND indicator was written when the tape was written, FFSF will not read past the TAPEND signal. If not, it is possible to run off the end of the tape.

STORAGE: $52_{10} = 64_8$

SUBJECT: FORTRAN IV Subroutine FFSR

PURPOSE: To move a designated FORTRAN binary tape forward a specified number of logical records while indicating whether end-of-file was reached.

METHOD: Logical Records (FSR)

The first word in each physical record on the designated tape unit is examined using the IOCS read routine, and if the address is nonzero, the specified record count is reduced by one. If an end-of-file is encountered, the tape is back-spaced so that the end-of-file mark is, in effect, not passed, and the error code is set equal to the number of records remaining to be read. If no end-of-file is encountered, reading continues until the record count reaches zero; zero is stored in the error code and the subroutine returns to the calling program. No redundancy checking is done.

Physical Records (FSPR)

The method is the same as for a logical record except that the record count is reduced by each physical record spaced forward and the first word of each record is not examined.

USAGE: CALL FSR (N, LTAPE, LERROR)

Input: N = Number of FORTRAN logical records to forward space. Zero means no spacing.

LTAPE = Logical tape unit.

LERROR = Error code.

CALL FSPR (NP, LTAPE, LERROR)

Input: NP = Number of physical records to forward space. Zero means no spacing.

LTAPE = Logical tape unit.

LERROR = Error code.

SUBPROGRAMS CALLED: IOCS
FVIO

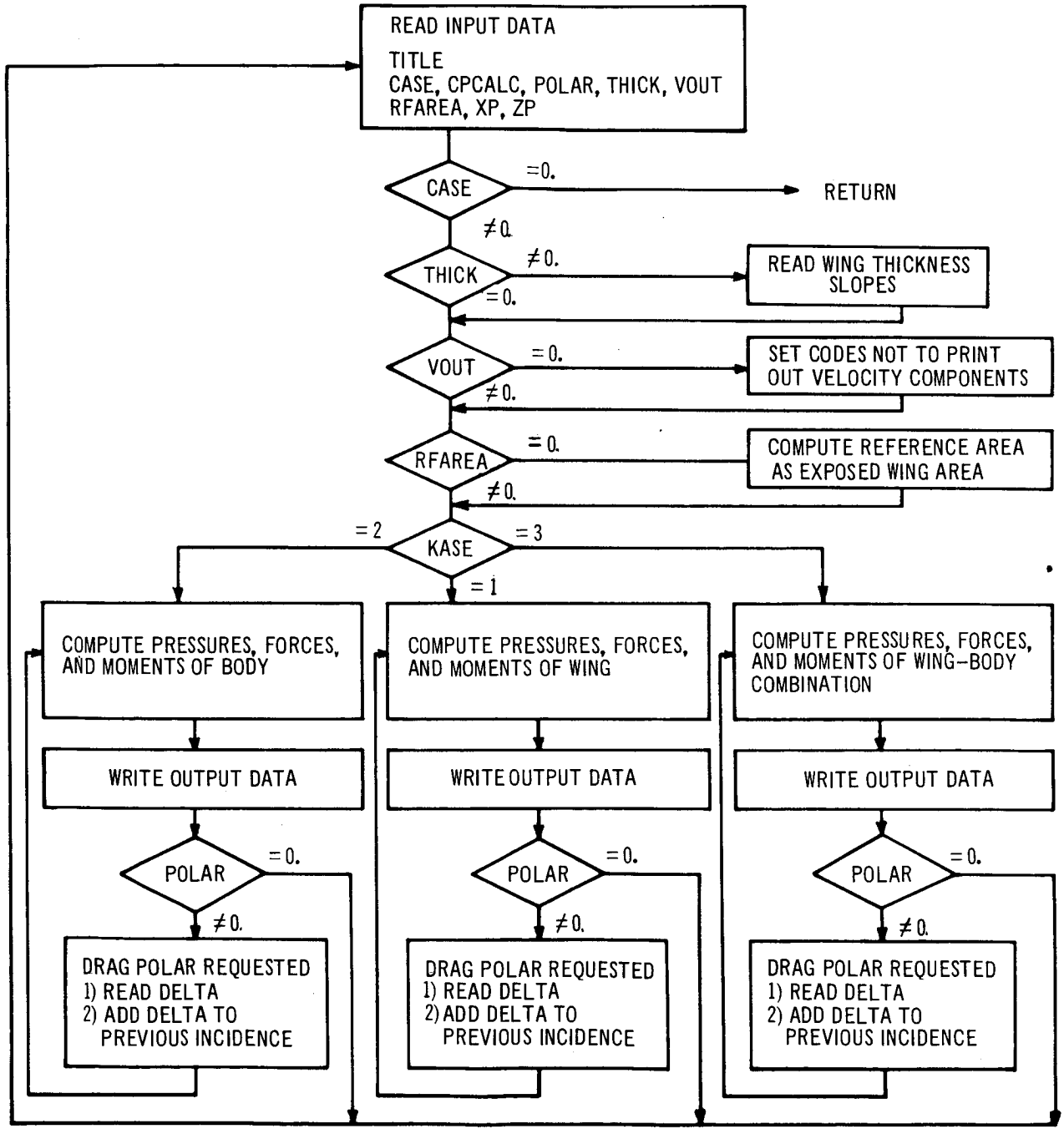
RESTRICTIONS: This routine will not proceed beyond an end-of-file mark on tape.

STORAGE: $53_{10} = 65_8$

SUBJECT: FORTRAN IV Subroutine FORCES

PURPOSE: This is the controlling subroutine for computing forces, moments, and pressures.

METHOD The subroutine computes the forces, moments, and pressures for the different cases of wing alone and body alone or for the wing-body combinations by calling a number of subroutines written to do a specific computation.



USAGE:

CALL FORCES

COMMON DATE (2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION AREA(200), ALPHA(100), ALPHAB(100), ALPHAL(100), ALPHAS(200), ALPHAU(100), ALPHAW (100), CL(200), CLBB(100), CLWW(100), CPL(100), CPU(100), CPB(100), THETA(200), TITLE(12), U(100), UWBT(100), UBWT(100), UWWT(100), V(100), VWBT (100), VBWT(100), VWWT(100), W(100), WWBT(100) WBWT(100), WWWT(100), XBAR(200), YBAR(200), ZBAR(200), ALPHAT(100), A(200), B(200), C(200), UBB (100), VBB(100), WBB(100), UWW(100), VWW(100), WWW(100), ALPHAX(100), UWB(100), VWB(100), WWB (100), UBW(100), VBW(100), WBW(100), ABX(100), AWX (100), WSCLL(20), WSCDL(20), WSCLU(20), WSCDU(20), WSCL(20), WSCD(20), CHORD(200), NROW(2), NFMT(8), XC(200), YC(200), ZC(200), XB(51), R(51), THETAB(10), ZDELTA(51), CPBB(51, 10)

Input: See Aerodynamic Card Set of Input Data Format (section 3.3).

Output: Body-Alone Case

BBCD = Coefficient of drag.

BBCL = Coefficient of lift.

BBCM = Coefficient of pitching moment

CPBB = Pressure coefficients.

Wing-Alone Case

WCD = Coefficient of drag.

WCL = Coefficient of lift.

WCM = Coefficient of pitching moment.

WSCD = Spanwise distribution of drag.

WSCL = Spanwise distribution of lift.

CPU = Upper surface wing pressure coefficients.

CPL = Lower surface wing pressure coefficients.

CL = Pressure difference
CL = CPL - CPU

ALPHAU = Upper surface wing panel slopes.

ALPHAL = Lower surface wing panel slopes.

Wing-body combination case

NOTE: In addition to the output for the wing alone and body alone, the incremental pressures, forces, and moments on the body due to the wing are given for this case:

BWCD = Coefficient of drag.

BWCL = Coefficient of lift.

BWCM = Coefficient of pitching moment.

CPB = Pressure coefficinets.

and also for the total wing-body combination:

WBCD = Coefficient of drag.

WBCL = Coefficient of lift.

WBCM = Coefficient of pitching moment.

SUBROUTINES CALLED:

INOUT
CAMBWB
CAMBW
BLDM
KARMOR
READ
WLDM
DCPD
DCPI
CVEL
INTPOL
OUTB
OUTW
OPTMWB
RITE
OPTMW
CP
COS, SQRT } (Built-in functions)

ERROR RETURNS: None

RESTRICTIONS: NBODY $\leq$ 100
NWING $\leq$ 100

STORAGE: $10978_{10} = 25342_8$

SUBJECT: FORTRAN Subroutine FROOTA

PURPOSE: To find the root of a single-valued function f(x).

METHOD: FROOTA is designed to be called repeatedly until a root is found within a given tolerance. On initial call it is given two points on the function; these two points bracket the root. FROOTA computes two values of x (estimates of the root location) and returns with a code indicating that it wishes to be called again. Calling program finds function values corresponding to the two root estimates and calls FROOTA a second time. FROOTA determines which points now bracket the root, computes new root estimates, and returns. Process continues until FROOTA signals that root has been found.

Root estimates are made as follows: (a) the first root is estimated by assuming the function to be linear between the bracketing points, and (b) the second root is estimated by bisecting the bracketing interval. A root is located when (1) exact root is found, (2) bracketing interval becomes smaller than given tolerance, or (3) root estimate made on the basis of linear interpolation is identical to x-value of bracketing point.

USAGE:

```
      DIMENSION R(10)
      R(9) = 0.
      R(10) = Tolerance
      R(1) =  }
      R(3) =  } Initial bracketing values of x
      DO α I = 1,50
      R(2) = Function value corresponding to R(1)
      R(4) = Function value corresponding to R(3)
      CALL FROOTA(R, J)
      IF (J) β, γ, α
α     CONTINUE
β     error exit
γ     success exit. Root is in R(1).
```

Input: (First call only. Assume root bracketed by x_a, x_b where $x_a > x_b$ or $x_a < x_b$.)

R(1) = x_a

R(2) = $f(x_a)$

R(3) = x_b

R(4) = $f(x_b)$

R(9) = 0.
R(10) = Tolerance (in x), which may be zero

Input: (2nd and succeeding calls)

R(2) = $f\left(R(1)\right)$

R(4) = $f\left(R(3)\right)$

Output: R(1) = Root estimate if J = 1
= Root if J = 0
R(3) = Root estimate if J = 1
R(5) = x_c
R(6) = $f(x_c)$
R(7) = x_d
R(8) = $f(x_d)$

where x_c and x_d are values of x which currently bracket the root ($x_c < x_d$).

R(9) = Number of times that FROOTA has been called.

J = 0 The root has been found.
= 1 The root has not yet been located.
= -2 R(9) < 0.
= -3 $f(x_a) * f(x_b) > 0$ on the first call, which means that the root is not bracketed.

SUBPROGRAM CALLED: None

RESTRICTIONS: Function values at limits of initial interval must not have the same sign. Function must be continuous over interval. It may have a number of roots but, of course, FROOTA will only find one. Last six elements of R-array must not be changed between successive calls for a particular case.

STORAGE: $186_{10} = 272_8$

SUBJECT: FORTRAN IV Subroutine FSF

(See Subroutine FFSF)

SUBJECT: FORTRAN IV Subroutine FSR

(See Subroutine FFSR)

SUBJECT: FORTRAN IV Subroutine FTAN

PURPOSE: To compute the tangent of a floating point radian argument.

METHOD: The tangent function is computed as follows:

1) if $0 \leq |x| \leq 10^{-4}$ radians, $\tan x = x$

2) if $10^{-4} < |x| \leq \pi/4$ radians,

$$\tan x = \frac{x}{1 - \frac{x^2}{3 - \frac{x^2}{5 - \frac{x^2}{7 - \frac{x^2}{9 - \frac{x^2}{11}}}}}}$$

3) if $\pi/4 < |x| < \pi/2$ radians, and

a) $x > 0$, then $\tan x = \dfrac{1}{\tan(\pi/2 - x)}$

b) $x < 0$, then $\tan x = \dfrac{1}{\tan(\pi/2 - |x|)}$

USAGE: The function name TAN(X) is used in an arithmetic expression where the argument X is a FORTRAN expression.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: X is restricted: $|X| < 2^{26} \cdot \pi$ and not within two bits of an odd multiple of $\pi/2$.

If $|X| > 2^{26} \cdot \pi$ an error message is given.

If the argument is within two bits of an odd multiple of $\pi/2$, an error message is given.

The accuracy is at least 8 significant digits for $0 \leq x \leq \pi$ radians, at least 7 significant digits for $\pi < x \leq 2\pi$ and at least 6 significant digits otherwise.

STORAGE: $124_{10} = 174_8$

SUBJECT: FORTRAN IV Subroutine GEOMD

PURPOSE: To control subprograms that deal with the first phase of wing-body geometry.

METHOD: A buffer is set up (see Appendix A) that is used by all lower-level subroutines for all variable-length storage arrays. Data cards are read by subroutine INTURP until a valid command (in columns 1-6) is found. After any command except DEFEND is executed, data cards are again read until another valid command is encountered; a DEFEND command returns control to the calling program. For a command BODY (or B0DY), subroutine BODY1 is called to read body definition data and compute points on body meridian lines. A command WING (or W1NG) results in a call of subroutine WINGA to read wing definition data and compute points on wing percent lines. A WBX command is used to find the intersections of previously defined wing percent lines with a body surface. The TDUMP command sets a code which will later result in a printout of the contents of the definition tape. A NOTAPE (or N0TAPE) command sets a code to suppress the writing of a definition tape (see Appendix C).

USAGE:

```
DIMENSION   DATE(2)
COMMON /LGEOMD/ LGDEF(3, 6)
LOGICAL   LGDEF
CALL GEOMD (DATE, LI, LO, LER, LTAPE)
```

Input: DATE = Date (alphameric).

LI = Input tape number.

LO = Output tape number.

LER = Output tape number for error comments.

LTAPE = Definition tape number.

Output: LDGEF = Logical array describing which definitions were requested and completed. LGDEF(i, j) is .TRUE. if the ith statement is true for the jth component.

<u>j^{th} component</u>

1. Upper wing surface percent lines.
2. Lower wing surface percent lines.

3. Body surface meridian lines.

4. Wing planform points.

5. Intersections of upper wing percent lines with body surface.

6. Intersections of lower wing percent lines with body surface.

i^{th} statement

1. This component requested.

2. This component successfully computed.

3. This component successfully computed and the results written on LTAPE.

SUBPROGRAMS CALLED:

CLEAR
INIBFR } (See Appendix A)
TDUMP
DEFEND
BODY1
INTURP
WINGA
WBXUL

ERROR RETURNS: The LGDEF array will reflect any errors detected. Lower-level subroutines write error messages on tape LER.

RESTRICTIONS: The suggested order for commands is BODY, WING, WBX, DEFEND. The minimum set of commands is BODY, DEFEND (or WING, DEFEND). The last command must always be DEFEND. NOTAPE must be placed before WBX (if present). BODY, ..., WING or WING, ..., BODY must precede WBX. A required set of data cards (see Input Data Format, 3.3) must follow each command except TDUMP, NOTAPE, and DEFEND.

STORAGE: $10289_{10} = 24061_8$

SUBJECT: FORTRAN IV Subroutine INOUT

PURPOSE: To print out all input aerodynamic data.

METHOD: The input data is written on the output tape for subsequent off-line printing.

USAGE: CALL INOUT (NTAPEO, KASE, CPCALC, POLAR, THICK, VOUT, RFAREA, XP, ZP, XMACH, SYM)

Input:

NTAPEO	=	Logical number of output tape.
KASE	=	Code that selects case option.
CPCALC	=	Code that selects pressure coefficient calculation option.
POLAR	=	Code that selects drag polar option.
THICK	=	Code that selects wing thickness option.
VOUT	=	Code that selects velocity components printout option.
RFAREA	=	Half wing reference area.
XP	=	x-coordinate of point about which the pitching moments are to be computed.
ZP	=	z-coordinate of point about which the pitching moments are to be computed.
XMACH	=	Mach number.
SYM	=	Configuration symmetry condition.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $472_{10} = 730_8$

SUBJECT: FORTRAN IV Subroutine INPUTB

PURPOSE: To read body definition and transformation data from scratch tapes as provided by previous program section and to read body paneling data from input cards.

METHOD: Program uses direct read tape statements to obtain body definition and transformation from scratch tapes (see Appendix C for tape format). Similarly program reads input tape directly for body paneling input card data.

USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description).

Input: ND1
ND4
NREAD
NWRITE

COMMON /COM1/ (See subroutine PANEL)

Output: KOPTB
KINT
XI
YI
ZI
NPER
NPER1
NPLANE
NPLN1

COMMON /COM2/ (See subroutine BODY)

Output: NPLNB
NPLNW
JLEAD
JTRAIL
NPTS
X
Y
Z
XCEPT
XCEPTB
XCEPTW
YCEPTW
ZCEPTW
CODEBW

CALL INPUTB

CALLED: FSR
CHECK

ERROR RETURNS: Error message indicates whether body cutting planes are incorrectly defined on input cards. Error messages also indicate whether size of subscripted array is exceeded by input card data or by data on scratch tapes.

STORAGE: $5972_{10} = 13524_8$

SUBJECT: FORTRAN IV Subroutine INPUTW

PURPOSE: To read wing definition and transformation data from scratch tape as provided by previous program section and to read wing paneling data from input cards. Program reads input cards that define wing cutting planes, and, if an option is used, program reads actual upper and lower airfoil surface coordinates from which wing camber and thickness are calculated.

METHOD: Program uses standard read tape instructions to obtain wing definition from scratch tapes (see Appendix C for tape format). Similarly program reads input tape for wing paneling card data.

USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description).

Input: ND4
NREAD
NWRITE

COMMON /COM1/ (See subroutine PANEL for description)

Output: KOPTW
KOPTF
NUMS
NPER
NPER1
NPLANE
NPLN1

COMMON /COM2/ (See subroutine WING for description)

Output: NPTS
X
Y
Z
KPNT
VALUE
YCEPT
SLOPE

CALL INPUTW

SUBPROGRAMS CALLED: FSR
CHECK

ERROR RETURNS: Error messages indicate if size of subscripted array is exceeded by data on input cards or on scratch tapes.

STORAGE: $5946_{10} = 13472_8$

SUBJECT: FORTRAN IV Subroutine INTAPE

PURPOSE: To input geometrical data to the aerodynamic links of the program.

METHOD: The geometrical data is read into the aerodynamic links of the program from binary scratch tapes generated in the geometry links. The format of the tape that transfers the panel geometry from the paneling links is described in Appendix C. The geometry of the isolated body is transferred from the transformation links. The format of this tape is given in Appendix C.

USAGE: CALL INTAPE (THKW)

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

/BLOCK/X(200, 3, 4), Y(200, 3, 4), XBAR(200) YBAR(200), ZBAR(200), XC(200), YC(200), ZC(200), THETA(200), ALPHAS(200), AREA(200) CHORD(200), NPART(200), NPANEL, ISYM, A(200), U(200), V(200), VPM(200), VV(200), W(200), WPM(200), WW(200), VPMM(200), WPMM(200), ALPHAC(100), ALPHAT(100)

DIMENSION JSAVE(2), NSAVE(2), NROW(2), XB(51), R(51) ZDELTA(51), ARRAY(3, 10), THETAB(10)

Input:

XMACH	=	Mach number.
SYM	=	Configuration symmetry condition.
NTAPEC	=	Logical tape number from which the panel geometry is read.
NTAPED	=	Logical tape number from which the geometry of the isolated body is read.
NBODY	=	Number of body panels.
NWING	=	Number of wing panels.

Output:

THKW	=	0., indicates velocity components due to wing sources are not to be computed.

= 1., indicates velocity components due to wing sources are to be computed.

KACE = 1, indicates wing only case.

= 2, indicates body only case.

= 3, indicates wing-body combination case

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: NWING $\leq$ 100

NBODY $\leq$ 100

STORAGE: $994_{10} = 1742_{8}$

SUBJECT: FORTRAN IV Subroutine INTPOL

PURPOSE: The drag of an individual panel may be obtained by multiplying the panel pressure by the product of the average downwash over the panel and the area. The program calculates panel pressure and downwash (n_i) at the control point . As the control point is normally located at 95 percent of the panel chord, the downwash at this point is often greater than the average downwash of the panel. Thus it is necessary to interpolate for the average downwash of the panel in calculating the drag.

METHOD: For the first panel of any chordwise column, the average downwash is obtained by a linear extrapolation of the downwash at the control points of the first two panels.

$$\bar{n}_1 = n_1 + \frac{(\bar{R} - R)\,(n_2 - n_1)}{1 + R\left(\frac{C_2}{C_1} - 1\right)}$$

where R = The location of the control point as a fraction of the local panel chord.

$\bar{R}$ = The location of the point as a fraction of the local panel chord at which the downwash used in drag calculations is to be found.

C = The local chord length of the panels.

For the remainder of the panels in any chordwise column of panels, the downwash is obtained by a linear interpolation of the downwash of the adjacent panels.

$$\bar{n}_i = n_i + \frac{C_i}{C_{i-1}} \frac{(\bar{R} - R)\,(n_i - n_{i-1})}{1 + R\left(\frac{C_i}{C_{i-1}} - 1\right)}$$

USAGE: CALL INTPOL (NM, NROW, RATIOX, RATIOD, CHORD, ALPHAM, ALPHAD)

DIMENSION CHORD(NM), ALPHAM(NM), ALPHAD(NM)

Input: NM = Total number of panels on wing or body.

NROW = Number of panels in a chordwise column.

RATIOX = Location of the control point as a fraction of the local panel chord.

RATIOD = Location of the point as a fraction of the local panel chord at which the downwash is to be found.

CHORD = Array of the local chord lengths of the panels.

ALPHAM = Array of panel downwash at the control point locations.

Output: ALPHAD = Array of interpolated panel downwash values.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $177_{10} = 261_8$

SUBJECT: FORTRAN IV Function INTURP

PURPOSE: To read and write a data card, then find which word in a given table is the same as the first word in the data card.

METHOD: Let LI be the input tape and LO the output tape. INTURP reads the next data card on LI with FORMAT (13A6, A2). If LO > 0, the contents of the card are written on LO with FORMAT (1H0,14A6). A fixed-point comparison is made between the first word on the card and each word in the table until a match is found or the end of the table is reached. The function value returned is the position in the table that matched the first word in the data card, or else zero if no match was found.

USAGE:

```
DIMENSION   T (N)
K   = INTURP (T, N, LI, LO)
```

T = Table.
N = Number of words in T.
LI = Input tape.
LO = Output tape.
K = Function value.

EXAMPLE: A data deck might have three types of data cards, each group with a header card having a code word in the first six columns, and perhaps comments in the remaining columns. Assume the code words are BODY, WING, and DEFINE. A portion of the program could be written as follows:

```
DIMENSION   T(3)
DATA T/4HBODY, 4HWING, 6HDEFINE/

10    K = INTURP (T, 3, 5, 6)
      IF (K) 10, 10, 20
20    GO TO (100, 200, 300), K
```

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $98_{10} = 142_8$

SUBJECT: FORTRAN IV Subroutine INVBB

PURPOSE: For the wing-body case, to preprocess the aerodynamic influence coefficients matrix.

$$\begin{bmatrix} A_{BB} & A_{BW} \\ A_{WB} & A_{WW} \end{bmatrix}$$

where

A_{BB} = The influence on the body due to the body

A_{WB} = The influence on the wing due to the body

A_{BW} = The influence on the body due to the wing

A_{WW} = The influence on the wing due to the wing

by inverting A_{BB} and partitioning A_{WB} in preparation for reduction of the matrix (subroutine REDUCE).

METHOD: The portion of the aerodynamic influence coefficients resulting from the influence of the body is read from logical tape NTAPEA by columns into core. A_{WB} is written immediately on logical tape NTAPED. A_{BB} is inverted in core by the single precision inversion subroutine SINVRT. The matrix inverse is then stored on NTAPED behind A_{WB}. Both matrices are written by columns as a logical record. Logical tape NTAPED is rewound and NTAPEA is left at the position of the last read instruction.

USAGE: CALL INVBB

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION ABB(102, 102), AWB(100)

Input:

NTAPEA	=	Logical number of scratch tape.
NTAPED	=	Logical number of scratch tape.
NTAPEO	=	Logical number of output tape.
NBODY	=	Number of body panels.
NWING	=	Number of wing panels.
A	=	Aerodynamic influence coefficients matrix on logical tape NTAPEA.

Output: AWB = Partition of the aerodynamic influence coefficients matrix resulting from the influence of the body on the wing written on logical tape NTAPED.

ABB^{-1} = Inverse of the partition of the aerodynamic influence coefficients matrix resulting from the influence of the body on the body written on logical tape NTAPED.

SUBPROGRAM CALLED: SINVRT (see Appendix B)

ERROR RETURNS: If an error occurs in the inverstion of the matrix, the message "ERROR IN INVERSION OF BODY MATRIX" will be written on the output tape. The values for IRR1, IRR2, and SCALE returned by subroutine SINVRT are also written on the output tape.

RESTRICTIONS: The matrix to be inverted must not exceed the order 100.

STORAGE: $10686_{10} = 24676_8$

SUBJECT: FORTRAN IV Subroutine INVRW

PURPOSE: For the wing-body case, to invert the reduced aerodynamic influence coefficients matrix.

METHOD: The reduced aerodynamic influence coefficients matrix is read from logical tape NTAPEE into core. The matrix is inverted in core by the single-precision inversion subroutine SINVRT. The matrix inverse is then stored on logical tape NTAPEE behind original matrix with an end-of-file separating the two matrices. An end-of-file is also written after the inverted matrix. The tape is then rewound.

USAGE: CALL INVRW

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION ARWW(102, 102)

Input: NTAPEE = Logical number of scratch tape.

NWING = Number of wing panels.

ARWW = Reduced aerodynamic influence coefficients matrix.

Output: $ARWW^{-1}$ = Inverse of the reduced aerodynamic influence coefficients matrix.

SUBPROGRAM CALLED: SINVRT (see Appendix B)

ERROR RETURNS: If an error occurs in the inversion of the matrix, the message "ERROR IN INVERSION OF REDUCED MATRIX" will be written on the output tape. The values of IRR1, IRR2, and SCALE returned by subroutine SINVRT are also written on the output tape.

RESTRICTIONS: The matrix to be inverted must not exceed the order 100.

STORAGE: $10578_{10} = 24522_8$

SUBJECT: FORTRAN IV Subroutine INVW

PURPOSE: For the wing-alone case, to invert the aerodynamic influence coefficients matrix and to store both the matrix and its inverse on a scratch tape.

METHOD: The aerodynamic influence coefficients matrix previously computed is read from logical tape NTAPEA into core and simultaneously stored on logical tape NTAPEE, where NTAPEA and NTAPEE are logical tape numbers of scratch tapes stored in COMMON. The matrix in core is inverted by a single precision inversion Subroutine SINVRT. The inverse is stored behind the original matrix on logical tape NTAPEE with an end-of-file mark separating the two matrices. An end-of-file mark is also written after the inverted matrix. Both tapes are then rewound.

USAGE: CALL INVW

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION AWW(102, 102)

Input: AWW = Aerodynamic influence coefficients matrix on logical tape NTAPEA.

Output: AWW = Aerodynamic influence coefficients matrix written on logical tape NTAPEE.

AWW^{-1} = Inverse aerodynamic influence coefficients matrix written on logical tape NTAPEE.

SUBPROGRAM CALLED: SINVRT (See Appendix B)

ERROR RETURNS: If an error occurs in the inversion of the matrix the message ERROR IN INVERSION OF WING ONLY MATRIX will be written on the output tape. The values for IRR1, IRR2, and SCALE as returned by subroutine SINVRT are also written on the output tape.

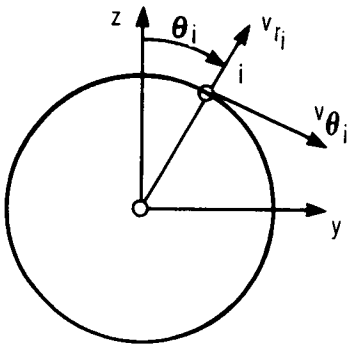
RESTRICTIONS: The matrix must not exceed the order 100.

STORAGE: $10598_{10} = 24546_8$

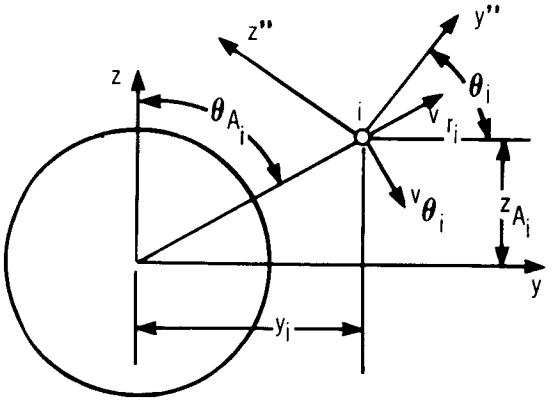
SUBJECT: FORTRAN IV Subroutine KARMOR

PURPOSE: For the isolated body, to compute the velocity components induced by source and doublets and the resulting pressures, forces, and moments.

METHOD: The body, considered to be made up of a series of K-truncated cones, is defined by the radius R at a series of stations X along its length.



CONTROL POINT ON BODY SURFACE



CONTROL POINT ON ARBITRARY PANEL

In general, the source strength of the ith segment is given by

$$T_i = \frac{\left(\frac{dr}{dx}\right)_i - \sum_{k=1}^{i-1}\left[a_{BBS_{ik}} - \left(\frac{dr}{dx}\right)_i u_{BBS_{ik}}\right] T_k}{a_{BBS_{ii}} - \left(\frac{dr}{dx}\right)_i u_{BBS_{ii}}}$$

where

$$\left(\frac{dr}{dx}\right)_i = \frac{r_i - r_{i-1}}{x_i - x_{i-1}}$$

$$u_{BBS_{ik}} = -\cosh^{-1}\frac{x_i - x_k}{\beta\, r_i}, \text{ the axial velocity}$$

$$a_{BBS_{ik}} = \beta\sqrt{\frac{(x_i - x_k)^2}{(\beta\, r_i)^2} - 1}$$

The strength of the doublets for the ith segment

$$T_{D_i} = \frac{\alpha - \left(\frac{dz_c}{dx}\right)_i + \sum_{k=1}^{i-1} \left[a_{BBD_{ik}} - \left(\frac{dr}{dx}\right)_i u_{BBD_{ik}} \right] T_{D_k}}{a_{BBD_{ii}} - \left(\frac{dr}{dx}\right)_i u_{BBD_{ii}}}$$

where $\left(\frac{dz_c}{dx}\right)_i = \frac{z_{c_i} - z_{c_{i-1}}}{x_i - x_{i-1}}$, the slope of the body camber line

$$u_{BBD_{ik}} = \beta \cos \theta_i \sqrt{\frac{(x_i - x_k)^2}{\beta^2 r_i^2} - 1},$$ the axial velocity component

$$a_{BBD_{ik}} = \frac{-\beta^2}{2} \cos \theta_{A_i} \left[\cosh^{-1} \left(\frac{x_i - x_k}{\beta r_i} \right) + \left(\frac{x_i - x_k}{\beta r_i} \right) \sqrt{\frac{(x_i - x_k)^2}{(\beta r_i)^2} - 1} \right]$$

x_k = x coordinate of the origin of the kth line singularity

The pressure coefficients on the body due to the line sources and doublets

$$C_{p_{B_i}} = 2u_{B_i} + \beta^2 u_{B_i}^2 - v_{r_{B_i}}^2 - v_{\theta_{B_i}}^2$$

$$u_{B_i} = \sum_{k=1}^{K} \left(u_{BBS_{ik}} \cdot T_k + u_{BBD_{ik}} \cdot T_{D_k} \right)$$

$$v_{r_{B_i}} = \sum_{k=1}^{K} \left(v_{r_{BBS_{ik}}} \cdot T_k + v_{r_{BBD_{ik}}} \cdot T_{D_k} \right)$$

$$v_{\theta_{B_i}} = \sum_{k=1}^{K} \left(v_{\theta_{BBS_{ik}}} \cdot T_k + v_{\theta_{BBD_{ik}}} \cdot T_{D_k} \right)$$

The lift, drag, and moment of the body due to the line and doublets are computed as follows:

1) The lift of a segment of the body between x_i and x_{i-1} is

$$\left(\frac{\Delta L}{q}\right)_{ij} = \frac{C_{p_{B_j}}}{2}(x_i - x_{i-1})(r_i + r_{i-1}) \cos \theta_j \cdot \Delta \theta_j$$

the drag for this segment is

$$\left(\frac{\Delta D}{q}\right)_{ij} = \frac{C_{p_{B_j}}}{2}(r_i^2 + r_{i-1}^2) \cdot \Delta \theta_j$$

where q is the dynamic pressure and the subscript j denotes a point on the circumference of the body.

2) The total lift, drag, and pitching moment is obtained by summing that of all the segments.

$$L_B = \sum_{j=1}^{J} \sum_{i=1}^{K} \left(\frac{\Delta L}{q}\right)_{ij}$$

$$D_B = \sum_{j=1}^{J} \sum_{i=1}^{K} \left(\frac{\Delta D}{q}\right)_{ij}$$

$$M_B = \sum_{j=1}^{J} \sum_{i=1}^{K} \left[\left(\frac{\Delta D}{q}\right)_{ij} \left(r_i \cos \theta_j - \bar{z}\right) - \left(\frac{\Delta L}{q}\right)_{ij} \left(\frac{x_i + x_{i-1}}{2} - \bar{x}\right)\right]$$

where the moments are computed about the point $(\bar{x}, \bar{z})$

The velocity components on the wing panels due to line sources and doublets are calculated by the summation formulas following equation (104) of Part I.

USAGE:

CALL KARMOR (NBODYS, NTHETA, CPCALC, VOUT, ZA, ARA, XP, ZP, XB, R, THETAB, ZDELTA, XC, YC, ZC, THETA, U, VV, WW, AN1, CPBB, BBCL, BBCD, BBCM)

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION XB(NBODYS), R(NBODYS), THETAB (NTHETA), ZDELTA(NBODYS), XC(NWING + NBODY), YC(NWING + NBODY), ZC(NWING + NBODY), U(NWING), VV(NWING), WW(NWING), AN1(NWING), CPBB(51, 10), THETA(NTHETA), ALPHAC(51), DTHETA(10), T(51) TC(51), TX(51), UB(51, 10), VB(51,10), VTB(51, 10)

Input:

NBODYS	=	Number of points defining the body line source segments.
NTHETA	=	Number of θ's around the body.
CPCALC	=	0., calculation of pressure coefficients to use linear equation.
	$\neq$	0., calculation of pressure coefficients to use non-linear equation.
VOUT	=	0., velocity components are not requested to be printed.
	=	1., velocity components are re- to be printed.
ZA	=	Height of wing reference plane above body axis.
ARA	=	Angle of attack with respect to body axis.
XP	=	X-coordinate of point about which the pitching moments are to be computed.
ZP	=	Z-coordinate of point about which the pitching moments are to be computed.
XB	=	Array of x-coordinates of the body line source and doublet segments.
R	=	Array of body radii defined at the XB coordinates.
THETAB	=	Array of output points around the circumference of the body.
ZDELTA	=	To define the body camber, the array of z distances defined at the XB coordinates.
XC	=	Array of the panel control point x-coordinates.

	YC	=	Array of the panel control point y-coordinates.
	ZC	=	Array of the panel control point z-coordinates.
	THETA	=	Array of the angles between the plane of panels and a plane parallel to the x-y plane.
Output:	U	=	The x component of velocity on the wing panels due to body sources and doublets.
	VV	=	The y component of velocity on the wing panels due to body sources and doublets.
	WW	=	The z component of velocity on the wing panels due to body sources and doublets.
	AN1	=	The normal component of velocity on the wing panels due to body sources and doublets.
	CPBB	=	The coefficients of pressure on the isolated body.
	BBCL	=	Lift of the body.
	BBCD	=	Drag of the body.
	BBCM	=	Pitching moment of the body.

SUBPROGRAM CALLED: ATAN2, ALOG, SIN, SQRT, COS (Built-in functions)

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $2996_{10} = 5664_8$

SUBJECT: FORTRAN IV Subroutine MDMATE

PURPOSE: To compute the matrix resulting from the application of the Lagrange multipliers technique in minimizing the drag for given wing constraints.

METHOD: The minimum drag of a wing in the presence of a body for given wing C_L and pitching moment may be determined by applying the method of Lagrange multipliers to the system of equations defining the drag. A function F is defined in terms of the pressure difference of the wing panels, p_{W_i}, and the auxiliary variables (Lagrange multipliers) λ_1 and λ_2.

$$F = D + \lambda_1 (L - \overline{L}) + \lambda_2 (M - \overline{M})$$

The condition for minimum drag for the constraints that L = L and M = M is found by setting the partial derivatives of F with respect to the variable p_{W_i}, $\lambda 1$, and $\lambda 2$ equal to zero. The result is the following system of linear equations, and the pressure difference across the wing panels may now be determined for minimum drag.

$$\begin{bmatrix} -2A_1 a_{11} & -(A_1 a_{12} + A_2 a_{21}) \cdots & -A_1 & -x_1 A_1 \\ -(A_2 a_{21} + A_1 a_{12}) & & -A_2 & \cdot \\ \vdots & & \vdots & \vdots \\ & & -A_N & -x_N A_N \\ -A_1 & \cdot\ -A_N & 0 & 0 \\ -x_1 A_1 & \cdot\ -x_1 A_N & 0 & 0 \end{bmatrix} \cdot \begin{Bmatrix} p_{W_i} \\ p_{W_2} \\ \vdots \\ p_{W_N} \\ \lambda_1 \\ \lambda_2 \end{Bmatrix} = \begin{Bmatrix} A_1(\bar{n}_1 + \bar{n}_{B_1}) \\ A_2(\bar{n}_2 + \bar{n}_{B_2}) \\ \vdots \\ A_N(\bar{n}_N + \bar{n}_{B_N}) \\ \overline{L} \\ \overline{M} \end{Bmatrix}$$

where $\bar{n}_i$ = The downwash on the wing due to the body sources.

$\bar{n}_{B_i}$ = The downwash on the wing resulting from the cancelation of normal velocity on the body due to the wing thickness.

If only the lift is constrained, the row and column of the matrix corresponding to λ_2 are omitted and p_{W_i} is found by inverting as before.

In this subroutine, the coefficients of the system of equations are computed and the inverses for the two cases are stored on a scratch tape in preparation for wing optimization by a later link. The matrices are stored with an end-of-file mark separating them.

USAGE:

CALL MDMATE

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION WW(110, 110), XBAR(200), AREA(200)

Input:	XBAR	=	Array of panel centroid x-coordinates.
	AREA	=	Array of panel areas.
	NBODY	=	Number of body panels.
	NWING	=	Number of wing panels.
	WW	=	Reduced matrix of the aerodynamic influence coefficients.
	NTAPEE	=	Logical tape number from which the reduced matrix of the aerodynamic influence coefficients is read.
	NTAPEA	=	Logical tape number on which the drag minimization matrices is stored.
Output:	WW	=	Drag minimization matrix written on NTAPEA.
	WW^{-1}	=	Inverse of the drag minimization matrix constrained for wing lift written on NTAPEA.
	WW^{-1}	=	Inverse of the drag minimization matrix constrained for wing lift and pitching moment written on NTAPEA.

SUBPROGRAM CALLED:

SINVRT (see Appendix B)

ERROR RETURNS:

If an error occurs in the inversion of the matrix, a message will be written on the output tape "ERROR IN INVERSION OF DRAG MINIMIZATION MATRIX" followed by the error codes from subroutine SINVRT.

RESTRICTIONS: The matrix to be inverted must not exceed the order 102.

STORAGE: $13020_{10} = 31334_8$

SUBJECT: FORTRAN IV Subroutine MEAN

PURPOSE: To smooth a set of data points defining a single-valued function of one variable.

METHOD: A sliding operation that deals with four points at a time $\left((x_i, y_i), i = 1, 2, 3, 4\right)$ is employed. Points whose ordinates may be changed are the interior two points at each setting of the operation. The abscissae of the input points are not changed. A straight line is drawn from (x_1, y_1) to (x_3, y_3) and another straight line from (x_2, y_2) to (x_4, y_4). If the abscissa of the intersection of these two lines lies in the interval (x_2, x_3), then the points are left as is, and the operation slides over one point and begins anew. If the lines are parallel or if the abscissa of the intersection is outside (x_2, x_3), then new ordinates y_2 and y_3 are calculated. An ordinate on the first line is evaluated at the abscissa x_2; similarly an ordinate on the second line is evaluated at the abscissa x_3. The new values y_2 and y_3 are the mean values of the original ordinates and new ordinates so calculated.

This operation slides along the set of points until all interior points have been subjected to smoothing. Five such passes made over the curve at each call of MEAN.

USAGE: CALL MEAN (X, Y, N)

Input: X = Array of abscissae: a strictly monotone sequence
Y = Array of ordinates to be smoothed
N = Number of points

Output: Y = Smoothed ordinates (the first and last points are not subjected to smoothing)

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: The abscissae must be strictly monotonic; thus vertical slopes are excluded.

STORAGE: $163_{10} = 243_8$

SUBJECT: FORTRAN IV Function MERR

PURPOSE: To write error messages for subroutines in the geometric definition portion of the program.

METHOD: An error array NU is given. NU(1) is an error code that should be the same as a given success code IOK if an error condition does not exist. NU(2), if greater than zero, is the number of an output tape on which an error message can be written. NU(3) is an error message limiter; a message is not to be written unless NU(3) is greater than 1. M is an error message number.

If NU(1) = IOK, the function value (K) is set to zero, and no further action is necessary. Otherwise, NU(3) is decreased by 1; then if NU(2) $\leq$ 0 or if NU(3) $\leq$ 0, K is set to -1 and return is made to the calling program. Otherwise, the following message is written:

"ERROR i, CODE j, IN SUBROUTINE xxxxxx DURING GEOMETRIC DEFINITION"

where i = M, j = NU(1), and xxxxxx is the given name of the subroutine in which the error occurred.

USAGE:

DIMENSION NU(3)

K = MERR (NU, IOK, M, SR)

Input: NU = Error array.

IOK = Success code.

M = Error message number.

SR = Subroutine name (alphameric).

Output: Message on error tape.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $87_{10} = 127_8$

SUBJECT: FORTRAN IV Function NWEED

PURPOSE: To remove excess points from an array of points in m-space.

METHOD: Each point is compared with the previous point. If all corresponding coordinates are the same with given tolerances, that point is eliminated by shifting the balance of the array and reducing the number of points by 1. However, if the last point is too near its neighbor, the neighbor is discarded and the last point retained.

USAGE:

```
DIMENSION  A(M, N), EP(M)
NN = NWEED (A, N, M, EP)
```

Input:
- A = Given array of points.
- N = Number of points in A.
- M = Space dimension.
- EP = Array of tolerances, one for each coordinate.

Output:
- A = Weeded array of points.
- NN = Number of points in the weeded array (NN may have the same storage location as N).

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: $N > 1$ and $M > 0$. If any $EP(i) \leq 0$, no weeding will be done. For weeding only on selected coordinates, EP is set to some large value for the remaining coordinates.

STORAGE: $153_{10} = 231_8$

SUBJECT: FORTRAN IV Program OPCAM

PURPOSE: To interpret data commands and control flow through the Program for Analysis and Design of Supersonic Wing-Body Combinations.

METHOD: The program first calls subroutine OPCAMI, which carries out the following operations: (1) assigns logical tape numbers to tape symbols (in COMMON); (2) obtains the data and places it in COMMON; (3) writes the program title on the output tape; (4) transfers the data cards to a scratch tape, which is then used as the input tape for the remainder of the program; (5) writes the data cards on the output tape.

Then function INTURP is used to read a data card, to write its contents on the output tape, and to see whether it is a command card. A command card has the words DEFINE, DEF1NE, PANEL, AERODY, AREODY, END OF or END OF in columns 1-6; the contents of the remaining columns are written on the output tape but are not otherwise used. If the data card is a command card, the appropriate action is taken; otherwise, successive cards are read until a command card is found (this feature permits the insertion of comment cards or blank cards immediately before any command card).

The command DEFINE (or DEF1NE) results in a call to subroutine GEOMD. On return, data cards are again read until a command card is encountered. If any GEOMD data cards were not read because of error, they will be listed by INTURP.

Subroutine TFLAT is called when a PANEL card is read. If TFLAT detects an error, it skips over data cards until the input tape is positioned just in front of a DEFINE or END OF card; OPCAM then goes on to process this card in the usual way. But if TFLAT signals success, then the argument NOBODY (set by TFLAT) is examined. If NOBODY = -1, meaning a body-alone case, OPCAM goes on to read the next data card; otherwise, OPCAM calls subroutine PANEL.

OPCAM calles subroutine AERO when an AERODY (or AERODY) card is read. An END OF (or END OF) card results in program termination.

SUBPROGRAMS CALLED:	AERO CLOK GEOMD INTURP TFLAT PANEL OPCAM1
ERROR RETURNS:	Program errors are dealt with in the program section where they occur.
RESTRICTIONS:	Each time that INTURP reads and writes a data card, OPCAM writes (on the same line, near the right margin) the word TIME followed by the time of day. The time of day is found by a subroutine CLOK. The basic program deck contains a dummy subroutine CLOK which merely returns blanks. A proper subroutine CLOK may replace the dummy at the installation where the program is used.
STORAGE:	$152_{10} = 230_8$

SUBJECT: FORTRAN IV Subroutine OPCAMI

PURPOSE: To perform initialization chores for the main program OPCAM.

METHOD: Tape symbols are assigned as follows:

Symbol	Tape Number	Description
LIN	5	System input tape.
LO	6	System output tape.
LI	9	Input tape used by remainder of program.
LA	1	Scratch tapes.
LB	2	
LC	3	
LD	4	
LE	7	
LF	8	

Subroutine DATE is called to obtain the date. A title is written on the output tape. Tape LI is rewound. Then all data cards are read with FORMAT(13A6, A2) from tape LIN, written on tape LO with FORMAT(1H 14A6), and written on tape LI with FORMAT(13A6, A2); the last data card is signaled by the word END OF or END 0F (in case the letter O is mistakenly keypunched as the number zero) in columns 1-6. Then an EOF is written on tape LI, and it is rewound.

USAGE: COMMON DAT(2), LA, LB, LC, LD, LE, LF, LI, LO

CALL OPCAMI

Output: DAT = Date (alphameric)

LA, LB, LC, LD, LE — Scratch tape numbers.

LF = Scratch tape number.

LI = Input tape number.

LO = Output tape number.

SUBPROGRAM CALLED: DATE

ERROR RETURNS: None

RESTRICTIONS: An END OF card must be placed at the end of the data deck. If the END OF card does not appear in the data deck, the job will be terminated by the EOF on the system input tape. If the END OF card is not the physically last card, only the data preceding it will be read by the remainder of the program.

STORAGE: $235_{10} = 353_8$

SUBJECT: FORTRAN IV Subroutine OPTIM3

PURPOSE: Given four successive points on the graph of a function of a single variable, to generate additional points between one selected pair of the given points, the spacing of the additional points to be such that straight-line segments joining them will represent the given function with a prescribed maximum error.

METHOD Four successive arguments x_i and their respective ordinates y_i are input to the subroutine, together with a tolerance z and integer k to indicate the interval (x_k, x_{k+1}) to consider. OPTIM3 then generates the biquadratic interpolating function described in subroutine BITURP. This function is a linear combination of two quadratics on the interval (x_2, x_3), hence it is there a cubic; on either of the other possible intervals it is a quadratic. In any case, the function has a continuous first derivative, unlike a Lagrangian interpolation polynomial of any order.

Having generated the interpolation function φ, OPTIM3 computes its function value y, its slope S, and its radius of curvature R, all at some particular x (at x_k in the first instance). In a small neighborhood of (x, y) the function φ can be represented by a circular arc of radius R, tangent at (x, y) to φ. OPTIM3 then computes the central angle of that arc so that the arc and its chord have a maximum separation, in the y-direction, of z. Then the x-component Δx of that chord is the distance from the current argument x to the next argument (unless x_{k+1} has been reached), and the process continues.

USAGE:

DIMENSION X(4), Y(4,) A(200), B(200)

CALL OPTIM3 (X, Y, Z, A, B, K, L, M)

Input:
- X = Array of x-coordinates of given points.
- Y = Array of y-coordinates of given points.
- Z = Allowable tolerance (in y direction).
- K = Subscript of x at lower end of interval to be interpolated.

Output:
- A = Array of x-coordinates of interpolated points. A(1) = X(K), A(2) = x of first interpolated point, etc.

B = Array of y-coordinates of interpolated points. B(1) = Y(K), B(2) = y of first interpolated point, etc.

L = One more than the number of interpolated points generated by the routine.

M = Error indicator, which is 1 if success.

SUBPROGRAMS CALLED: ACOS, ATAN, COS, SQRT (Built-in functions)

ERROR RETURNS: M = 0 if X(1) = X(4); M = 2 if L = 200 and more points are needed.

RESTRICTIONS: The x_i must form a strictly monotonic sequence, though not necessarily equally spaced. Note that the A and B arrays contain the first given point, but not the last.

STORAGE: $594_{10} = 1122_8$

SUBJECT: FORTRAN IV Subroutine OPTMW

PURPOSE: For wing alone, to compute the pressure difference across wing panels for a wing optimized for minimum drag.

METHOD: The method is outlined in the description of subroutine MDMATE. From the matrix inverse computed in MDMATE, the following multiplication yields the pressure difference across the panels.

$$\begin{Bmatrix} p_{W_1} \\ p_{W_2} \\ \vdots \\ p_{W_N} \\ \lambda_1 \\ \lambda_2 \end{Bmatrix} = \begin{bmatrix} -2A_1 a_{11} & -(A_1 a_{12} + A_2 a_{21}) \cdots & & -A_1 & -x_1 A_1 \\ -(A_2 a_{21} + A_1 a_{12}) & \cdot & & -A_2 & \cdot \\ \vdots & & & \vdots & \vdots \\ \cdot & & & -A_N & -x_N A_N \\ -A_1 & \cdot & \cdot\ -A_N & 0 & 0 \\ -x_1 A_1 & \cdot & \cdot\ -x_1 A_N & 0 & 0 \end{bmatrix}^{-1} \cdot \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \bar{L} \\ \bar{M} \end{Bmatrix}$$

If only the lift is constrained, the row and column of the matrix corresponding to λ_2 is omitted and p_{W_i} is found as before.

USAGE: CALL OPTMW(NW, NTAPEX, A, B, CONSNT, CLBAR, XCPBAR, RFAREA, AREA, CL)

DIMENSION A(1), B(1), AREA(1), CL(1)

Input:

NW	=	Number of wing panels.
NTAPEX	=	Logical number of tape from which the inverse of the minimum drag matrix is read.
A	=	Dummy array used by the subroutine.
B	=	Dummy array used by the subroutine.
CONSNT	=	0., indicates wing optimization for wing C_L constraint.
	=	1., indicates wing optimization for wing C_L and pitching moment constraints.
CLBAR	=	Wing coefficient of lift constraint.

		XCPBAR	= x-coordinate of the wing center of pressure constraint.
		RFAREA	= Wing reference area.
		AREA	= Array of wing panel areas.
	Output:	CL	= Array of the pressure difference across the wing panels optimized for minimum drag.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $131_{10} = 203_8$

SUBJECT: FORTRAN IV Subroutine OPTMWB

PURPOSE: In the presence of a body, to compute the pressure difference across wing panels for a wing optimized for minimum drag.

METHOD: The method is outlined in the descritpion of subroutine MDMATE. From the matrix inverse computed in MDMATE the following multiplication is carried out yielding the pressure difference across the panels.

$$\begin{Bmatrix} p_{W_1} \\ p_{W_2} \\ \vdots \\ p_{W_N} \\ \lambda_1 \\ \lambda_2 \end{Bmatrix} = \begin{bmatrix} -2A_1 a_{11} & -(A_1 a_{12} + A_2 a_{21}) \cdots & -A_1 & -x_1 A_1 \\ -(A_2 a_{21} + A_1 a_{12}) & & -A_2 & \cdot \\ \vdots & & \vdots & \vdots \\ & & -A_N & x_N A_N \\ -A_1 & \cdot\ -A_N & 0 & 0 \\ -x_1 A_1 & \cdot\ -x_1 A_N & 0 & 0 \end{bmatrix}^{-1} \cdot \begin{Bmatrix} A_1(\bar{n}_1 + \bar{n}_{B_1}) \\ A_2(\bar{n}_2 + \bar{n}_{B_2}) \\ \vdots \\ A_N(\bar{n}_N + \bar{n}_{B_N}) \\ \bar{L} \\ \bar{M} \end{Bmatrix}$$

where $\bar{n}_i$ = The downwash on the wing due to body sources

$$n_{B_i} = \sum_{j=1}^{NBODY} D_{ij} \cdot (-n_{B_j})$$ is the downwash on the wing due to the cancellation of the normal velocity on the body due to wing thickness.

D_{ij} = The interference influence coefficients matrix.

n_{Bj} = Downwash on the body due to wing sources

$\bar{L} = C_L \cdot S_W$ is the lift constraint

$\bar{M} = \bar{X} \cdot \bar{C}_L \cdot S_W$ is the pitching moment constraint

If only the lift is constrained, the row and column of the matrix corresponding to λ_2 is omitted and p_{W_i} is found as before.

CALL OPTMWB(NW, NB, NTAPEX, NTAPEY, NTAPEZ, A, B, ALPHA, ALPHAX, AREA, CONSNT, CLBAR, XCPBAR, RFAREA, CL)

DIMENSION A(1), B(1), ALPHA(1), ALPHAX(1), AREA (1), CL(1)

Input:

NW	=	Number of wing panels.
NB	=	Number of body panels.
NTAPEX	=	Logical number of the tape from which the inverse of the minimum drag matrix is read.
NTAPEY	=	Logical number of the tape from which D is read.
NTAPEZ	=	Logical number of the tape from which n_B is read.
A	=	Dummy array used by the subroutine.
B	=	Dummy array used by the subroutine.
ALPHA	=	Dummy array used by the subroutine.
ALPHAX	=	Array of downwash values on the wing due to body sources.
AREA	=	Array of panel areas.
CONSNT	=	0., wing optimization for wing C_L constraint
	=	1., wing optimization for wing C_L and pitching moment constraints.
CLBAR	=	Wing coefficient of lift constraint.
XCPBAR	=	x-coordinate of the wing center of pressure constraint.
RFAREA	=	Wing reference area.

Output:

CL	=	Array of the pressure difference across the wing panels optimized for minimum drag.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $249_{10} = 371_8$

SUBJECT: FORTRAN IV Subroutine OUTB

PURPOSE: To print out the data for body panels in a prescribed format.

METHOD: The data for the region of the body represented by panels are printed in the following format:

THETA (DEG.) ROW NO.	θ_1	θ_2	. . .	θ_{NTHETA}
1	$Z_{1,1}$	$Z_{1,2}$	. . .	$Z_{1,NTHETA}$
2	$Z_{2,1}$			.
3	.			.
.	.			.
.	.			.
.	.			.
NROW	$Z_{NROW,1}$	. . . .		$Z_{NROW,NTHETA}$

The numbers running horizontally represent the angles, θ around the body starting at the crown line and proceeding to the keel line. The numbers running vertically represent the rows of panels starting from the furthest forward panel and working aft.

USAGE:

```
CALL OUTB (NTAPEO, NBODY, NTHETA, NROWB,
           THETA, Z)
DIMENSION  Z(NBODY), THETA(NTHETA)
```

Input:

NTAPEO	=	Logical number of output tape.
NBODY	=	Number of body panels.
NTHETA	=	Number of θ angles.
NROWB	=	Number of rows of panels.
THETA	=	θ angles.
Z	=	Data to be output.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $119_{10} = 167_8$

SUBJECT: FORTRAN IV Subroutine OUTPTB

PURPOSE: To write body paneling data on output tape as printout and on scratch tape for use in other program sections.

METHOD: Program uses direct write tape statements to write body paneling data on output tape and on scratch tape (see Appendix C for tape format). Program uses a tolerance of ±0.00001 and sets all small negative and positive values equal to 0. Program also checks and eliminates any trivial secondary panel parts. An input tolerance is used for this check.

USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description).

Input: NTAPE2
NWRITE

Output: NBODY

COMMON /COM1/ (See subroutine PANEL)

Input: NPER
NPER1
NPLANE
NPLN1

COMMON /COM2/ (See subroutine BODY)

Input: KPANEL
XCOR
YCOR
ZCOR
XINT
YINT
ZINT
XCEN
YCEN
ZCEN
XCON
YCON
ZCON
AREA
THETA
ALPHA
CHORD

CALL OUTPTB

SUBPROGRAM CALLED:	SQRT (Built-in function)
ERROR RETURNS:	None
STORAGE:	$1392_{10} = 2560_8$

SUBJECT: FORTRAN IV Subroutine OUTPTW

PURPOSE: To write wing paneling data on output tape as printout and on scratch tape for use in other program sections.

METHOD: Program uses direct write tape statements to write wing paneling data on output tape and on scratch tape (see Appendix C for tape format). Program uses a tolerance of ±0.00001 and sets all small negative and positive values equal to 0. Program also checks all secondary panel parts of two-part panels on inboard and outboard columns and eliminates any "small" parts. An input tolerance is used for this check.

USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description).

Input: NTAPE2
NWRITE

Output: NWING

COMMON /COM1/ (See subroutine PANEL)

Input: KSTART
KEND
NPER
NPER1
NPLANE
NPLN1

COMMON /COM2/ (See subroutine WING)

Input: KPANEL
XCOR
YCOR
ZCOR
XINT
YINT
ZINT
XCEN
YCEN
ZCEN
XCON
YCON
ZCON
AREA
THETA
ALPHA
CHORD

CALL OUTPTW

SUBPROGRAM CALLED:	None
ERROR RETURNS:	None
STORAGE:	$1612_{10} = 3114_8$

SUBJECT: FORTRAN IV Subroutine OUTW

PURPOSE: To print out the data for wing panels in a prescribed format.

METHOD: The data for the wing panels are output in the following format:

SPANWISE STATION	1	2	3 . . .	NCOL
CHORDWISE STATION				
1	$Z_{1,1}$	$Z_{1,2}$	. . . .	$Z_{1,NCOL}$
2	$Z_{2,1}$			
3	.			.
.	.			.
.	.			.
.	.			.
NROW	$Z_{NROW,1}$	. . . .		$Z_{NROW,NCOL}$

The numbers running horizontally represent NCOL columns of panels from inboard to outboard, and the numbers running vertically represent NROW rows of panels from the leading to the trailing edge.

USAGE: CALL OUTW (NTAPEO, NWING, NCOLW, NROWW, Z)

DIMENSION Z(NWING)

Input:

NTAPEO	=	Logical number of output tape.
NWING	=	Number of wing panels.
NCOLW	=	Number of chordwise columns of panels.
NROWW	=	Number of rows of panels in a chordwise column.
Z	=	Data to be output.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $119_{10} = 167_8$

SUBJECT: FORTRAN IV Subroutine PANEL

PURPOSE: To serve as control program for geometry paneling subroutines.

METHOD: Program first reads control card used in later subroutines to calculate coordinates of panel control point. Program then reads control card containing literal name and transfers to appropriate subroutine. Either the program transfers to the subroutine BODY or to the subroutine WING, or the program returns to the subroutine OPCAM.

The program contains the following labeled COMMON statement that occurs in all lower level geometry paneling subroutines:

COMMON/COM1/KODEB, KODEW, KODEWU, KODEI, KODEC, XPER, YPER, KOPTB, KOPTW, KOPTF, NUMS, KTYPE, KSTART, KEND, KINT, XI(16), YI(16), ZI(16), NPER, NPER1, NPLANE, NPLN1

KODEB = Code for body definition error.

KODEW = Code for wing definition error.

KODEWU = Code for error in wing thickness (or camber) calculations.

KODEI = Code for body-wing intersection definition error.

KODEC = Code for panel control point location. If KODEC = 0, panel control point is located on the streamwise chord through the panel centroid. If KODEC = 1, input value for YPER is used to locate control point.

XPER = Fraction of local streamwise panel chord at which panel control point is located. $0. \leq XPER \leq 1.$

YPER = Fraction of local spanwise panel chord at which panel control point is located. $0. < YPER \leq 1.$
NOTE: If YPER = 0, panel control point is located on the streamwise chord through the panel centroid.

KOTPB = Code for additional body panel card input. If KOPTB = 1, body panel corner points are input on cards. If KOPTB = 0, otherwise.

KOPTW = Code for additional wing panel card input. If KOPTW = 1, wing panel corner points are input on cards. If KOPTW = 0, otherwise.

KOPTF = Code for additional wing panel card input. If KOPTF = 1, actual wing surface coordinates are input on cards. If KOPTF = 0, otherwise.

NUMS = Number of airfoil sections for which wing surface coordinates are read as additional card input.

KTYPE, KSTART, KEND = Codes for internal program control.

KINT = Code for body-wing intersection. If KINT = 1, body-wing intersection has been requested and calculated. If KINT = 0, otherwise.

XI, YI, ZI = x, y, z coordinates of body-wing intersection.

NPER = Number of columns of panels plus one.

NPER1 = Number of columns of panels.

NPLANE = Number of rows of panels plus one.

NPLN1 = Number of rows of panels.

USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description).

Input: NTAPE1
NTAPE2
ND1
NREAD
NWRITE

CALL PANEL

SUBPROGRAMS CALLED: WING

BODY

INTURP

ERROR RETURNS: Error messages indicate whether error occurred in reading various geometry definitions from scratch tape.

STORAGE: $11208_{10} = 25710_8$

SUBJECT: FORTRAN IV Subroutine PARTV

PURPOSE: To partition velocity components due to the surface distribution of vorticity in preparation for use in later aerodynamic links.

METHOD: The velocity components are read from a scratch tape written by subroutine EVAL in the following form, where a column is written as a logical record.

$$\begin{bmatrix} U_{BB} & U_{BW} \\ V_{BB} & V_{BW} \\ W_{BB} & W_{BW} \\ U_{WB} & U_{WW} \\ V_{WB} & V_{WW} \\ W_{WB} & W_{WW} \end{bmatrix}$$

where the subscripts are given:

BB = Influence on the body due to the body

WB = Influence on the wing due to the body

BW = Influence on the body due to the wing

WW = Influence on the wing due to the wing

The velocity components are reordered to a partitioned form with an end-of-file separating the partitions.

$$\begin{bmatrix} U_{BB} \\ V_{BB} \\ W_{BB} \end{bmatrix} \begin{bmatrix} U_{WB} \\ V_{WB} \\ W_{WB} \end{bmatrix} \begin{bmatrix} U_{BW} \\ V_{BW} \\ W_{BW} \end{bmatrix} \begin{bmatrix} U_{WW} \\ V_{WW} \\ W_{WW} \end{bmatrix}$$

USAGE: CALL PARTV

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION U(100), V(100), W(100), UU(100), VV(100), WW(100)

Input: NTAPEB = Logical tape number from which the velocity components are read.

	NTAPED	= Logical tape number on which the velocity components are written.
	NWING	= Number of wing panels.
	NBODY	= Number of body panels.
	U, V, W	= Velocity components in random order on logical tape NTAPEB.
Output:	U, V, W	= Velocity components in a specified order written on logical tape NTAPED.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: $0 < NBODY \leq 100$

$0 < NWING \leq 100$

STORAGE: $986_{10} = 1732_8$

SUBJECT: FORTRAN IV Subroutine POLXN

PURPOSE: To find the intersections of an m-dimensional polygonal arc and a plane (or line, if m = 2). A polygonal arc is defined as a set of points consecutively connected by straight lines.

METHOD: Consider any two consecutive points in the polygonal arc P, say P_i and P_{i+1}. If P_i (or P_{i+1}) lies within a given distance EP of the plane, it is said to lie in the plane. If one point lies in the plane, that point is returned as an intersection. If both points P_i and P_{i+1} lie in the plane (i.e., the line segment $\overline{P_i\ P_{i+1}}$ is contained in the plane), only the end points of that line segment are returned as intersections. If P_i and P_{i+1} lie on opposite sides of the plane, the intersection of the line segment $\overline{P_i\ P_{i+1}}$ with the plane is returned. The subroutine stops processing points when a specified maximum number of intersections has been found. The point subcripts associated with any intersection are also returned.

USAGE:

```
DIMENSION   C(M+1), P(M, N), Q(M, MAX), LQ(2, MAX)
CALL POLXN (C, P, M, N, EP, MAX, Q, LQ, L)
```

Input:

- C = Coefficients of the plane $C_1x_1 + C_2x_2 + \ldots + C_Mx_M + C_{M+1} = 0.$
- P = Array of point coordinates $x_{11}, x_{21}, \ldots, x_{M1}, x_{12}, x_{22}, \ldots x_{M2}, \cdot$
- M = Dimension of hyperspace.
- N = Number of points in P.
- EP = Tolerance.
- MAX = Maximum number of intersections to be found.

Output:

- Q = Array of intersection points.
- LQ = Array of point numbers of points in P that are adjacent to points in Q; for instance, if an intersection in Q is between the fourth and fifth points of P, the corresponding elements of LQ are (4, 5); if point i in P lies in the plane (or within distance EP from it), then the point i appears in Q,

and the corresponding elements in LQ are (i, i).

L = Number of intersection points in Q; L = 0 if no intersections are found.

SUBPROGRAM CALLED: SQRT (Built-in function)

ERROR RETURNS: L will be returned as -1 if $M \leq 0$, or if $MAX \leq 0$, or if all of the coefficients C are zero.

RESTRICTIONS: If any points are identical and lie in the plane, each such point will be returned as an intersection.

STORAGE: $268_{10} = 414_8$

SUBJECT: FORTRAN IV Subroutine QRAT

PURPOSE: To find the coefficients of the quadratic that passes through three given points, and optionally to evaluate the quadratic at a given value of x.

METHOD: The quadratic equation is

$$y = C_1 + C_2 x + C_3 x^2$$

Let (x_i, y_i), $i = 1, 2, 3$ be the three given points. Then if no two points have the same x-coordinate,

$$C_3 = \left(\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}\right) \Big/ (x_3 - x_1)$$

$$C_2 = (y_3 - y_1) \Big/ (x_3 - x_1) - C_3 (x_3 + x_1)$$

$$C_1 = y_2 - (C_2 + C_3 x_2) x_2$$

The quadratic is evaluated for a given value of x if a code is positive.

USAGE: DIMENSION X(1), Y(1), C(3)

CALL QRAT(X, Y, IXY, KODE, XI, YI, C, NU)

Input:

X = Location of the x-coordinate of the first point.

Y = Location of the y-coordinate of the first point.

IXY = Skip number for X and Y. IXY = location of w_{i+1} minus the location of w_i, where w is either x or y and i = 1 or 2. For example, IXY = 1 if X and Y are each dimensioned 3. Or if an array $W = (x_1, y_1, x_2, y_2, x_3, y_3)$ then CALL QRAT (W, W(2), 2, ...) may be used to avoid shifting x_i and y_i to separate arrays

KODE = If KODE > 0, the quadratic (if found) will be evaluated at x = XI and the result stored in YI. Otherwise, XI and YI are not used.

XI = See KODE.

Output: YI = See KODE.

C = Array of quadratic coefficients $C = (C_1, C_2, C_3)$.

NU = An error code that is zero if QRAT is successful.

SUBPROGRAM CALLED: None

ERROR RETURNS: NU = 1 if any two of the given points have the same x-coordinate.

RESTRICTIONS: IXY may not be zero; it may be negative if the point coordinates are stored in descending core locations.

STORAGE: $157_{10} = 235_8$

SUBJECT: FORTRAN IV Subroutine READ

PURPOSE: To provide, as options, three methods of inputting an array of values.

METHOD: The subroutine allows control over the form of input to the program by means of a literal word on a data card. Two words are stored in the program Data Dictionary. The first control word is CONSTANT, of which the first six letters CONSTA are stored. The second word is GIVEN. If the word on the data card matches either of the two control words, transfer in the program is made accordingly. The third option is provided by arbitrary letters in the first six columns of the input card. The three options provided are:

1) If the first data card contains CONSTANT, the program expects one additional data card on which the first field (columns 1-10) contains a value. This value is then set in the array S.

2) If the first data card contains GIVEN, the program expects to read an array of values off a specified logical tape unit.

3) If the data card contains any arbitrary alphanumeric characters, the program expects to read an array of values from data cards punched in a seven-field, ten-digit format.

USAGE: CALL READ (NTAPEI, NTAPEO, NTAPEC, NM, S)

DIMENSION S(NM), DICT(2)

Input:	NTAPEI	=	Logical number of input tape.
	NTAPEO	=	Logical number of output tape.
	NTAPEC	=	Logical number of scratch tape, if option 2.
	NM	=	Number of values in the input array.
Output:	S	=	Array of values.

SUBPROGRAM CALLED: INTURP

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $131_{10} = 203_8$

SUBJECT: FORTRAN IV Subroutine REDUCE

PURPOSE: To form the "reduced" aerodynamic influence coefficients matrix and three additional matrices in preparation for use in later links.

METHOD: The aerodynamic influence coefficients matrix is composed of the normal velocity components induced by a unit pressure difference on each panel.

The matrix may be written in terms of the sub-matrices

$$\begin{bmatrix} [A_{BB}] & [A_{BW}] \\ [A_{WB}] & [A_{WW}] \end{bmatrix} \cdot \begin{Bmatrix} p_B \\ p_W \end{Bmatrix} = \begin{Bmatrix} -\ n_{BWS} \\ \frac{dZc}{dx} - (\alpha) - n_{WB} \end{Bmatrix}$$

where

A_{BB} = The influence on the body due to the body

A_{WB} = The influence on the wing due to the body

A_{BW} = The influence on the body due to the wing

A_{WW} = The influence on the wing due to the wing

n_{BWS} = The normal velocity component on the body due to wing sources

$\frac{dZc}{dx}$ = The slope of the wing panels

α = The angle of attack

n_{WB} = The normal velocity component on the wing due to the body

The matrix equation is solved for p_B and p_W.

$$\{p_B\} = [A_{BB}]^{-1}\left\{\{n_{BWS}\} + [A_{BW}]\{p_W\}\right\}$$

$$\{p_W\} = [A_R]^{-1}\left\{[A_{WB}][A_{BB}]^{-1}\{n_{BWS}\} + \left\{\frac{dZc}{dx} - \alpha - n_{BWS}\right\}\right\}$$

where $[A_R] = \left[[A_{WW}] - [A_{WB}][A_{BB}]^{-1}[A_{BW}]\right]$

is referred to as the "reduced" aerodynamic matrix.

The "reduced" matrix is then written on logical tape NTAPEE.

The additional matrices formed are

$$[D] = [A_{WB}][A_{BB}]^{-1}$$

and

$$[E] = [A_{BB}][A_{BW}]$$

Three matrices are written on logical tape NTAPEF in the order $[A_{BB}]^{-1}$, $[D]$, $[E]$ with an end-of-file written after each matrix.

USAGE:

CALL REDUCE

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION ABB(100, 100), AWB(100, 100), ABW (100), AWW(100), E(100), D(100)

EQUIVALENCE (E, D)

Input:

NWING	=	Number of wing panels.
NBODY	=	Number of body panels.
NTAPEA	=	Logical number of tape from which the aerodynamic influence coefficients are read.
NTAPED	=	Logical number of tape from which $[A_{WB}]$ and $[A_{BB}]^{-1}$ are read.
NTAPEE	=	Logical number of tape on which the reduced aerodynamic influence coefficients matrix is written.
NTAPEF	=	Logical number of tape on which $[A_{BB}]^{-1}$, $[D]$, and $[E]$ are written.
AWB	=	Matrix $[A_{WB}]$
ABB	=	Matrix $[A_{BB}]^{-1}$
ABW	=	Matrix $[A_{BW}]$
AWW	=	Matrix $[A_{WW}]$

Output:

AWW	=	Matrix $[A_R]$
ABB	=	Matrix $[A_{BB}]^{-1}$
D	=	Matrix $[D]$
E	=	Matrix $[E]$

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: NWING $\leq$ 100
NBODY $\leq$ 100

STORAGE: $20564_{10} = 50270_8$

SUBJECT: FORTRAN IV Subroutine RICH3A

PURPOSE: Given a sparse array of points on a space curve, to compute a dense array.

METHOD: A given tolerance, CHD, regulates the number of additional points to be generated on the space curve. Between adjacent points in the dense array, the maximum distance between arc and chord will be approximately less than CHD. (However, if CHD ≤ 0 or if there are fewer than four points in the sparse array, no interpolation takes place and the dense array will be identical to the sparse array.)

The given sparse-array points are projected into a plane. Additional points are interpolated in this plane by subroutine ENRYCH (using the tolerance CHD). An inverse projection is then performed, mapping the two-dimensional points back to the original three-space.

A more detailed description of the enriching process follows. As a first step the sparse array of points is transformed to an X', Y', Z' coordinate system. The transformation is determined by a given code, KD. Let P_1 (x_1, y_1, z_1) be the first point in the given sparse array P, and P_n (x_n, y_n, z_n) be the last point. If KD = 0, the X' Y' Z' origin is at P_1, and the X' axis passes through P_n; the X' Z' plane contains the point in P farthest from the line P_1 P_n. If KD = 1 or 2, the X' Y' Z' origin is at (x_1, y_1, 0), and the X' axis passes through the point (x_n, y_n, 0); the Z' axis is parallel to the Z axis. In the next step, the X' Z' coordinates of the points are treated as two-dimensional points by subroutine ENRYCH to produce a dense array of points. To find the corresponding Y' coordinate of each point, a two-dimensional interpolation is made by subroutine BITURP: the interpolation table contains the sparse array X' and Y' coordinates, with X' as the independent variable. For each dense array point the Y' coordinate is interpolated from the table. The interpolation is linear if KD = 1, or biquadratic if KD = 0 or 2. In the final step, the dense array X', Y', Z' coordinates are transformed to the original X, Y, Z system.

USAGE: DIMENSION P(3, N), Q(3, N), A(3, MAX/3), NU(3)

CALL RICH3A (N, P, Q, KD, MAX, CHD, NA, A, NU)

Input: N = Number of points in P.

P = Array of given points (sparse array).

KD = Code (see METHOD, above).

MAX = Maximum number of elements that may be stored in the A-array.

CHD = Chord-height tolerance that determines the spacing of interpolated points (see ENRYCH

NU = Error indicator array: NU(1) is not used on input. NU(2) and NU(3) are used only by ENRYCH and BITURP. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written.

Output: Q = Scratch array that may occupy the same storage space as P if P need not be preserved.

NA = Number of points in the A-array.

A = Array of enriched points (dense array).

NU = Error indicator array: NU(1) = 0 if success. NU(3) is discussed above.

SUBPROGRAMS CALLED:

UVECN
VCROS
DMAXL
ENRYCH
BITURP
TROTPT
SQRT (Built-in function)
TRAV

ERROR RETURNS: NU = -2, if KD = 0 and the endpoints of P are coincident. NU = -1 (corresponding to error code 3 from ENRYCH) if more than 200 interpolated points would be required between two points in P: either CHD is too small or the points in P (after projection) are ill-conditioned. RICH3A adds no interpolated points in this interval, and continues. NU = 1, 4, 5, or 6 if MAX is too small. NU = 2 (error 2 from ENRYCH) if four consecutive points in P (after

transformation to the X', Y' system) are not strictly monotonic in either X' or Y'. NU = 7 if KD = 1 or 2 and the endpoints of P have the same X and Y coordinates. NU = 11, 12, 13, or 14 (BITURP errors 1, 2, 3, or 4) if an error was detected during the inverse projection (see METHOD, above): this can happen only if the given sparse array is ill-conditioned. NU = 21 if an error occurs when finding the projection plane.

RESTRICTIONS: After transformation to the X', Y', Z' system (see METHOD, above), the given sparse array points must be monotonic in X'; no two X' coordinates may be the same. Since only the projections of the given points on the X'-Z' plane are enriched, too few points may be generated if the given points do not lie approximately in the plane.

STORAGE: 509_{10} 775_{8}

SUBJECT: FORTRAN IV Subroutine RICHNA

PURPOSE: To enrich (interpolate additional points) a set of body meridian lines or wing percent lines.

METHOD: A given array A of length MAX contains M arrays of N three-dimensional points each. Each array will be called a percent line. Since array A is also used for output, all the points are first shifted to the end of A. A given code, KOD, determines the type of interpolation (see subroutine RICH3A). If KOD $\neq$ 0, all points in each percent line are checked: a comment is written for each point having X and Y coordinates that are nearly the same (within a given tolerance EPS) as those of the previous point but for which the Z-coordinates are different. Then all percent lines are screened by function NWEED to remove any points too close to the preceding point; If KOD = 0, all three coordinates must nearly match the previous point before a point is removed. But if KOD > 0, only the X and Y coordinates must nearly match; thus any points for which a comment was written will be removed.

(NOTE: KOD = 0 for body meridian lines; KOD = 0, 1, or 2 for wing percent lines.) If a given tolerance CHD is less than or equal to 0, no additional points will be interpolated. A "header" or "table of contents" is inserted at the beginning of array A; this is followed by the point coordinates. The format of the completed array A is given in Appendix C.

USAGE:

```
DIMENSION   A(MAX), P(M), NU(3)

CALL RICHNA (A, MAX, M, N, P, EPS, CHD, KOD,
             NA, NU)
```

Input:

- A = Array of M percent lines of N points each.
- MAX = Length of array A.
- M = Number of percent lines.
- N = Number of points in each percent line.
- P = Array of labels that are angles (if body meridian lines) or percent values (if wing percent lines); these are used in the "header" of the final array (see Appendix C).
- EPS = Tolerance of test for coincident points.

CHD = Chord-arc tolerance that controls the number of interpolated points. See subroutine RICH3A.

KOD = Code which determines type of interpolation (see KD in subroutine RICH3A).

NU = Error indicator array: NU(1) is not used on input. NU(2) is an output tape number on which to write a message if an error is detected in some lower-level subroutine; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) =(NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written.

Output: A = Array of M enriched percent lines, with "header" that contains a label, number of points, and starting location in A for each percent line. See Appendix C for format.

NA = Total number of points in array A. The total number of elements used in A is 3 * (M + NA).

NU = Error indicator array: NU(1) = 0 if no errors are detected. NU(3) may have been reduced by a lower-level subroutine if an error occurred.

SUBPROGRAMS CALLED: NWEED
RICH3A

ERROR RETURNS: NU = -1 if NWEED detected an error; this could be caused by a very large value of EPS. For any other nonzero value of NU, see RICH3A.

RESTRICTIONS: MAX $\geq$ 3 * M * (N + 1)

STORAGE: $400_{10} = 620_8$

SUBJECT: FORTRAN IV Subroutine RITE

PURPOSE: To output the x, y, and z components of velocity induced by the various signularities.

METHOD: If a printout of the velocity components is requested, the x, y, and z components of velocity are written on the output tape. The components are identified in the printout as the x component, AXIAL(W); the y component, TRANSVERSE(V); and the z component, VERTICAL(W). By an input code, the proper identification relating the velocity components to the singularity is also printed.

USAGE: Call RITE (NFMT, NTAPEO, NP, NROW, NCOL, THETA, U, V, W)

DIMENSION NFMT(7), U(NP), W(NP)

Input: NFMT control code

NFMT(I) (I = 1, 7) = 0, output of velocity components not requested.

NFMT(1) = 1, velocity components on body panels resulting from body panel pressure singularities.

NFMT(2) = 2, velocity components on body panels resulting from wing panel pressure singularities.

NFMT(3) = 3, velocity components on body panels resulting from wing sources.

NFMT(4) = 4, velocity components on wing panels resulting from wing panel pressure singularities.

NFMT(5) = 5, velocity components on wing panels resulting from wing sources.

NFMT(6) = 6, velocity components on wing panels resulting from body panel pressure singularities.

NFMT(7) = 7, velocity components on wing panels body line sources and doublets.

NTAPEO = Logical number of output tape.

NP = Number of wing or body panels.

NROW	=	Number of rows of panels on wing or body.
NCOL	=	Number of chordwise columns of panels on wing or number of longitudinal columns of panels on body.
THETA	=	If body case, the θ angles around the body of the data output points.
U	=	x component of velocity.
V	=	y component of velocity.
W	=	z component of velocity.

SUBPROGRAMS CALLED: OUTB
OUTW

ERROR RETURNS: None

RESTRICTIONS: None

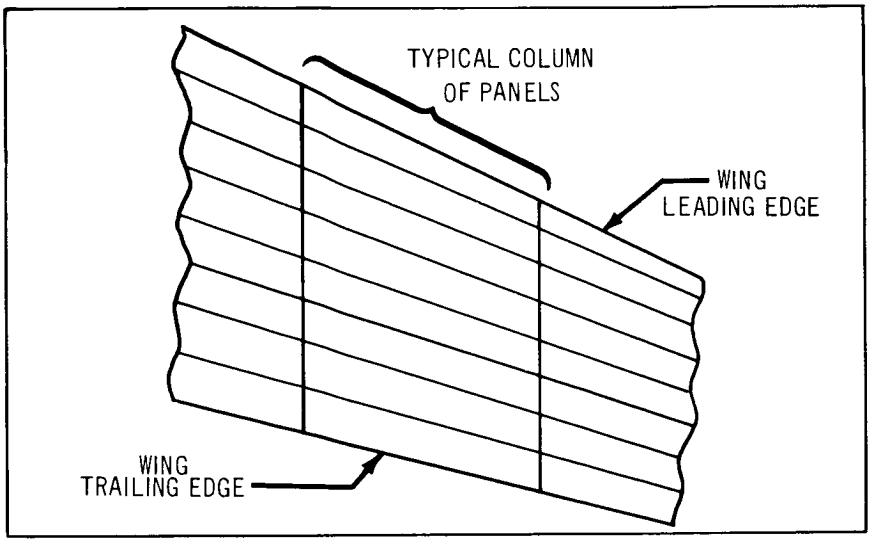
STORAGE: $409_{10} = 631_8$

SUBJECT: FORTRAN IV Subroutine SLOPEW

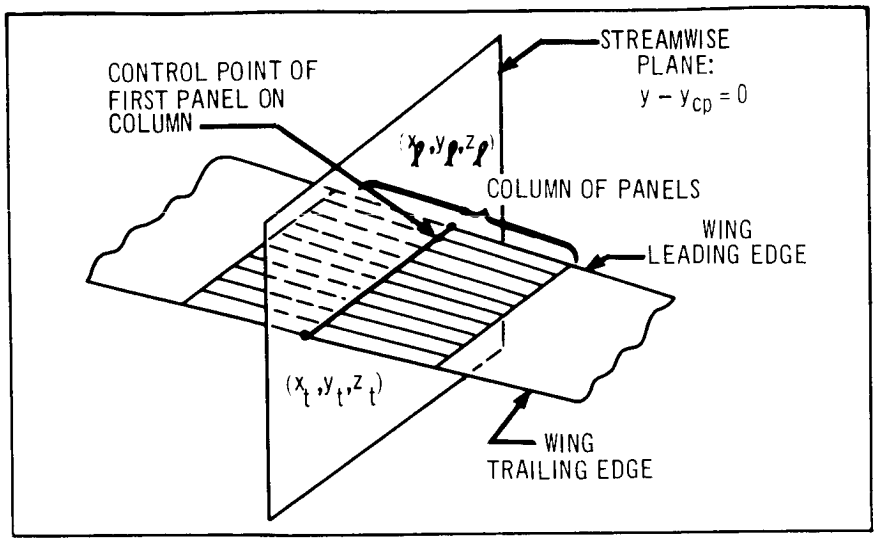
PURPOSE: To calculate camber and thickness ordinates and camber and thickness slopes for an airfoil.

METHOD: Program input can consist of single airfoil (or multiple airfoils) for which both an upper and lower surface are defined (see subroutine INPUTW). If a single airfoil is given, this airfoil is scaled accordingly and used to calculate camber and thickness for each column of panels on the wing. If multiple airfoils are given, each airfoil is reduced to a unit airfoil, scaled accordingly and used for camber and thickness calculations on but a single column of wing panels. Note that, for the latter case, as many airfoils as panel columns must be given on input cards.

For a given unit airfoil and the corresponding panel column, the program locates a local streamwise chord on the column, computes the chord length, and uses this value to scale the airfoil. For example, a typical panel column is represented below:



The program constructs a streamwise plane through the control point of the first panel on the column and intersects this plane with the wing leading and trailing edges. The panel side edges are streamwise; thus this construction is always possible and, because the control points for a given column of panels lie on the same streamwise chord, only one such construction is necessary. This is shown in the following sketch:



A chord length to be associated with this panel column is then calculated,

$$c = x_t - x_\ell$$

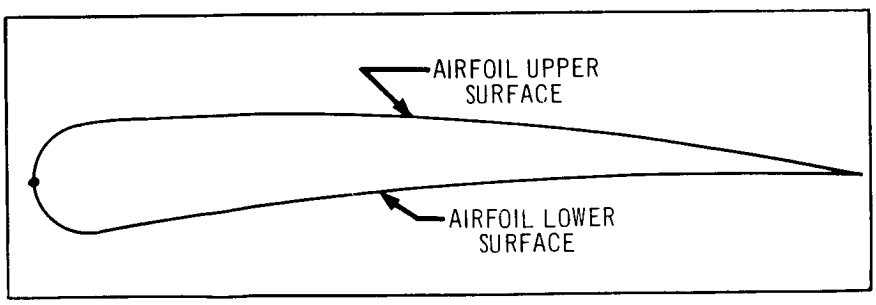
and this chord length is used to scale the airfoil so that,

$$x_k = x_{k_o} \cdot c$$

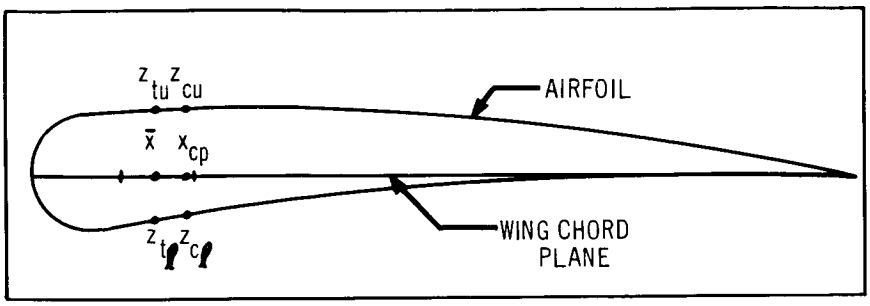
$$z_k = z_{k_o} \cdot c + \zeta$$

for which x_{k_o}, z_{k_o} are the x- and z-coordinates of the unscaled, unit airfoil, x_k, z_k are the coordinates of the scaled airfoil and ζ is the constant z-value for the wing.

Having scaled the airfoil to "fit" the wing for a particular column, the program then calculates camber and thickness ordinates for each panel of that column. To satisfy a condition of the program, the camber ordinate is computed at the panel control point and the thickness ordinate at the panel centroid. Because both the upper and lower surfaces of an airfoil are given as input (see sketch),



the program must interpolate on each surface for both the camber and thickness ordinate. As a result, four values are calculated



and used in camber and thickness calculations. For the camber ordinate,

$$z_{cam} = \frac{1}{2}(z_{cu} + z_{c\ell})$$

and for thickness,

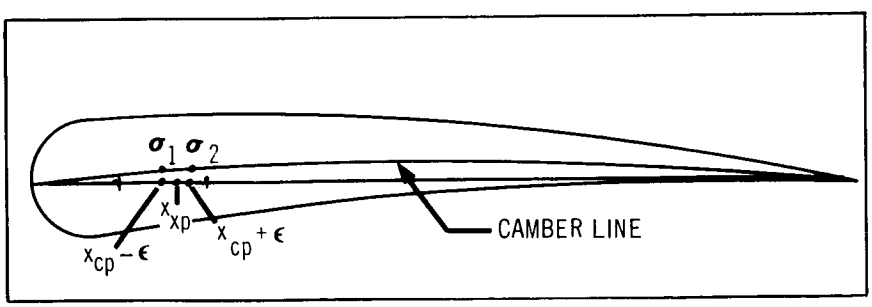
$$z_{thk} = \frac{1}{2}(z_{tu} - z_{t\ell}).$$

Using the camber and thickness ordinates that have been calculated, the program computes the camber and thickness slopes. As the camber is evaluated at the panel control point and the thickness at the centroid, the program constructs two distinct tables:

1) z_{cam} versus x_{cp}
2) z_{thk} versus $\bar{x}$

that are used for these calculations.

To obtain the camber slope at the x_{cp} for a given panel, the program evaluates the camber curve (1) at $x_{cp} - \epsilon$ and $x_{cp} + \epsilon$



for which

$$\epsilon = \frac{c}{10^5}$$

and c is the chord length.

The values σ_1, σ_2 of the curve at these two points are used to compute the camber slope,

$$\alpha_{cam} = \tan^{-1}\left(\frac{\sigma_2 - \sigma_1}{2\epsilon}\right)$$

Similarly, to obtain the thickness slope at the $\bar{x}$ of a given panel, the thickness curve (2) is evaluated at $\bar{x} - \epsilon$ and $\bar{x} + \epsilon$, and the values τ_1 and τ_2 of the curve at these two points are used,

$$\alpha_{thk} = \tan^{-1}\left(\frac{\tau_2 - \tau_1}{2\epsilon}\right)$$

As a last step the program smooths the camber and thickness slopes. The calculations of the slopes for a given column of panels result in two tables:

1) α_{cam} versus x_{cp}
2) α_{thk} versus $\bar{x}$

and each of these tables is smoothed through use of the subroutine MEAN.

The above calculations are used to obtain camber and thickness ordinates and camber and thickness slopes for panels of a column of the wing in all but two cases,

1) The inboard column on the wing for a body-wing case having a body contour in the intersection region;

2) The outboard wing column for wing having a nonstreamwise outboard wing edge.

For these two cases, the calculations are changed slightly because the edges of an arbitrary panel are not streamwise and the control points of all panels of the column do not lie on the same streamwise chord. A local streamwise chord must be constructed and a chord length determined for <u>each</u> panel in the column. The unit airfoil is scaled to "fit" the wing at the y_{cp} of each panel; then the camber and thickness are calculated.

USAGE:

COMMON (See subroutine OPCAMI for unlabeled COMMON description)

COMMON /COM1/ (See subroutine PANEL)

Input: KOPTF
NUMS
KSTART
KEND
NPER
NPER1
NPLANE
NPLN1

COMMON /COM2/ (See subroutine WING)

Input: NPTS
X
Y
Z
SLOPE
XCEN
XCON
YCON
XFOIL
ZFOIL
XNUM

Output: ZCEN
ZCON
THETA
ALPHA

CALL SLOPEW

SUBPROGRAMS CALLED:

ATAN (Built-in function)
BITURP
MEAN
POLXN
CHECK

ERROR RETURNS: Error messages indicate whether error occurred in use of subroutine BITURP to interpolate on airfoil surfaces or on camber and thickness curves. An error message also indicates if the airfoil x-ordinates are not defined as a (strictly) monotonic increasing sequence.

STORAGE: $2272_{10} = 4340_8$

SUBJECT: FORTRAN IV Subroutine TAN

(See subroutine FTAN)

SUBJECT: FORTRAN IV Subroutine TDUMP

PURPOSE: To read the binary tape produced by the Geometry Definition section and to write its contents on an output tape.

METHOD: A given logical array, which contains tape record status information, is checked. If there are no valid records on the binary tape, control is returned to the calling program. Otherwise, each of the 18 records is processed. The tape format is given in Appendix C.

USAGE:

```
LOGICAL LGDEF(3,6)
CALL TDUMP (LO, LTAPE, LGDEF)
```

Input:
- LO = Output tape number.
- LTAPE = Binary tape number.
- LGDEF = See subroutine GEOMD.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: Records must not contain more than 2000 words.

STORAGE: $2175_{10} = 4177_8$

SUBJECT: FORTRAN IV Subroutine TFLAT

PURPOSE: To transform body and wing to a new coordinate system, to flatten the wing, and to find the intersection of each wing percent line with the body surface.

METHOD: Three types of cases may be handled: body alone, wing alone, or wing-body combination. If a body surface is involved, it is assumed to be symmetrical about the X-Z plane; only half of it is given. Then a new x, y, z coordinate system is set up with origin at the centroid of forward end of the body and the x-axis passing through the aft-end centroid; the z-axis lies in the X-Z plane. In the new coordinate system, the body ends (unless of zero radius) are not necessarily parallel to the y-z plane; this is corrected by a linear adjustment of all x-coordinates of the body points. Also, the new body length (distance between body end centroids) may be different than the original length. The original set of body defining stations is converted to the new coordinate system by a transformation of the form $x = \alpha X + \beta$. The body radius and centroid are computed at each of these stations. In a body alone case, the radius and centroid are then found by interpolation at a given number of equally spaced stations; otherwise, this calculation is deferred.

In a wing-body combination, the wing is also transformed to the x, y, z coordinate system. Then the intersections of the wing leading and trailing edges with the body surface are computed; let z_A be the average of the z-coordinates of these two points. The wing is effectively flattened by changing each z-coordinate of the wing to z_A. The body is then intersected by each wing percent chord line of the flattened wing.

The body radius and centroid calculation can now be resumed. A set of equally spaced stations is chosen between the forward end of the body and the station at which the wing leading edge intersects the body; a similar set is selected from there on aft. The calculation of the interval between stations in each region is subject to the constraints that the total number of stations must agree with a given number and that the two intervals must be approximately the same in each region.

In a wing alone case, the transformation is omitted. However, the wing is flattened: z_A is taken

as the average of the z-coordinates of the first point in the leading- and trailing-edge percent lines.

The body and wing definitions are on a binary tape, LA, which has a format as described in Appendix C. The body is given by meridian lines, the wing by percent chord lines; meridian and percent lines are described by sets of three-dimensional points (see subroutine WBXUL). One data card is required; it is read with FORMAT (5F10.0) as follows:

Columns	Symbol	Explanation
1-10	XNRX	Number of body stations at which body radius and centroid are desired. If this number is less than 3 or greater than 50, it is set equal to 50.
11-20	CHD	A tolerance passed on to subroutine TFLAT1, and then to BODCR, for eventual use by subroutines ENRYCH and OPTIM3. CHD is used only to find body centroids, as follows: A section, normal to the x-axis, is taken through the body. The section has one point for each meridian line. If CHD $\leq$ 0, the centroid is found assuming the section to be polygonal. If CHD > 0 and there are at least four meridian lines, additional section points are generated by ENRYCH on a smooth curve through the vertices of the polygon: a sufficient number of points will be generated so that the difference between the chord of the resultant polygon and the arc of the smooth curve is (approximately) less than CHD. A positive value of CHD should not be used unless there are at least eight meridian lines on a non-symmetrical (about a horizontal plane) body.

Columns	Symbol	Explanation
21-30	BCODE (1)	An interpolation code used by BITURP to find body radius and z_c, (z-coordinate of the body centroid) at values of x between defining stations. BCODE (1) = 1 for linear interpolations, = 2 for biquadratic interpolation.
31-40	BCODE (2)	An interpolation code used by BITURP to find the intersections of wing percent lines with the body surface (see subroutine WBXX). This process involves finding body sections at various values of x. If BCODE (2) = 1, linear interpolation is used to find y as a function of z; this is equivalent to saying that the body is polygonal in cross section. If BCODE (2) = 2 (and there are at least three meridian lines), biquadratic interpolation is used; this is equivalent to specifying a body of curved cross section (but see RESTRICTIONS, under BITURP).
41-50	EPRC	A tolerance applied to body radius and z_c (z-coordinate of the body any value of x) is less than EPRC, the corresponding radius or z_c is set to zero. EPRC is introduced to help offset round-off errors.

Two binary tapes are written by TFLAT; their formats are described in Appendix C.

USAGE:

```
COMMON /LGEOMD/LGDEF(3,6)
LOGICAL LGDEF
DIMENSION DAT(2)
CALL TFLAT(DAT, LI, LO, LA, LB, LD, NBODY, IOK)
```

Input:	DAT	=	Date (alphameric).
	LI	=	Input tape number.
	LO	=	Output tape number.
	LA	=	Binary tape number; this tape is read by TFLAT (see Appendix C).
	LB	=	Binary tape number; three records are to be written on this tape, unless a wing-only case is processed (see Appendix C, section 3).
	LD	=	Binary tape number; six records are to be written on this tape (see Appendix C, section 2).
	LGDEF	=	Logical array, in COMMON, which indicates body and wing status. LGDEF(1, 3) = .TRUE. if a body definition was requested; LGDEF(3, 3) = .TRUE. if the body definition was successfully written on tape LA. LGDEF(1, 1) = .TRUE. if a wing definition was requested; LGDEF(3, 1) = .TRUE. if the wing definition was successfully written on tape LA. The other elements of LGDEF are not used by TFLAT.
Output:	NBODY	=	Code to indicate a successful body-alone case; then NBODY = -1. Otherwise, NBODY = 0.
	IOK	=	Error indicator, which is zero if success.

SUBPROGRAMS CALLED:

WBXX
TRAPCT
BITURP
TFLAT1
TFLATM
TFLATW
TFLATX
TRAV
FSR

ERROR RETURNS: If an error is detected, a message is written on the output tape. Then data cards on the input tape are skipped over until the next case, or end of data, is found. A new case is signaled by the word DEFINE or DEF1NE in columns 1-6 of a data card; end of data is signaled by the word END OF. The input tape is then backspaced one record.

The argument IOK is set to one of the following values if an error is found:

IOK	Explanation
2	A body or wing definition failed in a previous link.
4	Error reading body stations from tape LA. Either the number of stations is less than 2 or greater than 51, or something is wrong with the tape.
6	Error reading body meridian lines from tape LA. Something is wrong with the tape or record 4 has more than 4368 words.
8	Error from subroutine TFLAT1.
10	Error reading wing percent lines from tape LA. Record 7 is not correct.
12	Record 7 on tape LA indicates that record 8 has more than 4368 words.
14	Error from subroutine WBXX in finding wing percent line intersections with the body (before transformation).
16	Either the leading or trailing edge of the wing does not intersect the body.
24	Error from subroutine WBXX in finding the intersections of the flattened wing with the transformed body.
26	Error from subroutine TFLATX. One or more of the flattened wing percent lines did not intersect the transformed body.
32	Error from subroutine BITURP in interpolating points on the radius versus station or z-centroid versus station curve.

If IOK = 2, the message "ERROR IN PREVIOUS LINK" is written. If IOK > 2, the message:

"TFLAT ERROR i, CODE j
IREC = i_1 i_2 i_3 i_4"

is written, where i = IOK, j = error code from a subroutine (if IOK = 8, 14, 24, 26, or 32), and i_1, i_2, i_3, i_4 are the first four elements of the last odd-numbered record read from tape LA. If IOK = 14 or 16, the transformed wing percent lines are written by subroutine TFLATW to aid in diagnosing the error.

RESTRICTIONS: Only a single wing surface can be processed. If tape LA is written by the geometry definition program, a lower-wing case cannot be handled since TFLAT only examines the upper-wing portion of tape LA. The number of words in records 4 and 8 cannot exceed 4368.

STORAGE: $11957_{10} = 27265_8$

SUBJECT: FORTRAN IV Subroutine TFLAT1

PURPOSE: To transform body meridian lines to a new coordinate system, and to find the centroid and equivalent body radius at a set of body stations.

METHOD: This subroutine is used by subroutine TFLAT. The body centroid at each end of the body is found by subroutine BODCR. Then a new coordinate system is set up such that the origin is at the forward end centroid and the new x-axis passes through the aft-end centroid. After transforming the given body to the new coordinate system, the body radius and z-coordinate of the centroid is found at each given body station.

USAGE: DIMENSION STA(51), B(2448), S(100, 2), E(2000), RT(3, 4), ALFBET(2), RZ(51, 2) NU(3)

CALL TFLAT1 (CHD, NSTA, STA, NBPCT, B, S, E, RT, ALFBET, RZ, EPRC, NU)

Input:

CHD	=	A chord-arc tolerance, read from data card by subroutine TFLAT.
NSTA	=	Number of body stations.
STA	=	Array of original body stations (see Output, below).
NBPCT	=	Number of body meridian lines.
B	=	Array of points on original body meridian lines (see Output, below). This array (with a header) has a format as described in Appendix C.
EPRC	=	A tolerance, read from data card by subroutine TFLAT.
NU	=	Error indicator array. NU(1) is not used, on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, Then if NU(3) $>$ 0 and NU(2) $>$ 0, an error message is written.

Output:	STA	= Array of transformed body stations.
	B	= Array of points on transformed body meridian lines (with header).
	S	= Scratch array.
	E	= Scratch array.
	RT	= Transformation matrix (see TROTPT).
	ALFBET	= Transformation constants to convert original body stations to new stations (see TFLAT).
	RZ	= Array of body radii and z-centroid corresponding to each body station.
	NU	= Error indicator array (see Input) NU(1) = 0 for success.

SUBPROGRAMS CALLED:

UVECN
BSCALE
BODCR
VCROS
TRAPCT
TRAV

ERROR RETURNS: If an error is detected, the message "TFLAT ERROR i, CODE J, ..." is written.

i	j	Explanation
1	k	Error k from BODCR at forward body end.
2	k	Error k from BODCR at aft body end.
3	0	The original body length is zero.
4	k	Error k from BSCALE.
5	0	The transformed body length is zero.
6	k	Error k from BODCR at some transformed body station (an additional comment will identify the station).

RESTRICTIONS: It is assumed that the given body meridian lines describe only half of a body that is symmetrical about a plane parallel to the xz plane.

STORAGE: $381_{10} = 575_8$

SUBJECT: FORTRAN IV Subroutine TFLATM

PURPOSE: To write body meridian line points on an output tape.

METHOD: This is primarily an output routine for TFLAT. The meridian lines are given in a single array with a format such as that described in Appendix C. The polar coordinates of each point are computed in a plane normal to the x-axis, with polar origin at the body axis.

USAGE:

```
DIMENSION  B(1), AXIS(2), TITLE(12), DAYT(2), STA(1)
CALL TFLATM (B, NB, AXIS, TITLE, DAYT, LO, STA)
```

Input:	B	= Array of meridian lines, with header.
	NB	= Number of meridian lines.
	AXIS	= y, z coordinates of main body axis.
	TITLE	= Title (alphameric).
	DAYT	= Date (alphameric).
	LO	= Output tape number.
	STA	= Not used.
Output:		= Print-out on tape LO

SUBPROGRAMS CALLED: ATN1
SQRT (Built-in function)
TRAV

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $346_{10} = 532_8$

SUBJECT: FORTRAN IV Subroutine TFLATW

PURPOSE: To write wing percent lines on an output tape.

METHOD: This is an output subroutine used by subroutine TFLAT. The percent lines are given in a single array with a format as described in Appendix C.

USAGE:

```
DIMENSION  B(1), TITLE(12), DAYT(2)
CALL TFLATW (B, NB, TITLE, DAYT, LO)
```

Input:

B	=	Array of percent lines, with header.
NB	=	Number of percent lines.
TITLE	=	Title (alphameric).
DAYT	=	Date (alphameric).
LO	=	Output tape number.

SUBPROGRAM CALLED: TRAV

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $312_{10} = 470_8$

SUBJECT: FORTRAN IV Subroutine TFLATX

PURPOSE: To write wing-body intersection points on an output tape.

METHOD: This is an output subroutine for subroutine TFLAT (see subroutine TFLAT). The polar coordinates of each point are computed and written on the output tape along with the x, y, z coordinates.

USAGE:

```
DIMENSION  DAT(2), W(1), WB(3,NW)
CALL TFLATX (LO, DAT, NW, W, WB, NU)
```

Input:

- LO = Output tape number.
- DAT = Date (alphameric).
- NW = Number of wing percent lines (also number of wing-body intersection points).
- W = Array of wing percent lines, with header (see Appendix C).
- WB = Array of points (x_i, y_i, z_i) that are the intersections of wing percent lines with body surface.

Output:

- NU = Error indicator, which is zero for success.

SUBPROGRAMS CALLED:
ATN1
SQRT (Built-in function)
TRAV

ERROR RETURNS: NU = 1 if any intersection point is not valid (indicated by coordinates all 10^{30}).

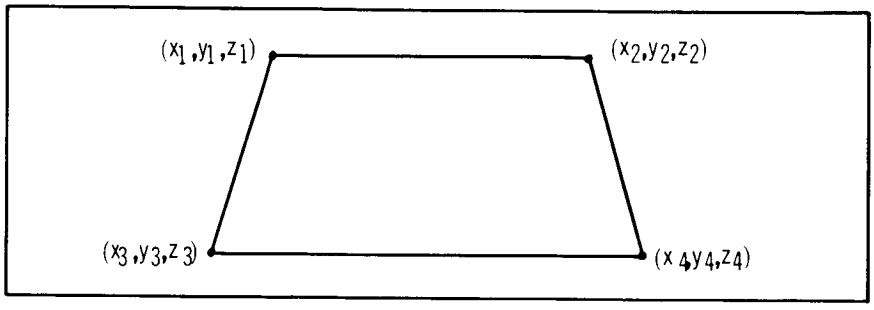
RESTRICTIONS: None

STORAGE: $240_{10} = 360_8$

SUBJECT: FORTRAN IV Subroutine THETAB

PURPOSE: To calculate body panel theta-inclination angles.

METHOD: The program calculates the panel theta-inclination angle by use of standard geometric formulae. The inclination angle θ is the angle of the panel in the x, y, z coordinate system of the body as viewed down the x-axis. A program check determines if the panel is "horizontal" by comparing the z-coordinates of the two trailing-edge panel corner points. If the panel is horizontal, theta is set equal to zero. For a nonhorizontal panel defined as follows,



theta is calculated from the formula,

$$\theta = \frac{|z_4 - z_3|}{(z_4 - z_3)} \cos^{-1}\left[\frac{A}{(A^2 + B^2)^{1/2}}\right]$$

for which,

$$A = \begin{vmatrix} x_1 & y_1 & 1 \\ x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \end{vmatrix}$$

$$B = - \begin{vmatrix} x_1 & z_1 & 1 \\ x_3 & z_3 & 1 \\ x_4 & z_4 & 1 \end{vmatrix}$$

USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description).

COMMON /COM1/ (See subroutine PANEL)

Input: NPER1
NPLN1

COMMON /COM2/ (See subroutine BODY)

Input: XCOR
YCOR
ZCOR

Output: THETA

CALL THETAB

SUBPROGRAMS CALLED: ACOS
SQRT (Built-in function)

ERROR RETURNS: None

STORAGE: $244_{10} = 364_8$

SUBJECT: FORTRAN IV Subroutine TRAPCT

PURPOSE: To perform a coordinate transformation on a set of body meridian lines or wing percent lines.

METHOD: Subroutine TROTPT is applied to transform each point, using a given transformation matrix.

USAGE:

```
DIMENSION  B(1), RT(3, 4)
CALL  TRAPCT (B, N, RT)
```

Input:
- B = Meridian or percent line array (see subroutines TFLAT and TFLAT1).
- N = Number of meridian or percent lines.
- RT = Transformation matrix (see subroutine TROTPT).

Output:
- B = Transformed array.

SUBPROGRAM CALLED: TROTPT

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $86_{10} = 126_8$

SUBJECT: FORTRAN Subroutine TRAV (TRAnsfer Vector)

PURPOSE: To move a block of N cells starting at location B to another region starting at location A.

METHOD:

```
        DO  100    I = 1, N
100     A(I)  =  B(I)
```

USAGE

```
CALL TRAV (A, B, N)
DIMENSION  A(N), B(N)
```

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: $N > 0$

STORAGE: $35_{10} = 43_8$

SUBJECT: FORTRAN IV Subroutine TROTPT

PURPOSE: Given the coordinates of a point (in 3-space) in one of two rectangular Cartesian coordinate systems and the rotation and translation matrices relating the systems, to compute the coordinates of the point in the other system.

METHOD: Let an arbitrary point have the coordinates $P = (X_1, Y_1, Z_1)$ in an $X_1\ Y_1\ Z_1$ coordinate system, and the coordinates $Q = (X_2, Y_2, Z_2)$ in an $X_2\ Y_2\ Z_2$ coordinate system. Let R be a 3-by-3 rotation matrix constructed as follows: the first column is a unit vector, in the $X_1\ Y_1\ Z_1$ system, along the $+X_2$ axis; similarly, the second column is a unit vector along the $+Y_2$ axis, and the third column is a unit vector along the $+Z_2$ axis. Let T be a 1-by-3 translation matrix that contains the $X_1\ Y_1\ Z_1$ coordinates of the origin of the $X_2\ Y_2\ Z_2$ system. Then, if P and Q are considered to be 1-by-3 matrices,

$$Q = (P - T)R$$

$$P = QR^{-1} + T$$

USAGE: DIMENSION R(3 3), T(3), P(3), Q(3)

CALL TROTPT (R, T, K, P, Q)

Input: R = Rotation matrix.

T = Translation matrix.

K = A code that is 1 if Q is input and P is output, or 2 if P is input and Q is output.

P or Q = Point coordinates.

Output: Q or P = Point coordinates.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: The rotation matrix must be orthogonal. The same array may be used for both P and Q; e.g., CALL TROTPT(R, T, 1, Q, Q).

STORAGE: $112_{10} = 160_8$

SUBJECT: FORTRAN IV Subroutine TVEL

PURPOSE: To compute the velocity components resulting from a specified wing thickness distribution from the velocity components resulting from a wing thickness distribution of unit incidence.

METHOD: The velocity components in the x, y, and z directions on panel i resulting from a specified wing thickness distribution of incidence α_{T_j} are found by

$$U_i = \sum_{j=1}^{NWING} U_{ij} \cdot \alpha_{T_j}$$

$$V_i = \sum_{j=1}^{NWING} V_{ij} \cdot \alpha_{T_j}$$

$$W_i = \sum_{j=1}^{NWING} W_{ij} \cdot \alpha_{T_j}$$

where U, V, and W = The velocity components resulting from a from a wing thickness distribution of unit incidence.

NWING = The total number of wing panels.

The normal component of velocity on a body panel i resulting from wing thickness is given by

$$n_{B_i} = \sum_{j=1}^{NWING} A_{BWS_{ij}} \cdot \alpha_{T_j}$$

where A_{BWS} = The normal velocity on the body resulting from wing thickness distribution of unit incidence.

USAGE: CALL TVEL

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

/BLOCK/ X(200, 3, 4), Y(200, 3,4), Z(200, 3, 4), XBAR(200), YBAR(200), ZBAR(200), XC(200), YC(200), ZC(200), THETA(200), ALPHAS(200), AREA(200), CHORD(200), NPART(200), NPANEL, ISYM, A(200), U(200), V(200), VPM(200), VV(200), W(200), WPM(200), WW(200), VPMM(200) WPMM(200), ALPHAC(100), ALPHAT(100)

DIMENSION B(200), C(200), D(200), AN(200)

EQUIVALENCE: (VV, B), (WPM, C), (WW, D), (VPMM, AN)

Input:

NBODY = Number of body panels.

NWING = Number of wing panels.

NTAPEA = Logical number of the tape from which the normal velocities resulting from a wing thickness distribution of unit incidence are read.

NTAPEB = Logical number of the tape from which the components of velocity in the x, y, and z directions resulting from a wing thickness distribution of unit incidence are read.

NTAPEC = Logical number of tape on which the velocity components resulting from a specified wing thickness distribution are written.

ALPHAT = Array of wing thickness incidences.

U_{ij} = Matrix of velocity components in the x-direction at panel i resulting from a unit thickness incidence of panel j.

V_{ij} = Matrix of velocity components in the y-direction at panel i resulting from a unit thickness incidence of panel j.

W_{ij} = Matrix of velocity components in the z-direction at panel i resulting from a unit thickness incidence of panel j.

$A_{BWS_{ij}}$ = Matrix of normal velocity components at panel i resulting from a unit thickness incidence of panel j.

A = Dummy array used by the program, represents U_{ij}.

	B	= Dummy array used by the program, represents V_{ij}.
	C	= Dummy array used by the program, represents W_{ij}.
	D	= Dummy array used by the program, represents $A_{BWS_{ij}}$.
Output:	U	= Array of velocity components in the x-direction resulting from a given wing thickness distribution.
	V	= Array of velocity components in the y-direction resulting from a given wing thickness distribution.
	W	= Array of velocity components in the z-direction resulting from a given wing thickness distribution.
	AN	= Array of normal velocity components on the body resulting from a given wing thickness distribution.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: NWING $\leq$ 100

NBODY $\leq$ 100

STORAGE: $235_{10} = 353_8$

SUBJECT: FORTRAN Function UVECN

PURPOSE: Given two points A and B in N-space, to find components of vector (or unit vector) directed from A to B and distance between A and B.

METHOD: Vector $\overline{V} = \overline{B} - \overline{A}$ and sum $S = V_1^2 + V_2^2 + \ldots + V_n^2$ are formed. If $S > 0$, then distance $D = \sqrt{S}$; if a given code indicates that a unit vector is required, then $\overline{V} = \overline{V}/D$. Function value UVECN is set to D before return. If $S = 0$, then $\overline{V} = \overline{0}$ and $D = 0$.

USAGE:

```
D = UVECN (A, B, V, K, N)
DIMENSION  A(N), B(N), V(N)
```

Input: A, B = Point coordinates

K = If K = 0, then a unit vector $V = (\overline{B} - \overline{A})/|(\overline{B} - \overline{A})|$ is desired; If $K \neq 0$, then $\overline{V} = \overline{B} - \overline{A}$.

N = Dimension of the space in which A and B are defined.

Output: V = Vector $\overline{V}$

D = Distance from A to B

SUBPROGRAM CALLED: SQRT (Built-in function)

ERROR RETURNS: None

RESTRICTIONS: $N > 0$

STORAGE: $80_{10} = 120_8$

SUBJECT: FORTRAN IV Subroutine VCROS

PURPOSE: To find cross-product of two vectors.

METHOD: Given vectors $\overline{A} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$ and $\overline{B} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$, then cross-product,

A x B, is vector

$$\overline{V} = \begin{pmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{pmatrix}$$

USAGE:

```
DIMENSION  A(3),  B(3),  V(3)
CALL VCROS (A,  B,  V,  D,  K)
```

Input: $A = \overline{A}$
$B = \overline{B}$
$K = 0$ if $\overline{V}$ is not to be normalized
$\neq 0$ if $\overline{V}$ is to be normalized

Output: $V = \overline{A} \times \overline{B}$ if K was 0
$= (\overline{A} \times \overline{B})/D$ if K was not 0 and $D > 0$.
D = Magnitude of V (before normalization)
$K = 0$ if success
$= 1$ if $K \neq 0$ on input and $D = 0$.

SUBPROGRAM CALLED:
VDOTP
SQRT (Built-in function)

RESTRICTIONS: Note that K is both an input and an output.

STORAGE: $106_{10} = 152_8$

SUBJECT: FORTRAN Function VDOTN

PURPOSE: To find dot-product of two vectors in N-space.

METHOD: Standard

USAGE: DP = VDOTN (A, B, N)

DIMENSION A(N), B(N)

Input: A = Vector of dimension N

B = Vector of dimension N

Output: $DP = \overline{A} \cdot \overline{B} = \sum_{i=1}^{N} A_i B_i$

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: $N > 0$

STORAGE: $40_{10} = 50_8$

SUBJECT: FORTRAN IV Function VDOTP

PURPOSE: To find dot-product of two vectors.

METHOD: Given two vectors $\overline{A} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$ and $\overline{B} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$, dot-product is the scalar $P = a_1 b_1 + a_2 b_2 + a_3 b_3$.

USAGE:

```
DIMENSION  A(3), B(3)
P = VDOTP (A, B)
```

Input: A = Vector $\overline{A}$

B = Vector $\overline{B}$

Output: $P = A \cdot B$

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $43_{10} = 53_8$

SUBJECT: FORTRAN IV Subroutine WBX

PURPOSE: To find the intersections of a set of wing percent chord lines with a body surface and to delete the portions of percent chord lines inside the body.

METHOD: The body surface is defined by an array of NB meridian lines and a code that determines body shape between meridian lines (see subroutine WBXUL). Each wing percent chord line consists of a series of straight-line segments connecting three-dimensional points. The intersection of each wing percent chord line with the body surface is then calculated.

The body is cut (see subroutine WBXC) by a transverse plane that passes through the first wing percent chord point. The horizontal distance normal to the x-axis from the percent point to the body surface is found in this region. This calculation is repeated for successive percent chord line points until a change in sign of the distance is detected; this means that the last two wing points, W_1 and W_2 for example, are on opposite sides of the body surface. An iterative procedure is now used to locate the intersection point.

Let α be the distance along the line segment $\overline{W_1 W_2}$ from W_1 to some point W on the segment. Let $f(\alpha)$ be the horizontal distance from W to the body surface. If L is the length of $\overline{W_1 W_2}$, then f(0) and f(L) have opposite signs (because f(0) corresponds to W_1 and f(L) corresponds to W_2). Subroutine FROOTA is used to iterate on α until $f(\alpha)$ is as small as possible. The point W is then on or very near the body surface; the corresponding point on the body surface is taken as the intersection.

If the first wing percent chord line point is forward or aft of the body, the transverse plane through the point does not intersect the body; WBX then continues to process points until a body intersection is found. If all points so far processed are on the same side of the body surface, and the current point is forward or aft of the body, then the previous point is taken as W_1; W_2 is found by subroutine WBXD as the intersection of the percent chord line with a transverse plane through the appropriate body end. If all percent chord lines are processed without finding two adjacent points on opposite sides of the body surface, the intersection point is taken as $(10^{30}, 10^{30}, 10^{30})$ to indicate that no intersection point was found.

USAGE:

DIMENSION B(1), W(1), S(2, NB), BW(3, NW), NU(3)

CALL WBX (B, NB, W, NW, EPS, KODE, S, BW, NU)

Input:

B	=	Array of body meridian lines (described in subroutine WBXUL).
NB	=	Number of meridian lines.
W	=	Array of wing percent chord lines (described in subroutine WBXUL).
NW	=	Number of percent chord lines.
EPS	=	Tolerance — a message will be written on the output tape (see NU) if the error in finding an intersection is greater than EPS.
KODE	=	A code that determines the shape of the body surface between meridian lines (see subroutine WBXUL).
NU	=	Error indicator array: NU(1) is not used on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) ≤ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written.

Scratch:

S	=	Storage for sections through the body.

Output:

W	=	Array of wing percent chord lines, truncated at the body surface.
BW	=	Array of intersection points (x_i, y_i, z_i).
NU	=	Error indicator array: NU(1) is zero if no error is detected. NU(3) may have been changed (see INPUT, above).

SUBPROGRAMS CALLED:

UVECN
FROOTA
TRAV
WBXA
WBXC
WBXD

ERROR RETURNS: The following message is written (see NU) if an error is detected: "ERROR i, CODE j IN SUBROUTINE WBX, WING PERCENT LINE m, POINT n."

i	j	Explanation
4	1	WBXA error; see note 1, below.
5	2	WBXD error; see message written by WBXD.
6	-3	WBXC error; first and last points in a body section (through a wing percent line point) have the same z-coordinate.
7	3	WBXC error; two points in body section, near the wing-body intersection, have the same z-coordinate
8	1 2	A wing percent chord line point is above or below the body section and has the same y-coordinate as the first or last section point.
9	-	See note 1.
10	k	WBXC error k in iteration loop.
11	k	FROOTA error; see note 1.
12	0	Iteration to find intersection did not converge; see note 1.

Note 1: Machine or program error.

RESTRICTIONS: If a wing percent chord line intersects the body surface more than once, only one intersection point will be found. Body sections in the wing-body intersection region must be single-valued in y as a function of z. Wing percent chord lines must have at least one point that lies between body station planes through the body ends. A percent chord line intersection will not be found, even though it exists, if all percent chord points except the last are forward (or aft) of the body.

STORAGE: $605_{10} = 1135_8$

SUBJECT: FORTRAN IV Subroutine WBXA

PURPOSE: To modify a wing percent chord line array so that the first part of a given percent chord line is removed.

METHOD: The format of the wing percent chord line array, B, is explained in Appendix C. K, the number of points to be removed, and P, the new first-point, are given. If K = 0, P is not used and the entire old percent chord line is moved to its new position. Otherwise, K points are removed and P, followed by the remaining points, is moved. The corresponding portion of the array header is adjusted to reflect the new location of the percent chord line in the array and the new number of points. The remaining percent chord lines are not disturbed.

USAGE:

```
DIMENSION   B(1), P(3)
CALL WBXA (B, I, K, P, NU)
```

Input: B = Percent chord line array.

I = The percent chord line to be modified.

K = Number of points to be removed.

P = New first-point.

Output: B = Modified percent chord line array.

NU = Error indicator, which is zero if successful.

SUBPROGRAM CALLED: None

ERROR RETURNS: NU = 1 if $K < 0$ or if $K \geqslant$ number of points in old percent chord line.

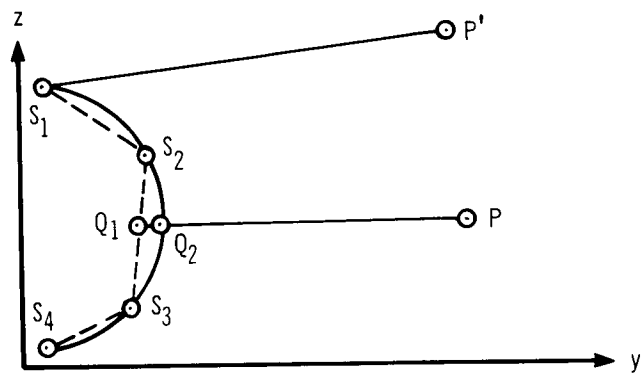
RESTRICTIONS: WBXA must be called successively with I = 1, 2, ..., N where N = number of percent chord lines.

STORAGE: $202_{10} = 312_8$

SUBJECT: FORTRAN IV Subroutine WBXC

PURPOSE: To find the distance from an arbitrary point to a body surface.

METHOD: The body is described by NB meridian lines (see subroutine WBXUL). Let $P(x_P, y_P, z_P)$ be the given point. Subroutine BCUTX is used to find the intersection points (y_i, z_i) of the plane $x = x_P$ with the meridian lines. The points form a body section, S.



Let $Q(x_Q, y_Q, z_Q)$ be a point on the body so that $x_Q = x_P$, $z_Q = z_P$. The value of y_Q is found by interpolation at $z = z_Q$ from the table (z_i, y_i), using subroutine BITURP. If a given code is 1, the interpolation is linear (Q_1 in figure); if the code is 2, the interpolation is biquadratic (Q_2 in figure). This code effectively determines the body shape between meridian lines. If P should be above the body (see P' in drawing), then Q is taken as the first meridian line point. Similarly, the last meridian line point is used for Q if P is below the body. The required distance from P to the body surface is $y_P - y_Q$.

USAGE: DIMENSION B(1), P(3), S(2, NB), Q(3)

CALL WBXC (B, NB, EP, P, KODE, S, D, Q, NU)

Input: B = Array of body meridian line points, with header (see subroutine WBXUL).

	NB	= Number of body meridian lines.
	EP	= Tolerance used by BCUTX in finding intersections of body meridian lines with the plane x = P(1).
	P	= x, y, z coordinates of given point.
	KODE	= Code, used by BITURP, that determines the shape of the body between meridian lines. KOD = 1 if polygonal, 2 if not (see METHOD, above).
Output:	S	= Section points (y_i, z_i), i = 1, 2, . . , NS, where NS $\leq$ NB.
	D	= Distance.
	Q	= Point on body surface.
	NU	= Error indicator array. NU = 0, 1 or 2 if successful; 0 if Q(3) = P(3), 1 if S_1 was used for Q, 2 if S_{NB} was used for Q.

SUBPROGRAMS CALLED: BITURP
BCUTX

ERROR RETURNS: NU = -5 if the number of section points is less than 2; this may be caused by NB < 2 or failure of the plane $x = x_P$ to cut some of the meridian lines. NU = -3 if S_1 and S_{NB} have the same z-coordinate.

RESTRICTIONS: The value of D returned is the horizontal distance between P and Q; if P is above or below the body, the true distance is greater than D. If all meridian lines do not have the same x-range, the number of section points (S) may possibly be less than NB; see BCUTX.

STORAGE: $139_{10} = 213_8$

SUBJECT: FORTRAN IV Subroutine WBXD

PURPOSE: To find a point on a wing percent chord line which lies in the first or last body station plane, and to determine the horizontal distance of that point from the body surface.

METHOD: Subroutine WBXD is called by subroutine WBX only if the following situation arises: At least one point, W_1, for instance, on some wing percent chord line has been processed; but the next point, W_2, is either forward or aft of the body. Let BXF and BXL be the extreme x-coordinates of body meridian line 1; let WXF and WXL be the x-coordinates of W_1 and W_2, respectively. If BXF lies in the range (WXF, WXL), a plane x = BXF is formed; otherwise BXL is assumed to be the range and the plane is taken as x = BXL. The line segment $\overline{W_1W_2}$ is intersected with the plane to obtain point P, which lies on the wing percent line. The horizontal distance from P to the body, and the corresponding point Q on the body are then found by subroutine WBXC.

USAGE: DIMENSION R(4), B(1), P(3), Q(3), S(2, NB), W(3, 2), NU(3)

Call WBXD (R, B, NB, KODE, P, Q, S, W, NU)

Input:

- B = Array of body meridian line points, with header (see subroutine WBXUL).
- NB = Number of body meridian lines.
- KODE = Body shape code (see WBXC).
- W = Two adjacent points on a wing percent chord line (see W_1 and W_2, METHOD, above).
- NU = Error indicator array. NU(1) is not used, on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written.

Output:

- S = Section points (Y_i, Z_i).
- R = R(1) = x-coordinate of P; R(2) = horizontal distance = P(2) - Q(2); R(3) = R(1); R(4)

P	=	x, y, z coordinates of point on wing percent chord line.
Q	=	Corresponding point on body surface.
NU	=	Error indicator array. NU(1) is zero if success. NU(3) may have changed if an error was detected (see Input).

SUBPROGRAMS CALLED: POLXN
WBXC

ERROR RETURNS: A message "ERROR i, CODE j, IN SUBROUTINE WBXD" is written if an error is detected.

i	j	Explanation
1	0	Machine or program error.
2	k	Error k from WBXC. If k = 1 or 2, point P is above or below the body and P(2) = Q(2).

RESTRICTIONS: See WBXC

STORAGE: $219_{10} = 333_8$

SUBJECT: FORTRAN IV Subroutine WBXUL

PURPOSE: To find the intersections of upper and lower wing percent chord lines with a body surface, to delete the portions of percent chord lines that are inside the body, and to write results.

METHOD: A set of NB meridian lines defines the basic body surface. These lines may be visualized as stringers running lengthwise along the surface of the body. Each meridian line consists of a series of straight-line segments connecting three-dimensional points. The surface between meridian lines is determined by taking any section, normal to the X-axis, through the body; this yields the (Y, Z) coordinates of NB points on the surface. Under one option, the surface at this section is assumed to be polygonal (adjacent points connected by straight lines). In the other option, the surface between each pair of points is assumed to be represented by a cubic; points between meridian lines are interpolated by subroutine BITURP.

The wing surface (upper, lower, or both) is defined by NW wing percent chord lines, which are similar to meridian lines. The 0-percent line forms the leading edge of the wing, and the 100-percent line is the trailing edge. The wing surface is not defined between percent chord lines. The main function of WBXUL is to find the point of intersection of each wing percent chord line with the body surface.

All of the body meridian line points are stored in a single array. The first 3*NB cells form a "header" that is a key to the rest of the array. Cell 1 is a number that forms a label for the first meridian line. Cell 2 contains the number of points in the first meridian line. Cell 3 contains the cell number at which the x-coordinate of the first point in the first meridian line is located. Similarly, cells 4, 5, and 6 apply to the second meridian line, etc. This storage scheme, described in Appendix C, is also used for the upper wing percent lines and the lower percent lines.

The subroutine begins by a check that the body and at least one wing surface are defined. Then one card is read with FORMAT(2F10.0). The first field contains a code, later used by BITURP, that determines body shape between

meridian lines (1 if polygonal, 2 if not). The second field contains a tolerance, EPS, that is set to 10^{-5} if less than 10^{-5}; subroutine WBX writes a message if the error in locating an intersection point is more than EPS. The following process is then carried out, for the upper wing surface and then the lower.

If the wing surface is defined, storage is reserved in the buffer for the intersection points (see Appendix A). Subroutine WBX is called to find the intersection points and remove that part of each wing percent chord line inside the body surface. Each wing percent chord line now starts at the body surface; this, of course, requires a new "header" for the array. The X, Y, Z and polar coordinates of each intersection point are written on the output tape. The polar coordinate system for each point lies in a plane through the point and normal to the X-axis, with origin at the known body axis; the angle is measured from the vertical (Z-axis). If any percent line fails to intersect the body surface, a message is written but the routine continues.

USAGE:

```
DIMENSION  B(1), DAT(2), LOK(3, 9), BTITLE(12),
WTITLE(12), BAXIS(2), NU(3)

LOGICAL  LGDEF(3, 6)

CALL WBXUL(B, LI, LO, DAT, LGDEF, LOK, BTITLE,
WTITLE, BAXIS, NU)
```

Input:	B	= Storage buffer for variable length arrays.
	LI	= Input tape number.
	LO	= Output tape number.
	DAT	= Date (alphameric).
	LGDEF(2, 1)	= .TRUE. if upper wing is defined.
	(2, 2)	= .TRUE. if lower wing is defined.
	(2, 3)	= .TRUE. if body is defined.
	LOK	= Array that tells where in B the wing percent chord line arrays and body meridian line array are located, and the length of each array. Consider LOK(i, j): The upper wing percent chord line corresponds to

LOK(i, 1), the lower to LOK(i, 2); the body corresponds to LOK(i, 3). An array starts in B(LOK(1, j)). LOK(2, j) = number of percent chord lines or meridian lines. LOK(3, j) = total number of elements in array.

BTITLE = Body title (alphameric).

WTITLE = Wing title (alphameric).

BAXIS = Y and Z coordinates of the body axis.

NU = Error indicator array: NU(1) is not used, on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) =(NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written.

Output: B = Storage buffer for variable-length arrays.

LGDEF(1, 5) = Set equal to LGDEF(1, 1) to indicate that a wing-body intersection was requested.

(1, 6) = Set equal to LGDEF(1, 2).

(2, 5) = Set to .TRUE. if intersections of the upper wing percent chord lines with the body were found.

(2, 6) = Set to .TRUE. if intersections of the lower wing percent chord lines with the body were found.

LOK = Array that tells where in B various arrays are stored (see Input). The truncated upper wing percent chord lines (part outside body) correspond to j = 3, the lower to j = 4; the upper intersection point array corresponds to j = 8, the lower to j = 9. An array starts in

= B(LOK(1, j)). LOK(3, j) = total number of elements in array. LOK(2, j) = number of percent lines if j = 3 or 4. LOK(2, j) is not used for j = 8 or 9.

NU = Error indicator array. NU(1) is zero if success. NU(3) may have been changed if an error was detected; see Input, above.

SUBPROGRAMS CALLED:

ATN1
WBX
MERR
SQRT (Built-in function)
TRAV
IRSERV } (see Appendix A)
IPACK }

ERROR RETURNS: Function MERR is used to write a message "ERROR i, CODE j IN SUBROUTINE WBXUL DURING GEOMETRIC DEFINITION" if an error is detected.

i	j	Explanation
1	0	Either the body or wing was not successfully defined. However, if both upper and lower wing surfaces were requested but one surface failed, this error will not stop WBXUL (provided the body is defined).
2	0	The storage buffer, B, is not large enough or, a storage error has been detected.
3	0	
4	k	Error k from WBX.

RESTRICTIONS: The storage buffer, B, must have been initialized by subroutine INIBFR (see Appendix A) or its equivalent. The original wing percent lines (stored starting in B(LOK(1, i)), i = 1 and 2) are destroyed.

STORAGE: $758_{10} = 1366_8$

SUBJECT: FORTRAN IV Subroutine WBXX

PURPOSE: To find the intersections of a set of wing percent chord lines with a body surface.

METHOD: The body surface is defined by an array of NB meridian lines and a code that determines body shape between meridian lines (see subroutine WBXUL). Each wing percent line consists of a series of straight-line segments connecting three-dimensional points. The intersection of each wing percent chord line with the body surface is found in turn.

The body is cut (see subroutine WBXC) by a transverse plane that passes through the first wing percent chord line point. In this section, normal to the X-axis, the horizontal distance (with sign) from the percent point to the body surface is found. This process is repeated for successive percent chord line points until a change in sign of the distance is detected; this means that the last two wing points, W_1 and W_2, for example, are on opposite sides of the body surface. An iterative procedure is now used to locate the intersection point.

Let α be the distance along the line segment $\overline{W_1W_2}$ from W_1 to some point W on the segment. Let $f(\alpha)$ be the horizontal distance from W to the body surface. If L is the length of $\overline{W_1W_2}$, then f(0) and f(L) have opposite signs (because f(0) corresponds to W_1 and f(L) corresponds to W_2). Subroutine FROOTA is used to iterate on α until $f(\alpha)$ is as small as possible. The point W is then on or very near the body surface; the corresponding point on the body surface is taken as the intersection.

If the first wing percent chord line point is forward or aft of the body, the transverse plane through the point does not intersect the body; WBXX then continues to process points until a body intersection is found. If all points so far processed are on the same side of the body surface, and the current point is forward or aft of the body, then the previous point is taken as W_1; W_2 is found by subroutine WBXD as the intersection of the percent chord line with a body station plane through the appropriate body end. If all percent chord lines are processed without finding two adjacent points on opposite sides of the body surface, the intersection point is taken as (10^{30}, 10^{30}, 10^{30}) to indicate that no intersection point was found.

USAGE:

DIMENSION B(1), W(1), S(2, NB), BW(3, NW), NU(3)

CALL WBXX (B, NB, W, NW, EPS, KODE, S, BW, NU)

Input:

B = Array of body meridian lines (described in subroutine WBXUL).

NB = Number of meridian lines.

W = Array of wing percent chord lines (described in subroutine WBXUL).

NW = Number of percent chord lines.

EPS = Tolerance; a message will be written on the output tape (see NU) if the error in finding an intersection is greater then EPS.

KODE = A code that determines the shape of the body surface between meridian lines (see subroutine WBXUL).

NU = Error indicator array. NU(1) is not used, on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) $>$ 0 and NU(2) $>$ 0, an error message is written.

Scratch: S = Storage for sections through the body.

Output:

BW = Array of intersection points (x_i, y_i, z_i).

NU = Error indicator array. NU(1) is zero if no error is detected. NU(3) may have been changed (see Input, above).

SUBPROGRAMS CALLED:

UVECN
FROOTA
TRAV
WBXC
WBXD

ERROR RETURNS: The following message is written (see NU) if an error is detected: "ERROR i, CODE j IN SUBROUTINE WBXX, WING PERCENT LINE m, POINT n."

i	j	Explanation
5	2	WBXD error; see message written by WBXD.
6	-3	WBXC error; first and last points in a body section (through a wing percent line point) have the same z-coordinate.
7	3	WBXC error; two points in body section, near the wing-body intersection, have the same Z-coordinate
8	1 2	A wing percent line point is above or below the body section and has the same Y-coordinate as the first or last section point.
9	-	See note 1, below.
10	k	WBXC error k in iteration loop.
11	k	FROOTA error; see note 1.
12	0	Iteration to find intersection did not converge; see note 1.

Note 1: Machine error or program error.

RESTRICTIONS: If a wing percent chord line intersects the body surface more than once, only one intersection point will be found. Body sections in the wing-body intersection region must be single-valued in y as a function of z. Wing percent chord lines must have at least one point that lies between body station planes through the body ends. A percent chord line intersection will not be found, even though it exists, if all percent points except the last are forward (or aft) of the body.

STORAGE: $584_{10} = 1110_8$

SUBJECT: FORTRAN IV Subroutine WING

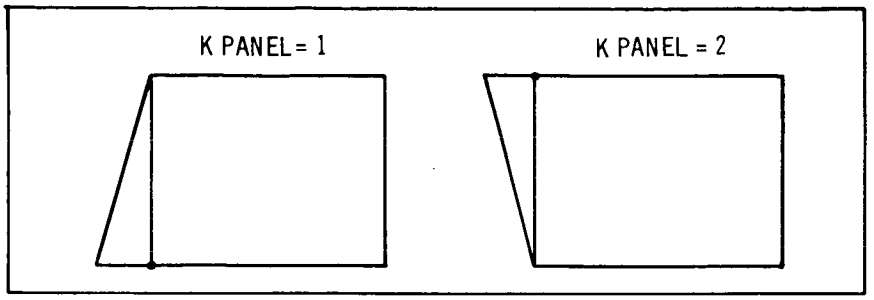
PURPOSE: To serve as control program for wing paneling subroutines.

METHOD: Program uses a series of FORTRAN IV language "CALL" instructions to call required wing paneling subroutines.

The program has the following labeled COMMON statement that is also in all lower level wing geometry paneling subroutines,

COMMON /COM2/ IJ, NPTS(16), X(90, 16), Y(90, 16), Z(90, 16), KPNT, VALUE(5), YCEPT(16), SLOPE, KPANEL(2, 15), XCOR(16, 16), YCOR(16, 16), ZCOR(16, 16) XINT(2, 15), YINT(2, 15), ZINT(2, 15), XCEN(15, 15), YCEN(15, 15), ZCEN(15, 15), XCON(15, 15), YCON(15, 15), ZCON(15, 15), AREA(15, 15), ARAT(15, 15), THETA(15, 15), ALPHA(15, 15), CHORD(15, 15), XFOIL(16, 25, 2), ZFOIL(16, 25, 2), XNUM(16, 2), XTAB1(25), ZTAB1(25), XTAB2(25), ZTAB2(25)

IJ	= Code for internal program control.
NPTS	= Number of points on the successive wing percent chord lines that define the wing.
X, Y, Z	= x, y, z coordinates of points on wing percent chord lines.
KPNT	= Input constant
VALUE	= Input array.
YCEPT	= y-intercepts of wing cutting planes.
SLOPE	= Slope (dx/dy) of outboard wing edge.
KPANEL	= Code for wing panel type. If KPANEL (I, J) = 1, additional subpanel corner point is located on trailing edge. If KPANEL(I, J) = 2, point is located on leading edge. Note this code is determined <u>only</u> for panels on inboard column <u>if</u> body and wing intersect in nonstreamwise line, <u>and</u> for panels on outboard edge <u>if</u> outboard edge is a nonstreamwise line.



XCOR YCOR ZCOR	= x, y, z coordinates of wing panel corner points.
XINT YINT ZINT	= x, y, z coordinates of additional wing panel point.
XCEN YCEN	= x and y coordinates of wing panel centroid.
ZCEN:	= Wing panel thickness ordinate.
XCON YCON	= x and y coordinates of wing panel control point.
ZCON	= Wing panel camber ordinate.
AREA	= Wing panel area.
ARAT	= Ratio of subpanel area to panel area.
THETA	= Wing panel thickness slope.
ALPHA	= Wing panel camber slope.
XFOIL ZFOIL XNUM	= Input arrays.
XTAB1 XTAB2 ZTAB1 ZTAB2	= Temporary storage arrays used in subroutine slope.

USAGE: COMMON (See subroutine OPCAMI for description of unlabeled COMMON)

CALL WING

SUBPROGRAMS CALLED:

INPUTW
CRNRW
AREAP
CENTRD
CNTRLW
CHORDW
SLOPEW
OUTPTW

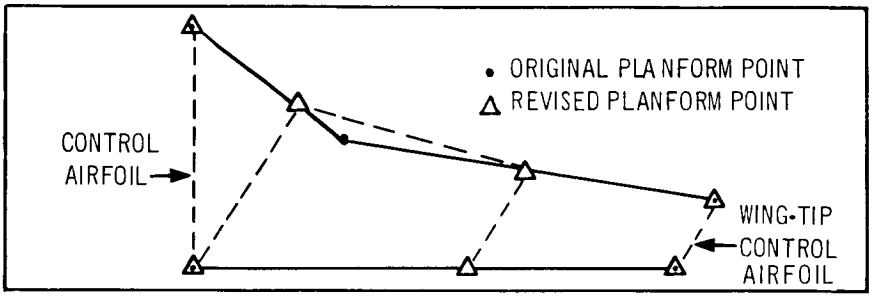
ERROR RETURNS: Error message indicates whether error occurred in calculation of wing panel geometry.

STORAGE: $220_{10} = 334_8$

SUBJECT: FORTRAN IV Subroutine WING1

PURPOSE: To read data cards that describe a wing by a planform and a set of control airfoils, and put this data in a standard form suitable for other subroutines.

METHOD: All of the data cards described in the attachment to subroutine WINGA are read. After the first three cards are read and checked, appropriate information is written on the output tape. Then storage is reserved in a buffer for variable length and scratch arrays (see Appendix A). Points describing the planform are read from cards 4 and 5. Subroutine WING1A is called to read cards 6, 7, and 8, scale the control airfoils to fit on the planform, and find the leading and trailing edge points corresponding to each airfoil. The original planform is assumed to be straight between given points; a new planform is constructed by joining the airfoil leading edge points and the trailing edge points. These planforms may differ as shown below.



After releasing scratch storage from the buffer, B, the percentages at which wing percent chord lines are to be constructed are read and checked. If an error is detected, a message is written.

USAGE:

```
DIMENSION   B(1), DAT(2), KOD(2), TITLE(12), NU(3)
LOGICAL   LOGUL(2)
CALL WING1 (B, LI, LO, DAT, LLETE, LOGUL, NAF,
            NPL, LAFN, LPCT, KOD, CHD, EPS,
            TITLE, NU)
```

Input: LI = Input tape number.

LO = Output tape number.

DAT = Date (alphameric).

NU = Error indicator array: NU(1) is not used on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written.

Output: B = Storage buffer for variable-length arrays.

LLETE = Array containing the intersection points (X_i, Y_i) of the control airfoil with the wing leading and trailing edges is stored starting in B(LLETE). The point corresponding to the first airfoil, leading edge, is stored first, followed by the point for the trailing edge. The intersection points corresponding to the remaining airfoils are similarly stored. This array corresponds to the PT array in subroutine WING1A.

LOGUL = LOGUL(1) = .TRUE. if the upper wing surface is defined; LOGUL(2) = .TRUE. if the lower wing surface is defined.

NAF = Number of control airfoils.

NPL = Number of wing percent chord lines.

LAFN = An array, starting in B(LAFN) tells where, in B, the control airfoil points are stored and the number of elements (twice the number of points) in the airfoil upper or lower surface. This array is the same as array NAME in subroutine WING1A. The elements in this array are type INTERGER, although B is type REAL. The first four elements apply to the first airfoil, the second four elements to the second foil, etc.

LPCT	=	Array of percentages (to specify set of wing percent chord lines) is stored starting in B(LPCT).
KOD	=	See ACODE and PCODE, card 3 (described in subroutine WINGA).
CHD	=	See CHD, card 2.
EPS	=	See EPS, card 3.
TITLE	=	See TITLE, card 1.
NU	=	Error indicator array. NU(1) is zero if no errors are detected. NU(3) may have been changed if an error was found (see Input, above).

SUBPROGRAMS CALLED:

WING1A
MERR
IDLETE
IRSERV
IPACK
} (See Appendix A)

ERROR RETURNS: Function MERR is used to write an error message "ERROR i, CODE j IN SUBROUTINE WING1 DURING GEOMETRIC DEFINITION" if an error is detected.

i	j	Explanation
1	0	WUL (see card 2) is not 0., 1., or 3.
2	0	PNLE, PNTE, AFN, or PLN (see card 2) is less than 2 or greater than 150. However, this is not considered a fatal error.
3	0	The storage buffer, B, is not large enough or a storage error has been detected.
5	0	
8	0	
9	0	
7	k	Error code k from subroutine WING1A.
10	0	A percentage (see card 9) is less than zero or greater than 100.

RESTRICTIONS: None

STORAGE: $802_{10} = 1442_8$

SUBJECT: FORTRAN IV Subroutine WING1A

PURPOSE: To read data cards that define a set of control airfoils, to compute points on the airfoils (scaled to fit planform), and to find the intersections of each airfoil with the planform.

METHOD: The wing planform is given by a set of points (X_i, Y_i) on the leading edge and another set on the trailing edge; adjacent points of each set are considered to be connected by straight lines. A card is read (card 6, see WINGA) that determines the placement of the airfoil on the planform and the number of points on the upper and lower surface of the airfoil. After these data are written, the intersections of the control airfoil and the planform leading and trailing edges are found. The airfoil chord length is the distance between these intersections.

A given code determines whether the upper wing, the lower wing, or both, are defined. Depending on this code, subroutine WING1B is called once or twice to read cards 7 and/or 8. These cards contain points (U_i, V_i) which determine shape of the control airfoil. The points are scaled so that the magnitude of the difference of the U-coordinates of the first and last point is equal to the chord length (as computed from the planform). The scaled points have coordinates (U', Z) where U' is distance along the airfoil and Z is in the wing coordinate system. WING1B also writes airfoil data on the output tape. A control airfoil is considered to be a single point if only a single point is given or if the first and last given points have the same abscissa; in either case no scaling takes place, and the ordinate of the first point will be taken as Z. If the set of given control airfoil points contain n > 2 points and has a nonzero chord length, but the true chord length (from the planform) is zero, then the scaled airfoil will have n points with all coordinates (0, 0); this is an alternate way of specifying a point airfoil in the x,y plane.

USAGE:

```
DIMENSION  B(1), NAME(2, 2, 1), LPLF(2), NLTEP(3),
           PT(2, 2, 1), TITLE(12), DAT(2)

LOGICAL  LOGUL(2)

CALL WING1A (B, NAME, LPLF, NLTEP, LOGUL, PT,
             LI, LO, TITLE, DAT, MU)
```

Input:	B	= Storage buffer for variable length arrays.
	LPLF	= The points (X_i, Y_i) that define the leading edge of the wing are stored starting in B(LPLF(1)); similarly, the points on the trailing edge start in B(LPLF(2)).
	NLTEP(1)	= Number of points on the leading edge.
	NLTEP(2)	= Number of points on the trailing edge.
	NLTEP(3)	= Number of control airfoils = NAF.
	LOGUL	= LOGUL(1) = .TRUE. if the upper wing surface is to be defined; LOGUL(2) = .TRUE. if the lower wing surface is to be defined.
	LI	= Input tape number.
	LO	= Output tape number.
	TITLE	= Title (alphameric)
	DAT	= Date (alphameric)
Output:	B	= Storage buffer for variable length arrays.
	NAME	= Array that tells where in B the control airfoil points are stored, and the number of elements in each airfoil. The points (U_i, Z_i) for airfoil j are stored starting in B(NAME(1, i, j)), where i = 1 for upper surface of the airfoil, i = 2 for lower surface of the airfoil. The corresponding number of elements (twice the number of points) is stored in NAME(2, i, j).
	PT	= Array containing the intersections (X_i, Y_i) of the control airfoils with the leading and trailing edges of the wing. The X-coordinate of the intersection of control airfoil j with edge i is stored in PT(1, i, j), where i = 1 for leading edge, and i = 2 for trailing edge; similarly, the Y-coordinate is stored in PT(2, i, j).
	MU	= Error indicator, which is zero for success.

SUBPROGRAMS CALLED:

DISPTA
WING1B
SIN (Built-in function)
POLXN
COS (Built-in function)
IRSERV (See Appendix A)

ERROR RETURNS: See WINGA, card 6, for explanation of symbols.

MU = 5	AFK is not 1., 2., or 3.
MU = 6, 7	YL (or YT) is outside the range of the leading edge (or trailing edge). If AFK = 1. or 2., MU = 6, if AFK = 3., MU = 7.
MU = 8	The control airfoil (defined by AFK, BETA, YL, YT) does not intersect the leading edge (or trailing edge) in exactly one point.
MU = 9	AFNU or AFNL is zero, indicating that this control airfoil has the same shape as the previous airfoil, but this is the first airfoil.
MU = 10	The storage buffer, B, is not large enough to store the control airfoil points.
MU = 11	This control airfoil has the same shape as the previous airfoil, but the previous one was a single-point airfoil.

RESTRICTIONS: The given control airfoil points are not checked to see that they have reasonable values. The storage buffer, B, must have been initialized by subroutine INIBFR (Appendix A) or its equivalent.

STORAGE: $719_{10} = 1317_8$

SUBJECT: FORTRAN IV Subroutine WING1B

PURPOSE: To read and write data points describing a control airfoil and to scale both coordinates of each point so that the airfoil has a given chord length.

METHOD: After reading the control airfoil points (see NOTE, below), from an input tape, the point coordinates are written on an output tape. The chord length of a control airfoil is taken as the magnitude of the difference in X-coordinates of the first and last points. Then the airfoil is scaled by multiplying both coordinates of each point by the ratio of a given chord length to the computed chord length.

An option is provided so that instead of reading control airfoil points from cards, a given array of points is used; thereafter, the procedure is the same. This feature is useful when processing a series of airfoils which are similar to a preceding airfoil. Instead of writing out the (unscaled) points, the remark "THE GIVEN....AIRFOIL IS THE SAME AS PRECEDING ONE" is written.

An error code is returned if a control airfoil has less than two points or if the computed chord length is zero. Then the remark "THE AIRFOIL IS A POINT" is written. Since a single-point control airfoil may be legitimate, the calling program can take appropriate action.

USAGE: DIMENSION B(1), A(1)

CALL WING1B (LI, LO, UL, CHORD, NB, B, NA, A, NU

Input:

LI	=	Input tape number.
LO	=	Output tape number.
UL	=	Hollerith word "UPPER" or "LOWER."
CHORD	=	Chord length of the control airfoil.
NB	=	Number of elements (twice the number of points) in array B.
B	=	Array of points in a given control airfoil, stored (x_1, z_1, x_2, z_2,...). B is not used unless NA < 2.
NA	=	If NA < 2, this control airfoil is similar to the given airfoil, B, but has a chord length = CHORD; the airfoil points in B will

be used instead of being read from cards. If NA $\geq$ 2, NA is the number of elements (twice the number of points) in the control airfoil. NA elements will be read on tape LI.

Output:	NA	Twice the number of points in A.
	A	Array of NA/2 points in the control airfoil, scaled to chord length = CHORD, and stored (x_1, z_1, x_2, z_2,...).
	NU	Error indicator, which is zero for success.

SUBPROGRAM CALLED: None

ERROR RETURNS: NU = -1 if the control airfoil has less than two points or if the first and last point have the same x-coordinate. NU = 1 if NA is less than 2 (on input), and NB is less than 4.

RESTRICTIONS: A warning message is printed if the points in A are not increasing in x, but this condition is not considered to be an error.

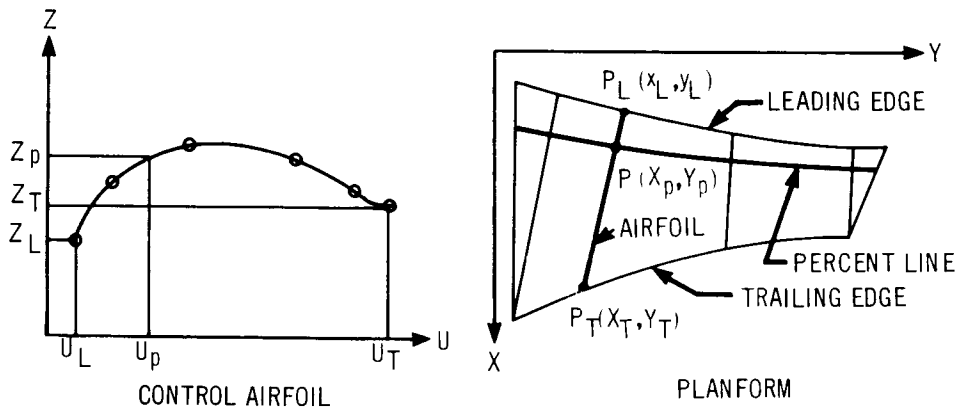
STORAGE: $332_{10} = 514_8$

NOTE: See cards 7 or 8, Wing Input Data Format (attachment to subroutine WINGA).

SUBJECT: FORTRAN IV Subroutine WING2

PURPOSE: To construct three-dimensional points on wing percent chord lines (corresponding to given control airfoils), and, if required, to interpolate additional points.

METHOD: The input includes NAF control airfoils and NPL percent values. It is necessary to compute NAF points (X_i, Y_i, Z_i) on each of the NPL percent chord lines. A point corresponding to a particular control airfoil and a percent chord line is found thus:



A set of points (U_i, Z_i) which describe the control airfoil is illustrated in first figure above. The (X, Y) coordinates are given for points P_L and P_T, shown in second figure. The control airfoil is to be placed on the planform so that U_L corresponds to P_L, and U_T to P_T; the U-coordinates of the airfoil may require scaling to do this, but the Z-coordinates are not to be scaled. Let p = percentage/100. Then

$$X_P = (1 - p) X_L + pX_T$$

$$Y_p = (1 - p) Y_L + pY_T$$

$$U_p = (1 - p) U_L + pU_T$$

The value of Z_P is found by interpolation (using subroutine BITURP) from the control airfoil points (U_i, Z_i) at $U = U_P$, if a control airfoil is described by only one point, $X_P = X_L$, $Y_P = Y_L$, and $Z_P = Z_L$.

After the basic percent chord lines are computed, they are written on the output tape by subroutine WING2P. Subroutine RICHNA is then called to remove any near-coincident points in each percent chord line, to interpolate additional points (if CHD > 0.), and to place a "header" (table of contents) in front of the array. If both upper and lower wing surfaces are defined, the entire process is repeated for the lower surface. Then the control airfoil point arrays and the NAME array are deleted from buffer B, since they are no longer needed.

USAGE:

```
DIMENSION  B(1), LOK(3,2), KOD(2), PLT(2,2,NAF),
NAME(2,2,NAF), PCT(NPL), TITLE(12), DAYT(2), NU(3)

LOGICAL  LGDEF(3,6), LOGUL(2)

CALL  WING2 (B, LO, LGDEF, LOK, LOGUL, NAF,
NPL, LAFN, KOD, PLT, NAME, PCT, CHD, EP, TITLE,
DAYT, NU)
```

Input:

B = Storage buffer for variable length arrays.

LO = Output tape number.

LOGUL = LOGUL(1) = .TRUE. if upper wing is defined;

LOGUL(2) = .TRUE. if lower wing is defined.

NAF = Number of control airfoils.

NPL = Number of percent chord lines.

LAFN = The array NAME starts in B(LAFN); that is, the location in core of NAME (1,1,1) is the same as the location of B(LAFN).

KOD = Interpolation codes. KOD(1) is used by subroutine BITURP: KOD(2) is used by subroutine RICHNA. (See ACODE and PCODE, card 3, subroutine WINGA.)

PLT = Array of coordinates (X_i, Y_i) of intersection points of control airfoils with the leading and trailing edges of the

wing. The intersection point of the kth control airfoil with edge j is stored starting in PLT (1, j, k), where j = 1 for leading edge, = 2 for trailing edge. PLT corresponds to B(LLETE) in subroutine WING1.

NAME = An array that tells where in B the control airfoil points are stored, and the number of elements in each airfoil. The points (U'_i, Z_i) for control airfoil j are stored starting in B(NAME(1, i, j)), where i = 1 for upper surface of the airfoil, i = 2 for lower surface of the airfoil. The corresponding number of elements (twice the number of points) is stored in NAME(2, i, j). The NAME array is generated in subroutine WING1A.

PCT = Array of percentages associated with the wing percent chord lines.

CHD = Chord-arc tolerance. (See CHD, card 2, subroutine WINGA.)

EP = See EPS, card 3, subroutine WINGA.

TITLE = Title (alphameric).

DAYT = Date (alphameric).

NU = Error indicator array: NU(1) is not used, on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) < 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written.

Output: B = Storage buffer for variable-length arrays.

LGDEF = LGDEF(2,1) is set to .TRUE. if upper wing percent lines are successfully generated; LGDEF(2,2) is set to .TRUE. if lower wing percent chord lines are successfully generated.

LOK = Array which tells where in B the percent chord line arrays are located, and the length of each percent chord line array. Consider LOK(i, j). The upper wing surface corresponds to j = 1, the lower to j = 2. The percent chord line array, consisting of a header followed by point coordinates (see Appendix C), starts in B(LOK(1, j)). The number of percent chord lines in the array is stored in LOK(2, j). The total number of elements in the array is stored in LOK(3, j).

NU = Error indicator array. NU(1) is zero if no errors are detected. NU(3) may have been changed (see Input).

SUBPROGRAMS CALLED:

RICHNA
WING2P
BITURP
MERR
IDLETE
IRLEAS
IRSERV
IPACK

(See Appendix A)

ERROR RETURNS: Function MERR is used to write a message "ERROR i, CODE j IN SUBROUTINE WING2 DURING GEOMETRIC DEFINITION" if an error is detected.

i	j	Explanation
1	0	The storage buffer, B, is not large enough or a storage error has been detected.
5	0	
6	0	
8	0	
2	0	No points are given for a control airfoil (this error should be detected in a previous subroutine).
3	k	Error k from BITURP in interpolation from a given control airfoil to get a percent chord line point. The control airfoil has invalid points.
7	k	Error k from RICHNA.

RESTRICTIONS: The storage buffer, B, must have been initialized by subroutine INIFBR (Appendix A) or its equivalent.

STORAGE: $550_{10} = 1046_8$

SUBJECT: FORTRAN IV Subroutine WING2P

PURPOSE: To write wing percent chord lines on an output tape.

METHOD: The basic wing percent chord lines, computed by subroutine WING2, consist of three-dimensional points on the surface of a wing. After title and headings are written on each page, the points are written, one point per line.

USAGE:

```
DIMENSION  W(3, NPT, NPCT), TITLE(12), DAYT(2),
           PCT(NPCT)

CALL WING2P (LO, W, NPCT, NPT, TITLE, UPLO,
             DAYT, PCT)
```

Input:

LO	=	Output tape number.
W	=	Array on NPCT percent chord lines. Each line contains NPT points (x_i, y_i, z_i).
NPCT	=	Number of percent chord lines.
NPT	=	Number of points in each percent chord line.
TITLE	=	Title (alphameric).
UPLO	=	6H UPPER or 6H LOWER.
DAYT	=	Date (alphameric).
PCT	=	Array of percentages, corresponding to the percent chord lines in W.

Output: Printout on tape LO.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $205_{10} = 315_8$

SUBJECT: FORTRAN IV Subroutine WINGA

PURPOSE: To read data cards defining a wing and construct percent chord lines on its surface.

METHOD: Subroutine WING1 is called to read data cards (see attachment) that define the wing. Then subroutine WING2 constructs three-dimensional percent chord lines that lie on the wing surface.

USAGE:

```
DIMENSION  B(1), DAT(2), LOK(3, 9), TITLE(2), NU(3)
LOGICAL  LGDEF(3, 6)
CALL WINGA (B, LI, LO, DAT, LGDEF, LOK, LWPF,
            TITLE, NU)
```

Input:	B	= Storage buffer for variable length arrays.
	LI	= Input tape number.
	LO	= Output tape number.
	DAT	= Date (alphameric).
	NU	= Error indicator array used by lower level routines: NU(1) is not used on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written.
Output:	B	= Storage buffer for variable length arrays.
	LGDEF(1, 1)	.TRUE. if upper wing was requested.
	(1, 2)	.TRUE. if lower wing was requested.
	(1, 4)	.TRUE.
	(2, 1)	.TRUE. if upper wing successfully defined.
	(2, 2)	.TRUE. if lower wing successfully defined.

	(2, 4)	.TRUE. if WING1 was successful.
LOK	=	Array that tells where in B the wing percent chord line arrays and wing planform point array are located and the length of each array. Consider LOK (i, j). The upper wing percent chord line array corresponds to LOK(i, 1), the lower to LOK(i, 2); the planform point array corresponds to LOK(i, 7). An array starts in B(LOK(1, j)); the total number of elements in the array = LOK(3, j). LOK(2, 1) = number of upper percent chord lines, LOK(2, 2) = number of lower percent chord lines, and LOK(2, 7) = number of control airfoils = number of pairs of planform points (leading edge, trailing edge).
LWPF	=	The location of LWPF is the same as that of LOK(1, 7). (LWPF need not have been an argument.)
TITLE	=	Wing title (alphameric).
NU	=	Error indicator array. NU(1) is zero if no error is detected. NU(3) may have been changed (see INPUT, above).

SUBPROGRAMS CALLED:

WING1
WING2
IPACK (See Appendix A)

ERROR RETURNS: NU(1) $\neq$ 0 if an error occurred in WING1 or WING2. LGDEF(2, 4) will be .FALSE. if the error was in WING1, .TRUE. if in WING2.

RESTRICTIONS: The storage buffer, B, must have been initialized by subroutine INIBFR (Appendix A) or its equivalent. LGDEF(i, j), i = 1, 3; j = 1, 2, 4 must have been set to .FALSE.

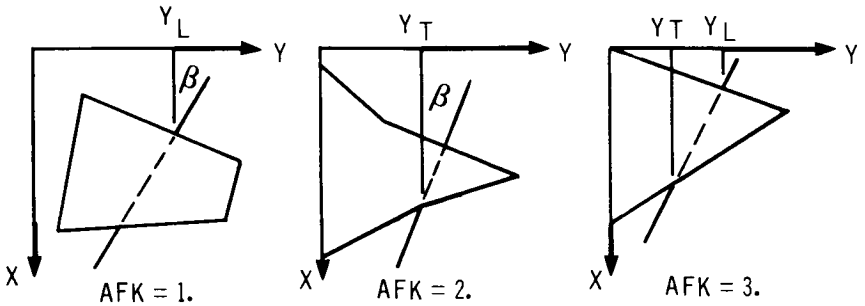
STORAGE: $153_{10} = 231_8$

ATTACHMENT: Wing Input Data Format

	Column	Code	Explanation
Card 1	1-72	TITLE	Any desired title.
Card 2	1-10	PNLE	Number of corner or break points defining the planform leading edge.
	11-20	PNTE	Number of corner or break points defining the planform trailing edge.
	21-30	AFN	Number of planform control airfoils. AFN $\geq$ 2. including the wing tip airfoil.
	31-40	PLN	Number of constant percent chord lines. PLN $\geq$ 1.
	41-50	WUL	= 1. only the upper wing surface is defined. = 2. only the lower wing surface is defined. = 0. both upper and lower wing surfaces are defined.
	51-60	CHD	Dimensional tolerance for the density of points interpolated (between control airfoils) on percent chord lines. If CHD $\leq$ 0., or if AFN $<$ 4., no additional points are generated. Do not specify CHD $>$ 0. unless <u>every</u> basic percent chord line has at least four <u>distinct</u> points.
Card 3	1-10	PCODE	Code that controls the type of three-dimensional interpolation to be used for generating additional points on percent chord lines. PCODE is not effectively used on any basic percent chord line of less than four distinct points or if CHD $\leq$ 0.. For a more detailed explanation of PCODE, see KD in subroutine RICH3A.

	Column	Code	Explanation
			= 0., enriching (interpolation of additional points) is done on points projected into a plane which "best" fits the basic percent chord line points. Enriched percent lines can appear curved between control airfoils in the planview.
			= 1., enriching is done on points projected into a plane normal to the x-y plane through the percent chord line ends. Resulting percent lines appear as straight lines between control airfoils in the planview.
			= 2., enriching is done as for PCODE = 1. but resultant percent lines can appear curved between control airfoils in the planview.
			NOTE: If PCODE = 1. or 2., and the percent chord line lies roughly in a horizontal plane (e.g., leading or trailing edge), enriching may not add points to the basic percent line.
	11-20	ACODE	= 1., linear interpolation is used between points on control airfoils to find points on the basic percent chord lines.
			= 2., biquadratic interpolation is used.
	21-30	EPS	A tolerance used to eliminate coincident percent chord line points. The basic percent chord lines consist only of intersections of percent chord lines with control airfoils. If two of these points are closer together than EPS, one is eliminated. If EPS $\leq$ 0., a value of 0.01 is used.
Cards 4	1-10	X_1	Array of points defining the planform leading edge, arranged in order from inboard to outboard. There must be PNLE point pairs, three coordinate
	11-20	Y_1	
	21-30	X_2	

	Column	Code	Explanation
	31-40	Y_2	pairs per card. Note that the leading edge may be modified by the program, depending on the placement of control airfoils on the planform.
	41-50	X_3	
	51-60	Y_3	
Cards 5	1-10	X_1	Similar to cards 4, but for the trailing edge.
	11-20	Y_1	
	21-30	X_2	
	31-40	Y_2	
	41-50	X_3	
	51-60	Y_3	
Cards 6			These cards, one for each control airfoil, give the airfoil location on the planform and the number of points on the upper and lower parts of the control airfoil. Each card 6 is followed by one or more cards 7 and 8 (unless the previous airfoil is to be repeated).
	1-10	AFK	Code to indicate how the control airfoil is oriented on the planform. See sketches below.



Two of the three quantities β , y_L, y_T must be given to specify the control airfoil location. AFK indicates which two quantities are to be used.

	11-20	BETA	The angle β, positive if as shown on the sketches. BETA is ignored if AFK = 3..
	21-30	Y_L	Y_L, Y-coordinate at which the control airfoil intersects the leading edge. Y_L is ignored if AFK = 2..

	Column	Code	Explanation
	31-40	Y_T	Y_T, Y-coordinate at which the control airfoil intersects the trailing edge. Y_T is ignored if AFK = 1..
	41-50	AFNU	Number of points on the control airfoil upper surface. The coordinates of these points (cards 7) follow this card. However, if AFNU = 0., then the previous airfoil upper surface points are used and a card 7 must not follow. If WUL = 2. (upper wing not defined), then AFNU is ignored, and a card 7 must not follow.
	51-60	AFNL	Number of points on the control airfoil lower surface. The coordinates of these points (cards 8) follow cards 7 (or card 6 if no card 7 is used). However, if AFNL = 0., then the previous airfoil lower surface points are used and a card 8 must not follow If WUL = 1. (lower wing not defined), then AFNL is ignored, and a card 8 must not follow. Note that the leading and trailing edge points must be included, even if they repeat those of the airfoil upper surface.
Cards 7	1-10	U_1	Array of points (U_i, V_i) on the control airfoil upper surface, in increasing order from leading edge to trailing edge. The points will later be scaled to fit the planform. This card must not follow a card 6 if WUL = 2. or if AFNU = 0..
	11-20	V_1	
	21-30	U_2	
	31-40	V_2	
	41-50	U_3	
	51-60	V_3	
Cards 8	1-10	U_1	Similar to card 7, but apply to the control airfoil lower surface points. Cards 8 follow cards 7 (or a card 6 if a card 7 is not used). A card 8 must not be used if WUL = 1. or if AFNL = 0..
	11-20	V_1	
	21-30	U_2	
	31-40	V_2	
	41-50	U_3	
	51-60	V_3	

	Column	Code	Explanation
Cards 9	1-10	P_1	Array of constant percent chord values corresponding to the spanwise panel edges. For example, if percent chord lines 0, 20, 50, 100 are desired, then $P_1 = 0.$, $P_2 = 20.$, $P_3 = 50.$, and $P_4 = 100.$.
	11-20	P_2	
	.		
	.		
	51-60	P_6	

SUBJECT: FORTRAN IV Subroutine WLDM

PURPOSE: To compute the coefficients of lift, drag, and pitching moment of the wing and the spanwise distribution of the coefficients of lift and drag.

METHOD: The force normal to a panel is given as the product of dynamic pressure, surface pressure coefficient, and panel area

$$F_i = q \cdot C_{p_i} \cdot A_i$$

Resolving into the components normal and parallel to the free-stream direction yields

$$L_i = - F_i \cdot \cos \theta_i$$

$$D_i = F_i \cdot n_i$$

where n_i is the component of velocity normal to the x-axis and θ_i is the angle between the plane of the panel and a plane parallel to the x-y plane.

The moment of force with respect to a point (x, 0, z) gives the pitching moment.

$$M_i = - L_i (\bar{x}_i - x) + D_i (\bar{z}_i - z)$$

where $\bar{x}$ and $\bar{z}$ are the x and z coordinates of the panel centroids.

The coefficients of lift, drag, and moment for the entire wing are given by

$$C_L = \frac{1}{q \cdot S_W} \sum_{i=1}^{NM} L_i$$

$$C_D = \frac{1}{q \cdot S_W} \sum_{i=1}^{NM} D_i$$

$$C_M = \frac{1}{q \cdot S_W} \sum_{i=1}^{NM} M_i$$

where S_W is the wing reference area, and NM is the number of wing panels.

The spanwise distribution of the coefficients of lift and drag is given by

$$C_{L_j} = \frac{1}{q \cdot S_W} \sum_{i=1}^{NROW} L_i$$

$$C_{D_j} = \frac{1}{q \cdot S_W} \sum_{i=1}^{NROW} D_i \quad , j = 1, \ldots, NCOL$$

where NROW is the number of panels in a chordwise column, and NCOL is the number of chordwise columns.

USAGE:

```
CALL WLDM (NM, NROW, XP, ZP, RFAREA, AREA,
           XBAR, ZBAR, ALPHAM, THETAM, CPM,
           SCL, SCD, CL, CD, CM)

DIMENSION  AREA(NM), XBAR(NM), ZBAR(NM),
           ALPHAM(NM), THETAM(NM), CPM(NM),
           SCL(NCOL), SCD(NCOL)
```

Input:

NM	=	Number of wing panels.
NROW	=	Number of panels in a steamwise column.
XP	=	x-coordinate of the point about which the pitching moments are to be calculated.
ZP	=	z-coordinate of the point about which the pitching moments are to be calculated.
RFAREA	=	Wing reference area.
AREA	=	Array of wing panel areas.
XBAR	=	Array of wing panel centroid x-coordinates.
ZBAR	=	Array of wing panel centroid z-coordinates.
ALPHAM	=	Array of velocity components normal to the x-axis.
THETAM	=	Array of angles between the plane of the panel and a plane parallel to the x-y plane.

Output:

SCL	=	Array of spanwise lift coefficients.
SCD	=	Array of spanwise drag coefficients.
CL	=	Coefficient of lift.

CD	=	Coefficient of drag.
CM	=	Coefficient of pitching moment.

SUBPROGRAM CALLED: COS

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $187_{10} = 273_8$

5. APPENDIXES

5.1 Appendix A — Dynamic Storage Allocation Package

SUBJECT: Dynamic Storage Allocation in FORTRAN

PURPOSE: To provide a means for allocating, using, and releasing blocks of storage on a dynamic basis during execution of FORTRAN programs.

METHOD: The buffer format is illustrated in figure 1, page 419. Each buffer is divided into a (variable) number of arrays, some of which may be full while others are available to be filled. Initially the buffer contains only one array; this is empty. The first word of each buffer is a "header" which contains the total length of the buffer.

Each array is preceded by a "secret" header of two codes, illustrated in figure 1, which contains the following information:

1) Whether the array is full or empty

2) The length of the array

3) The relative address of the first word of the buffer

4) The absolute address of the array name (if the array is full).

In using the buffer, a programmer can ignore the headers, because the subroutines he uses will accommodate this bookkeeping. These routines will also indicate situations such as an overflowing buffer or reference to an array in the buffer that is not a legitimate array.

The subroutines involved in the scheme may be divided into three classes:

1) Basic subprograms

2) Primitive subprograms

3) User's subprograms

The first two of these classes are an integral part of the automatic storage scheme and are discussed in more detail below. Their structure is illustrated in figure 2, page 420. The user's subprograms are provided by the authors of the computing system and will call the basic subprograms into execution. The user should never have to call a primitive subprogram directly. When setting up an array in the buffer, either by use of the function IADARY or by reserving space with the function IRSERV, the user will obtain a "name" for that array stored as a FORTRAN integer in a cell that may be called NAME. This name is now uniquely

associated with the array. If the buffer is called BUFFER, the first word of the array will always be BUFFER(NAME) and the ith word will be BUFFER(NAME + I - 1), regardless of any shuffling that may take place automatically in the buffer. However, because data shuffling may occur, the array must always be referred to by the same NAME, which will be kept updated automatically. Obviously, each array can have only one name.

USAGE:

Basic Subprograms: The following seven subprograms form the basic set which can be called directly by the user. With the exception of INIBFR and CLEAR they are coded in FORTRAN.

1) Subroutine CLEAR (BFRA, BFRB, ...)

   To reinitialize and make available for reuse any number of buffers BFRA, BFRB, ..., etc.

2) Subroutine INIBFR (BFRA, NA, BFRB, NB, ...)

   To initialize any number of buffers, BFRA, etc., of length NA, etc., words. This routine must be called before a buffer can be used.

3) Function IADARY (ARRAY, N, BFR, NAME)

   To place an array, ARRAY, of length N words, into the buffer, BFR, and to assign to it the name, NAME. The function value will be the same as NAME unless some error or misuse is detected, in which case it will be negative. To circumvent the oddities of the FORTRAN subscripting, procedure IADARY should be used thus:

   NAME = IADARY (ARRAY, N, BFR, NAME)

4) Function IDLETE (BFR(NAME))

   To delete (i.e., make available for reuse) an array in the buffer, BFR, whose name is NAME. The function value will be zero unless some error is detected, when delection will not occur.

5) Function IPACK (BFR)

   The pack a buffer, BFR, by moving all full arrays to the beginning of the buffer and condensing all the empty arrays into one at the end. The function value will be equal to the length of the final empty array unless an error is detected, when it will be negative and packing will not occur.

6) Function IRLEAS (BFR(NAME), N)

To release all cells beyond cell N in an array in the buffer, BFR, whose name is NAME; i.e., BFR(NAME) - BFR(NAME + N-1) are retained but BFR(NAME + N) - the end of the array are made available for reuse. The function value will be zero unless some error is detected, when release will not occur.

7) Function IRSERV (N, BFR, NAME)

To reserve an array of N words in the buffer, BFR, and to assign to it the name, NAME. The comments under IADARY concerning function value apply also to IRSERV.

Primitive Subprograms: For completeness, the eight primitive subprograms of the scheme are briefly described here. They are all coded in assembly language.

1) Function ARFRE (NAME)

To set up the first code word of an empty array. FORTRAN IV version has deck name ARFREE.

2) Function ARFUL (NAME)

To set up the first code word of a full array.

3) Function ARSEC (N, NAME)

To set up the second code word of an array. FORTRAN IV version has deck name ARSECR.

4) Function UFOOL(I)

To store a fixed point variable in a floating point location. The FORTRAN IV version has deck name UFOOL and two entry points, FOOL and IFOOL.

5) Function IBFR (A)

To locate the first word of a buffer relative to the first word of an array.

6) Function ICODE (A)

To determine whether an array is full or empty.

7) Function LENG (A)

To determine the length of an array

8) Subroutine UPDATE (A)

To update the contents of an array name following data shuffling in the buffer.

EXAMPLE: An example is given in figure 3, page 421, of a program which reads in an (ℓx m) and an (m x n) matrix, multiplies them together and prints out the (ℓx n) resultant. The input arrays are then deleted from the buffer.

FLOWCHARTS: Flowcharts of all the primitive and basic subprograms of the dynamic storage package are included. These flowcharts are for the FORTRAN II versions. The FORTRAN IV subprograms differ slightly because of the forward storage of arrays and full word instead of decrement format of fixed-point variables.

SUBPROGRAM CALLED: None

EQUIPMENT: IBM 7094 II

STORAGE:

	FORTRAN II		FORTRAN IV	
Subprogram	Octal	Decimal	Octal	Decimal
ARFREE	10	8	7	7
ARFUL	10	8	7	7
ARSECR	10	8	11	9
CLEAR	25	21	22	18
IADARY	273	187	255	173
IBFR	24	20	23	19
ICODE	23	19	22	18
IDLETE	65	53	62	50
INIBFR	33	27	33	27
IPACK	316	206	266	182
IRLEAS	173	123	165	117
IRSERV	42	34	45	37
LENG	4	4	5	5
UFOOL	2	2	2	2
UPDATE	14	12	13	11
TOTALS	1334	732	1252	682

BFR (1)	4	0	7	N
2	P_1	0	7	4
3	0	M_1	0	$ANAME_1$
4 ⋮ M_1+K_1-1	Array #1			
K_i-2	P_i	0	7	K_i
K_i-1	0	M_1	0	$ANAME_i$
K_i	Array #i			
K_j-2	P_j	0	7	K_j
K_j-1	0	M_j	0	$ANAME_j$
K_j ⋮ $N = K_j+M_j-1$	Array #j (last)			

NOTES

For a full array:

$P_i = 7$

$ANAME_i$ is the absolute address of the array name, $NAME_i$. The word $NAME_i$ contains K_i as a decrement integer.

For an empty array:

$P_i = 0$

$ANAME_i$ is irrelevant.

In general:

M_i is the length of the ith array.

K_i is the first word of the array relative to the beginning of the buffer.

N is the total length of the buffer.

The first code word of the buffer and of the arrays are illegal both as FORTRAN integers and as floating point numbers.

FIGURE 1 BUFFER FORMAT

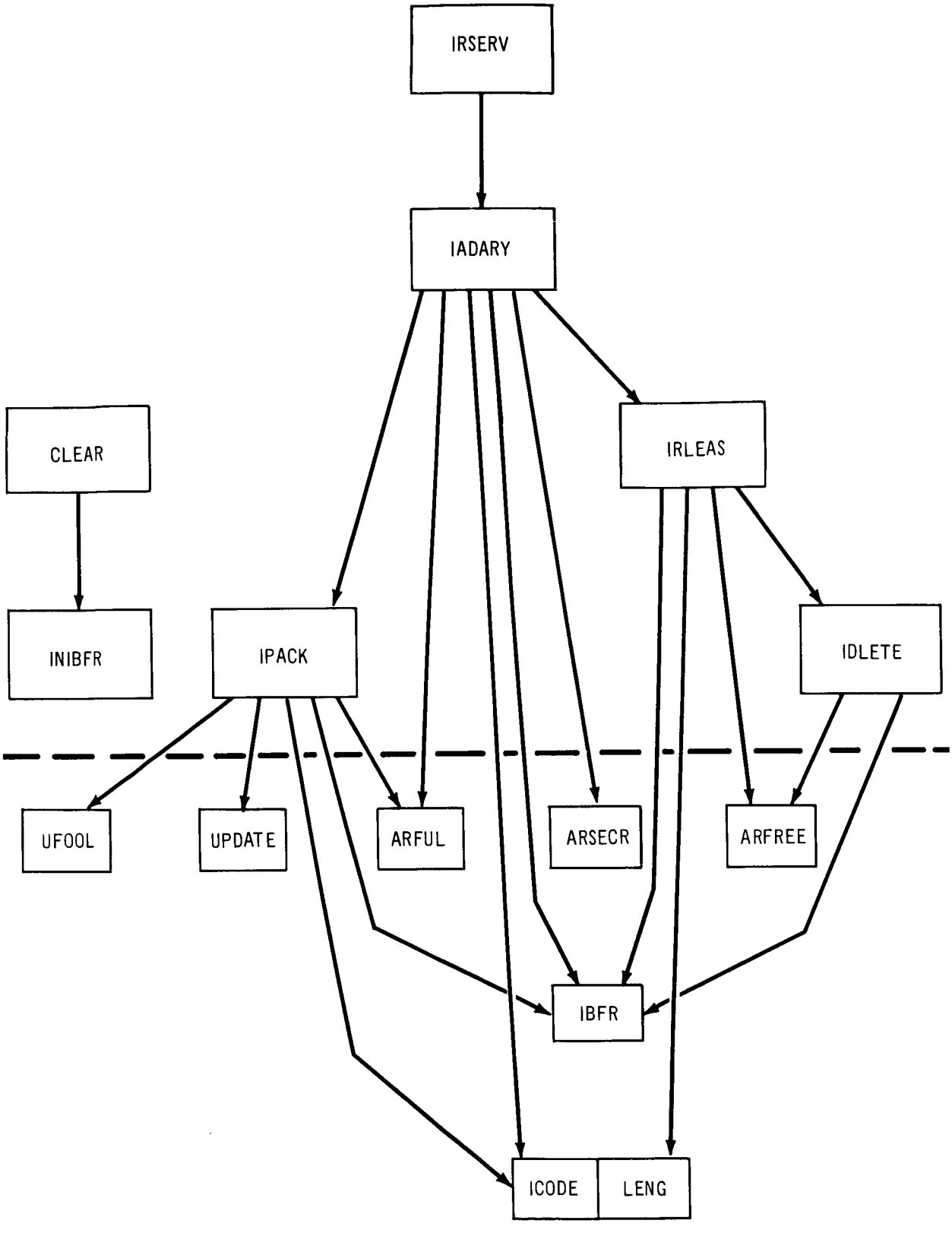
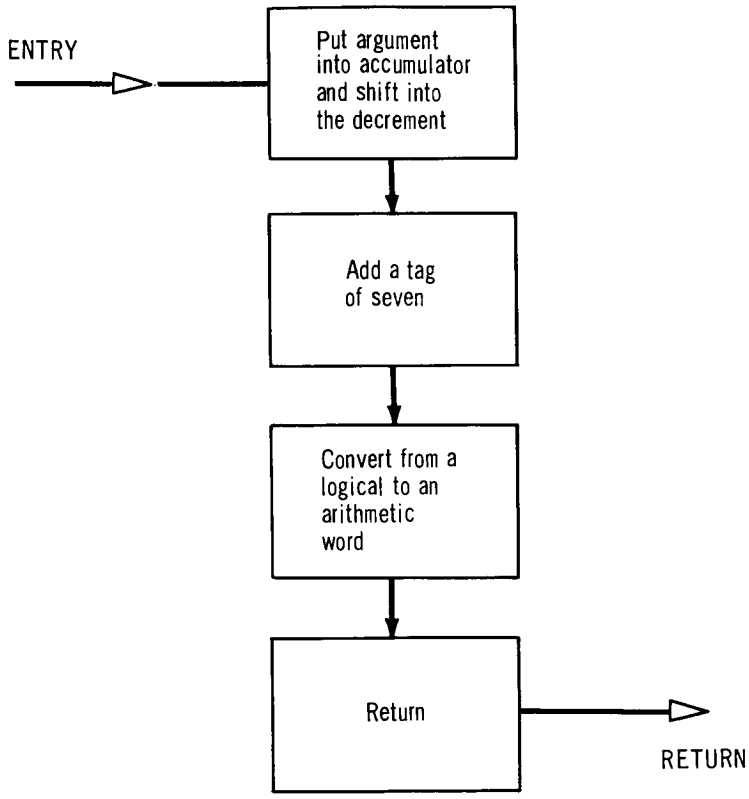


FIGURE 2 STRUCTURE OF THE DYNAMIC STORAGE SCHEME

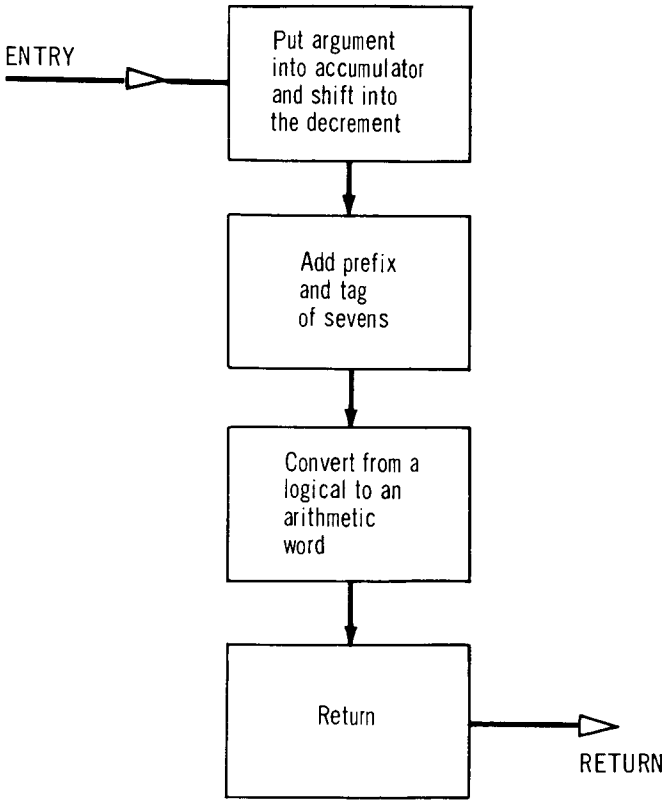
```
      DIMENSION BFR(1000)
C
      CALL INIBFR (BFR,1000)
C
      READ INL

      READ INPUT TAPE 5,1,L,M,N
    1 FORMAT (6G)
X

C
      NAM1=IRSERV(L*M,BFR,NAM1)
      NAM2=IRSERV(M*N,BFR,NAM2)
      IJ=IRSERV(L*N,BFR,NAM3)-1

C
      K=NAM1+L=M-1
      READ INPUT TAPE 5,1,(BFR(I),I=NAM1,K)
      K=NAM2+M*N-1
C     READ INPUT TAPE 5,1,(BFR(1),I=NAM2,K)

      DO 20  J=1,N
      DO 20  I=1,L
      IK=NAM1+I-1
      KJ+NAM2+(J-1)*M

C
      DO  10  K=1,M
      B=B+BFR(IK)*BFR(KJ)
      IK=IK+L
   10 KJ=KJ+1

C
      IJ=IJ+1
   20 BFR(IJ)=B
C
      CALL  IDLETE(BFR(NAM1))
      CALL  IDLETE(BFR(NAM2))
C
      WRITE OUTPUT TAPE 6,2,(BFR(I),I=NAM3,IJ)
    2 FORMAT (G20.6)
      STOP
      END(1,1,0,0,0,0,0,1,0,1,0,0,0,0,0)
```

FIGURE 3 EXAMPLE FOR DYNAMIC STORAGE

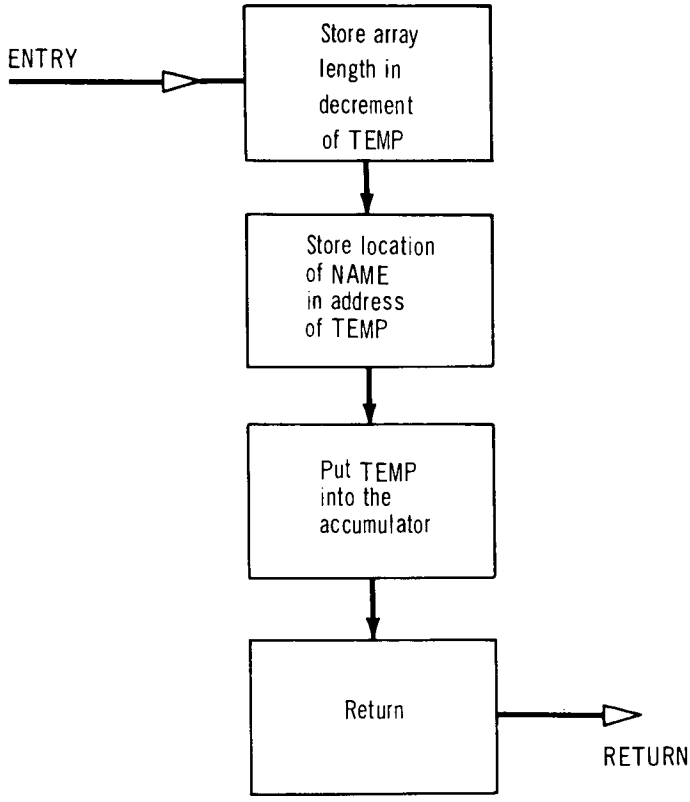
ARFREE


ENTRY
Put argument into accumulator and shift into the decrement
Add a tag of seven
Convert from a logical to an arithmetic word
Return
RETURN

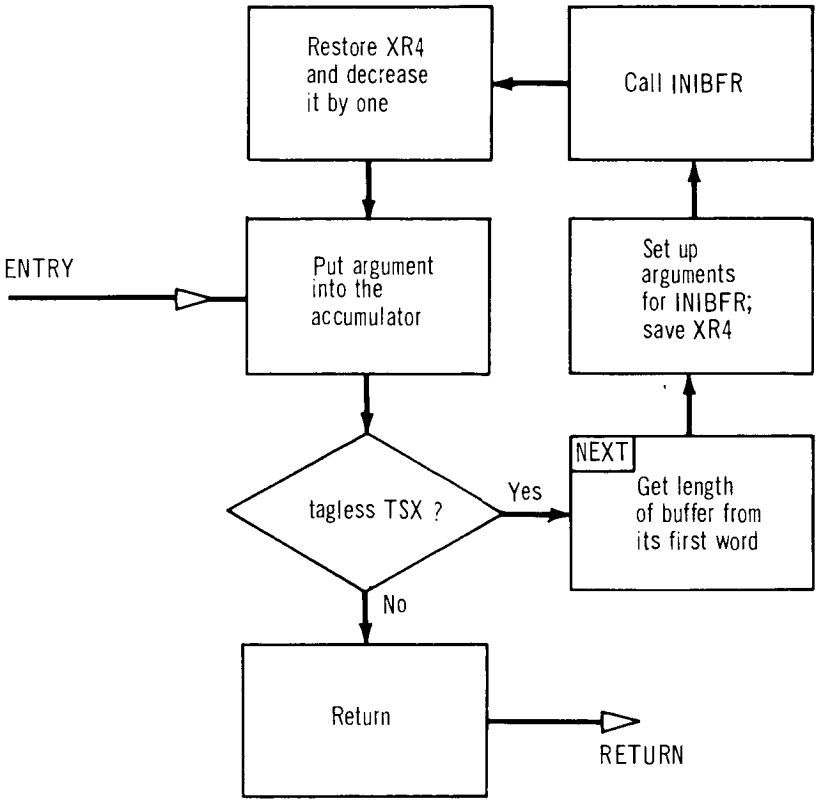
ARFUL

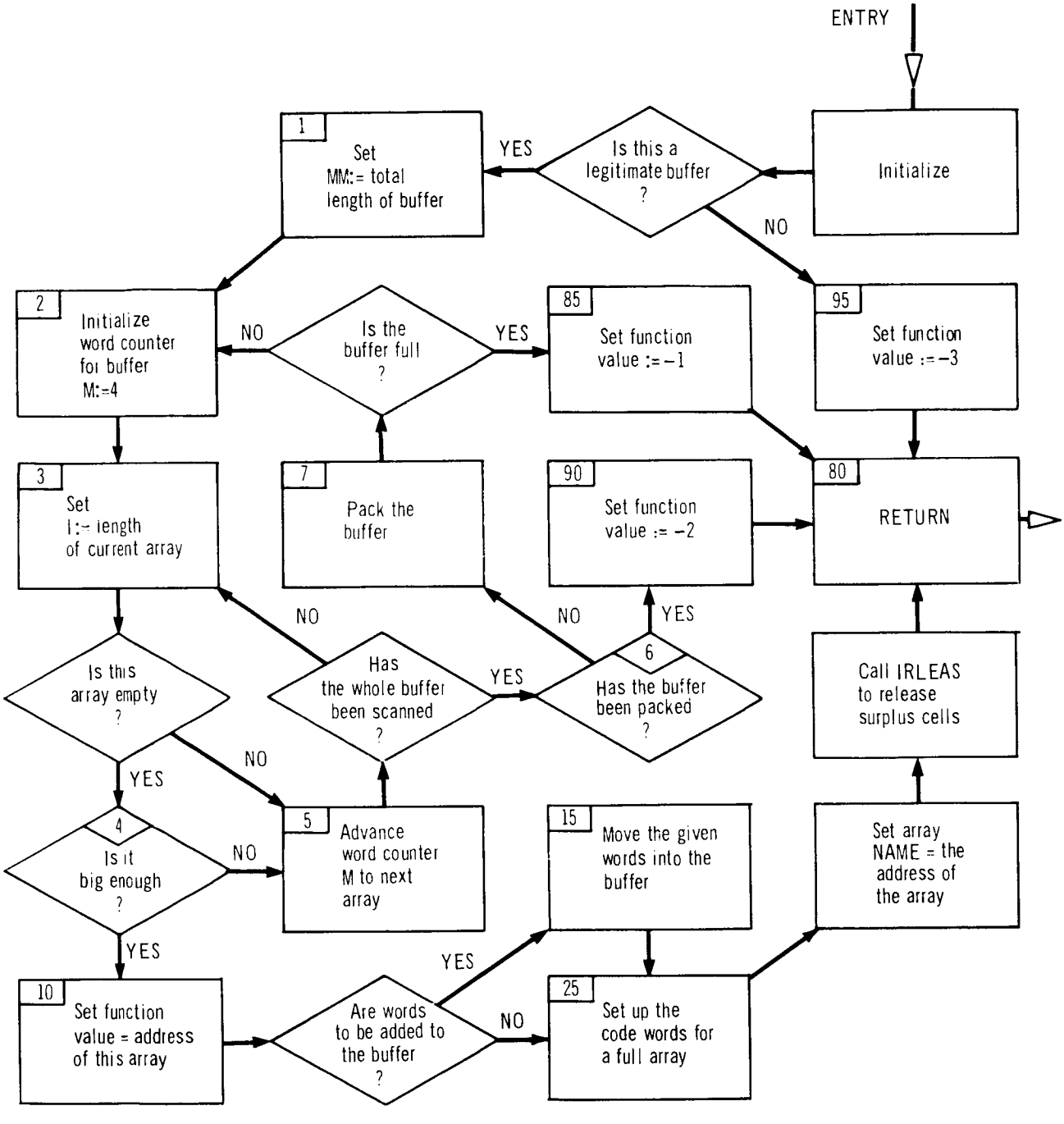

ENTRY
Put argument into accumulator and shift into the decrement
Add prefix and tag of sevens
Convert from a logical to an arithmetic word
Return
RETURN

ARSECR

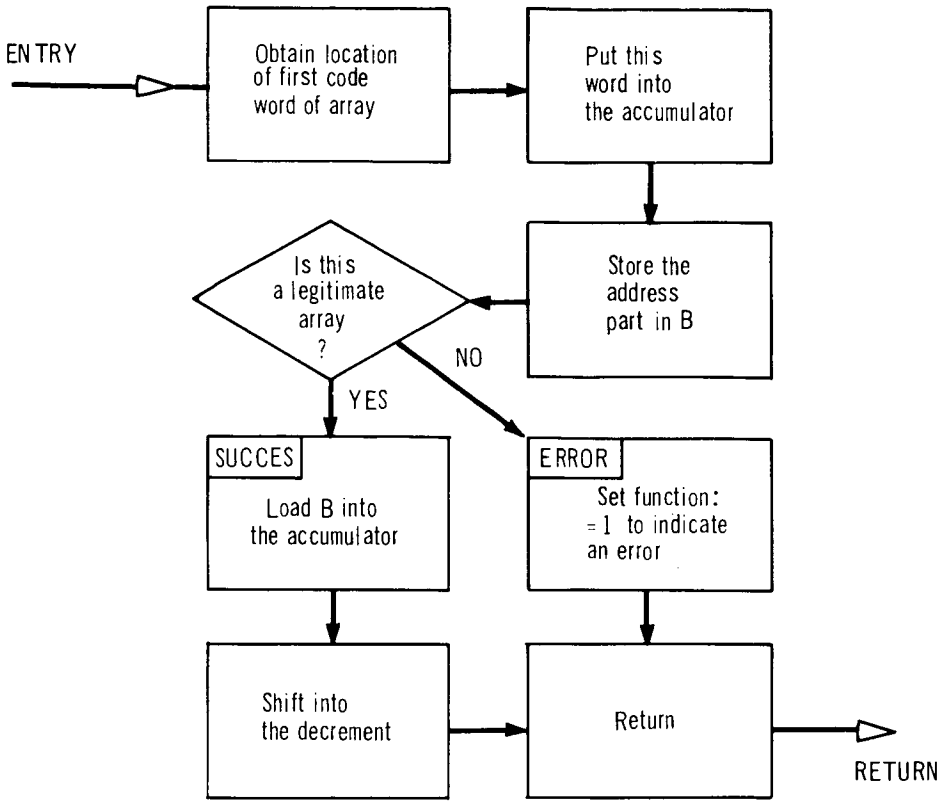

ENTRY
Store array length in decrement of TEMP
Store location of NAME in address of TEMP
Put TEMP into the accumulator
Return
RETURN

CLEAR

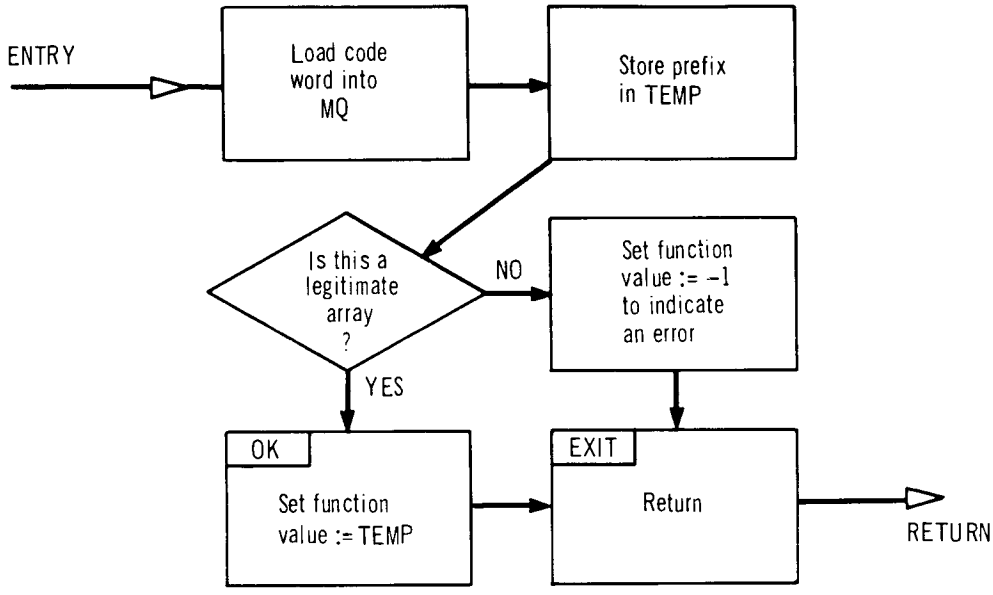

Restore XR4
and decrease
it by one
Call INIBFR
ENTRY
Put argument
into the
accumulator
Set up
arguments
for INIBFR;
save XR4
NEXT
tagless TSX ?
Yes
Get length
of buffer from
its first word
No
Return
RETURN


ENTRY
Initialize
Is this a legitimate buffer ?
YES
NO
1
Set MM:= total length of buffer
95
Set function value := −3
2
Initialize word counter for buffer M:=4
NO
Is the buffer full ?
YES
85
Set function value := −1
80
RETURN
3
Set I:= length of current array
7
Pack the buffer
90
Set function value := −2
NO
NO
YES
6
Has the buffer been packed ?
Has the whole buffer been scanned ?
YES
Is this array empty ?
NO
YES
Call IRLEAS to release surplus cells
4
Is it big enough ?
NO
5
Advance word counter M to next array
15
Move the given words into the buffer
Set array NAME = the address of the array
YES
YES
10
Set function value = address of this array
Are words to be added to the buffer ?
NO
25
Set up the code words for a full array

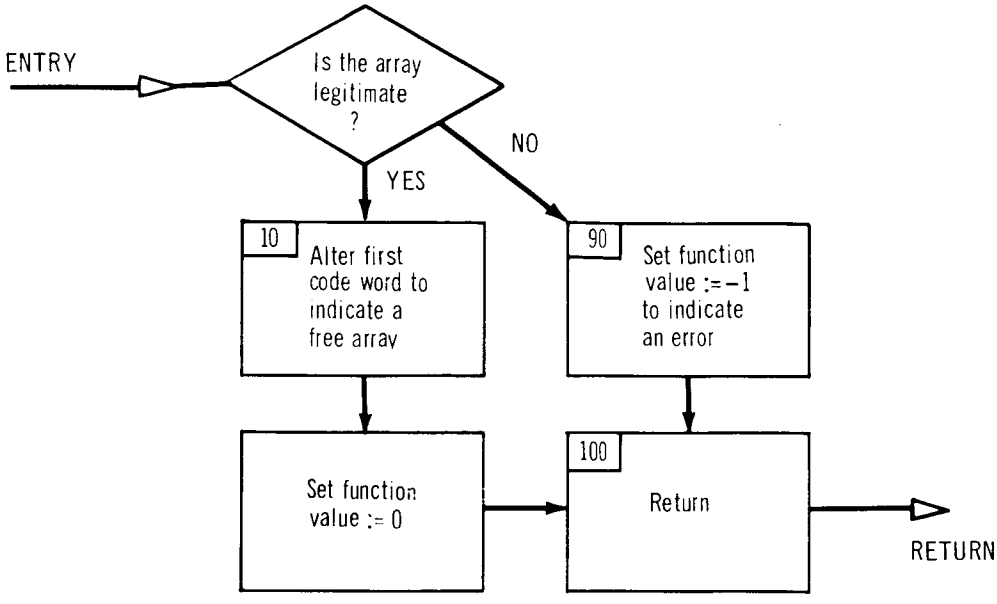
IBFR


ENTRY
Obtain location of first code word of array
Put this word into the accumulator
Store the address part in B
Is this a legitimate array ?
NO
YES
SUCCES
Load B into the accumulator
ERROR
Set function: =1 to indicate an error
Shift into the decrement
Return
RETURN

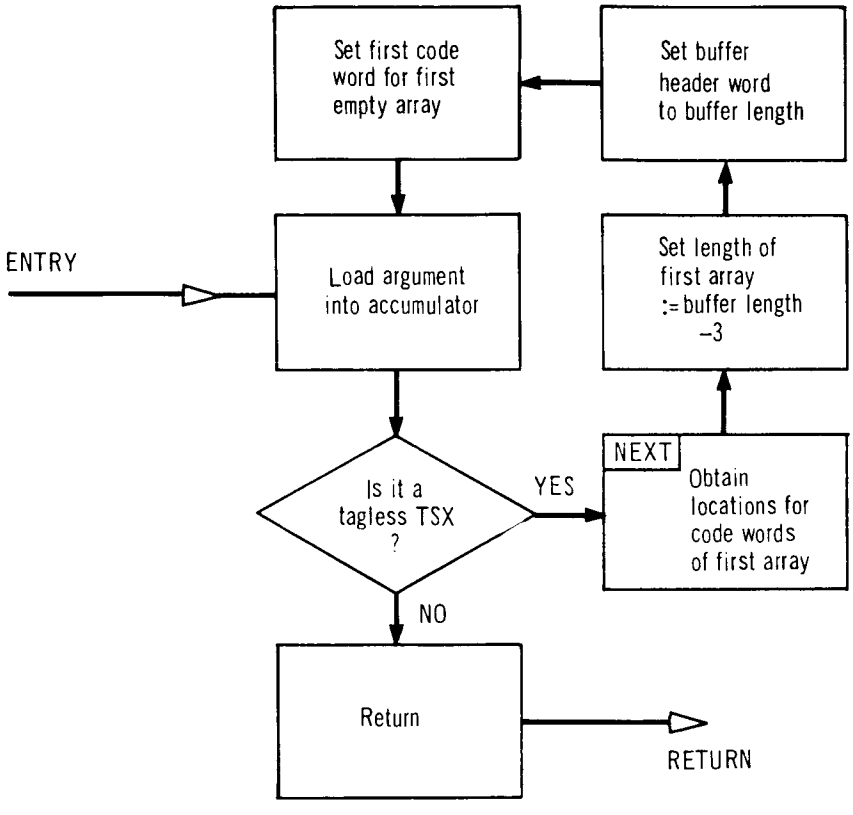
ICODE

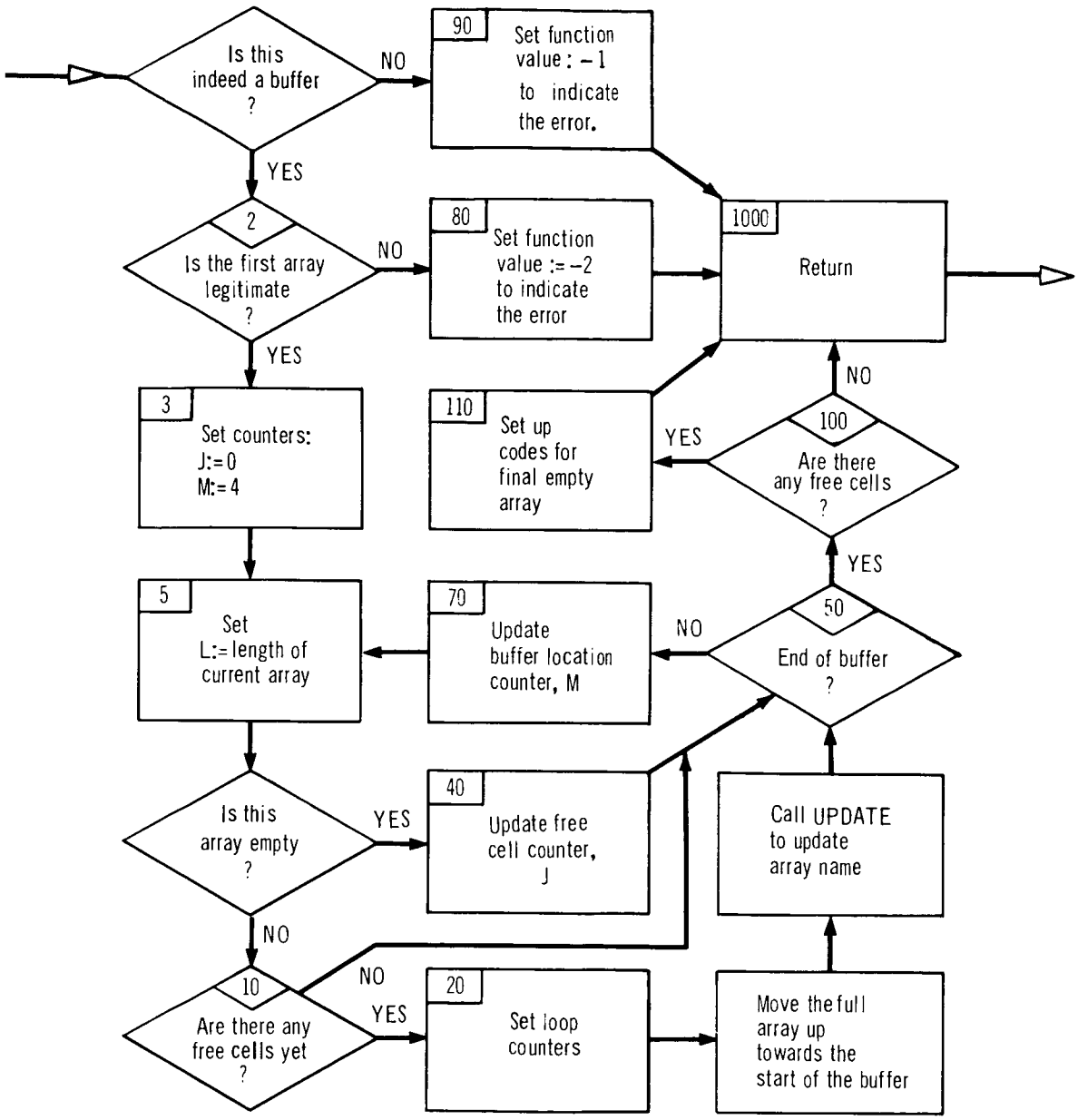

ENTRY
Load code word into MQ
Store prefix in TEMP
Is this a legitimate array ?
NO
Set function value := −1 to indicate an error
YES
OK
Set function value := TEMP
EXIT
Return
RETURN

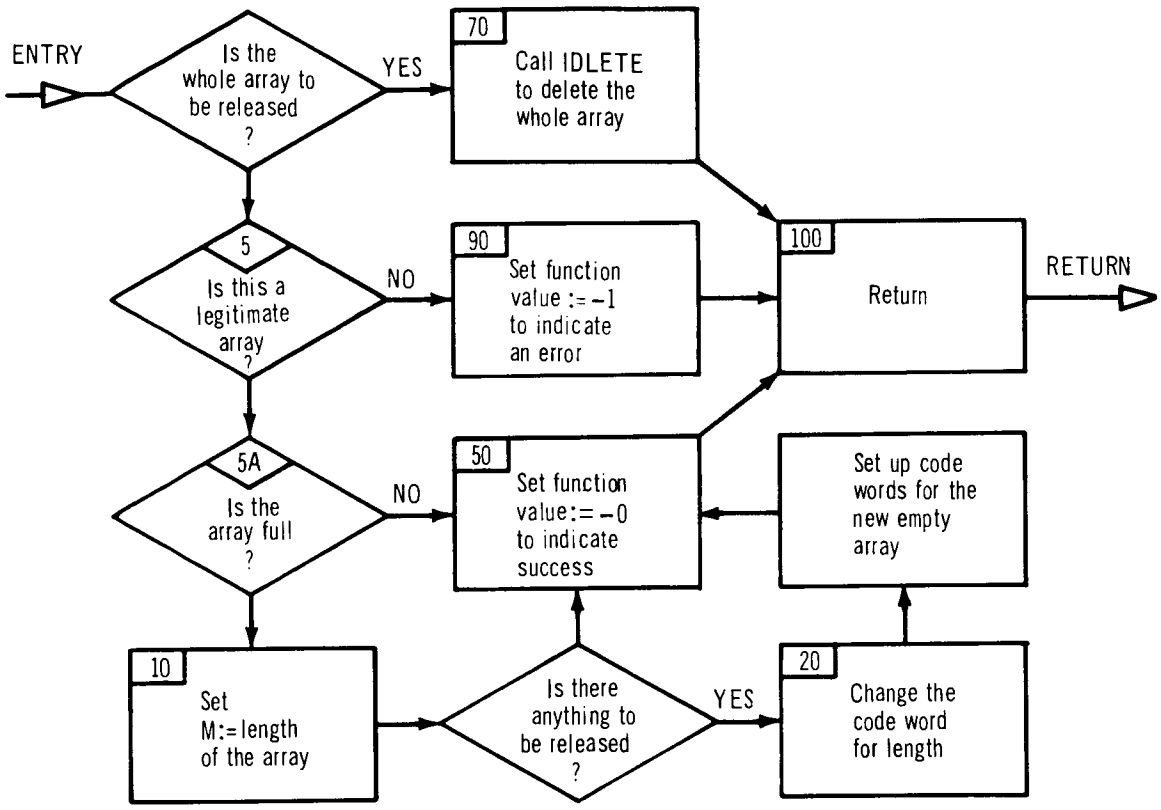
IDLETE


ENTRY
Is the array legitimate ?
NO
YES
10
Alter first code word to indicate a free array
90
Set function value := −1 to indicate an error
Set function value := 0
100
Return
RETURN

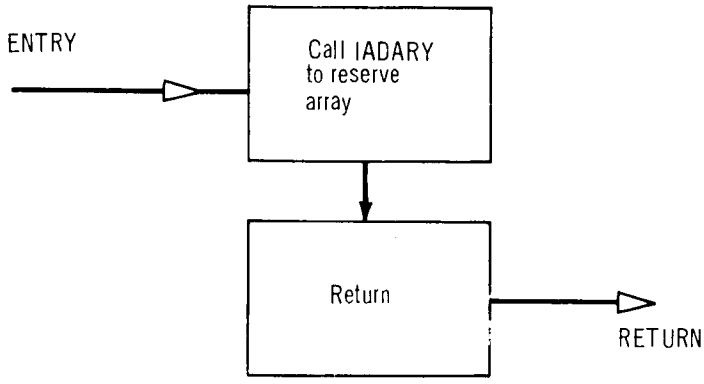
INIBFR


Set first code word for first empty array
Set buffer header word to buffer length
ENTRY
Load argument into accumulator
Set length of first array := buffer length −3
NEXT
Is it a tagless TSX ?
YES
Obtain locations for code words of first array
NO
Return
RETURN

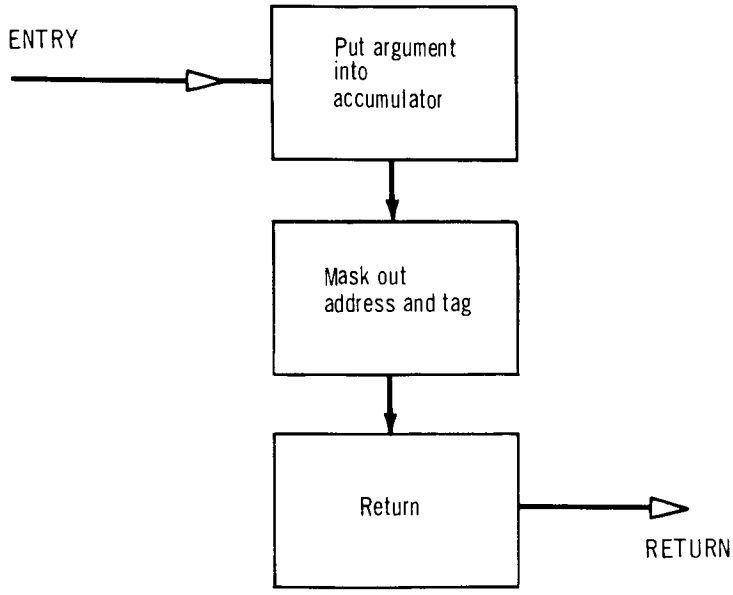

Is this indeed a buffer ?
NO
90
Set function value : −1 to indicate the error.
YES
2
Is the first array legitimate ?
NO
80
Set function value := −2 to indicate the error
1000
Return
YES
3
Set counters: J:=0 M:=4
110
Set up codes for final empty array
YES
100
Are there any free cells ?
NO
YES
5
Set L:= length of current array
70
Update buffer location counter, M
NO
50
End of buffer ?
Is this array empty ?
YES
40
Update free cell counter, J
Call UPDATE to update array name
NO
10
Are there any free cells yet ?
NO
YES
20
Set loop counters
Move the full array up towards the start of the buffer


ENTRY
Is the whole array to be released ?
YES
70
Call IDLETE to delete the whole array
5
Is this a legitimate array ?
NO
90
Set function value := -1 to indicate an error
100
Return
RETURN
5A
Is the array full ?
NO
50
Set function value:= -0 to indicate success
Set up code words for the new empty array
10
Set M:= length of the array
Is there anything to be released ?
YES
20
Change the code word for length

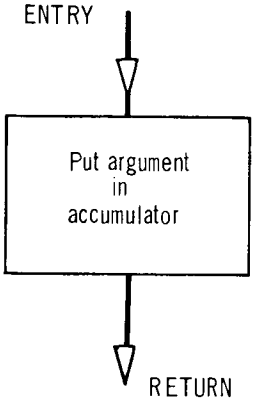
IRSERV



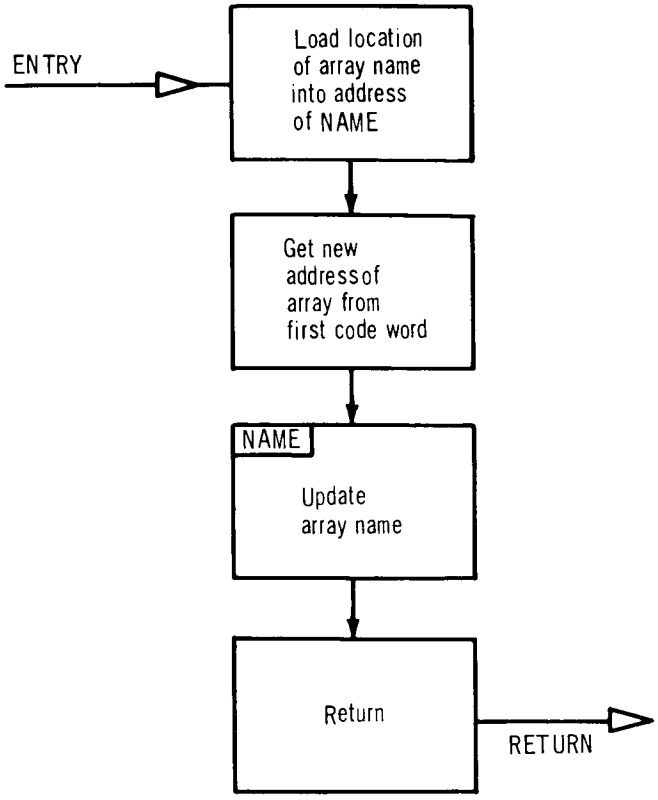
LENG


ENTRY
Put argument into accumulator
Mask out address and tag
Return
RETURN

UFOOL



UPDATE


ENTRY
Load location of array name into address of NAME
Get new address of array from first code word
NAME
Update array name
Return
RETURN

5.2 Appendix B — Matrix Inversion Package

SUBJECT: FORTRAN IV Subroutine SINVRT and MAP Subroutine INV4S

PURPOSE: To invert an n-by-n matrix and compute its determinant.

METHOD:

1. Before the inversion process starts, the matrix is scaled so that the element of largest magnitude is greater than or equal to 1/4 and less than 1/2. This is accomplished through fixed-point addition or subtraction in the characteristic, which avoids the introduction of round-off during scaling.

2. A theorem of matrix algebra states that for any nonsingular matrix A, there exist elementary transformations R and S such that:

$$RAS = I \qquad (1)$$

where I is the identity matrix. Because R and S are elementary transformations, they are necessarily nonsingular which permits taking the inverse of both sides of equation (1).

$$S^{-1} A^{-1} R^{-1} = I$$

Then solving for A^{-1} gives,

$$A^{-1} = SIR = SR$$

The transformations S and R are determined by reducing A to the identity matrix. In INV4S R consists of row interchanges and arithmetic operations on rows, while S consists solely of column interchanges.

3. The inversion is performed in single precision, and it leaves a single-precision inverse and determinant. Normal 7094 single-precision floating-point restrictions apply.

4. The inversion of a sparse matrix is speeded up by a checking feature that can sense a zero reduction factor and bypass a row reduction.

5. At the start of each reduction stage, rows and columns are interchanged to select the element of largest magnitude from the remaining submatrix and place it in the pivotal position.

USAGE:

CALL SINVRT (A, K, N, IRR1, IRR2, SCALE, DET, NDETXP)

A = Matrix to be inverted.

K = Row DIMENSION statement entry for A.

N = Number of rows (columns) in matrix (decrement integer).

IRR1 = Error code from INV4S (decrement integer).

IRR2 = Error code from INV4S (decrement integer).

SCALE = Error cell from INV4S.

DET = Mantissa part of determinant (single precision):

$$(1. < |DET| < 10.)$$

NDETXP = Exponent part of determinant (decrement integer):

$$-(2^{17} - 1) < NDETXP < (2^{17} - 1)$$

determinant = $DET \times 10^{(NDETXP)}$

CALL INV4S (A, N, IRR1, IRR2, SCALE, DET, NDETXP)

A = Location of (1, 1) matrix element.

N = Number of rows (columns) in matrix (address integer).

IRR1 = Error code 1 (address integer).

IRR2 = Error code 2 (address integer).

SCALE = Error cell.

DET = Mantissa part of determinant (single precision):

$$(1. < |DET| < 10.)$$

NDETXP = Exponent part of determinant:

$$-(2^{35} - 1) < NDETXP < (2^{35} - 1)$$

determinant = $DET \times 10^{(NDETXP)}$

SUBPROGRAMS CALLED:

A. SINVRT calls INV4S

B. INV4S None

ERROR RETURNS: Error codes for both subroutines:

IRR1	= 0	inversion successful
	= 1	overflow occurred
	= 2	matrix is singular
	= 3	scaled inverse cannot be rescaled without causing overflow
	= 4	rows and columns cannot be rearranged
	= 10 + I	error code 3 occurred subsequent to error code 1
	= 20 + I	error code 4 occurred subsequent to error code 1

IRR2 = Rank if IRR1 = 2
= Reduction stage during which overflow occurred if IRR1 = 1
= 0 for all other values of IRR1

SCALE = Scale factor if IRR1 = 3
= 0 for all other values of IRR1

If IRR1 = 1 or 2, then the matrix will be rearranged and rescaled as though the inversion were successful, and the determinant will be set to zero.

RESTRICTIONS:

A. These must apply to the two routines:

1. They must be used on the IBM 7094 only.
2. The matrix is always destroyed during the inversion process, since the inverse is stored over the matrix.
3. Because of the internal scaling process, some matrix elements will be set to zero if the ratio of the largest to smallest magnitudes exceeds 10^{38}.
4. If underflow occurs, a normal zero is stored for the answer. Overflow always results in an error return.
5. The machine is left in single tag mode and floating point trap mode upon return to the calling program.

B. These apply to SINVRT only:

1. The inversion is performed by INV4S.
2. The matrix to be inverted must be stored in normal FORTRAN IV order and in single precision.

3. During the inversion process a total of $n^2 + 2n + 3$ locations must be available starting at the matrix (1, 1) element and ending at a lower core location.

C. This applies to INV4S only:

The total storage required is $n^2 + 2n + 3$.

STORAGE:

A. SINVRT $315_{10} = 473_8$

B. INV4S $489_{10} = 751_8$

TIMING:

For INV4S and SINVRT, the time in seconds to invert a matrix of order n is given by:

$$\text{Time} = 4.0 \times 10^{-5} \times n^3 \text{ seconds}$$

CHECKOUT:

All of the routines were tested on various-order Hilbert matrices and their inverses. Below is a table of the results.

Hilbert Matrix Order	Minimum Number of Significant Figures in Inverse (INV4S/SINVRT)
2	8
3	7
4	5
5	4
6	3

Also tested was the PEI matrix (Comm. A.C.M. V.5, 1962, page 508) with various diagonal terms and various orders. All input was single precision.

Order of PEI Matrix	Diagonal Term	Minimum Number of Significant Figures in Inverse (SINVRT)
40	1.001	5
60	1.001	6
65	1.00001	3
100	1.001	5
100	1.0001	4
100	1.00001	3
100	1.000001	2
100	1.0000001	2
100	1.5	7
100	1000.	7

Order of PEI Matrix	Diagonal Term	Minimum Number of Significant Figures in Inverse (SINVRT)
130	1000.	6
130	1000.	6

RECOMMENDED USAGE:

Below is a portion of a FORTRAN IV program which uses SINVRT acceptably.

```
      DIMENSION   A(60, 60), B(123), D(3723)
      EQUIVALENCE (A, C), (B, C(3601))
      .
      .
      .
      READ (5, 8000) N, ((A(I, J), J=1, N), I=1, N)
      CALL SINVRT (A, 60, N, IRR1, IRR2, SCALE, DET,
                   NDETXP)
      IF (IRR1) 6000,20,6000
   20
      .
      .
 6000 WRITE (6, 9000)IRR1, IRR2, SCALE

      CALL EXIT

 8000 FORMAT  (I10/(7F10.0))
 9000 FORMAT  (1H1,5X, 18H ERROR CODE 1 = I5, 14H
               ERROR CODE 2=I5, 14H SCALE FACTOR =
               E16.8)
```

The array C has $N^2 + 2N + 3 = 3723$ cells to ensure sufficient core to invert A. An extra 123 locations are then available in B for use at other times.

5.3 Appendix C — Tape Formats

Format 1 — Geometry definition tape format

This tape, written by the Geometry Definition section and read by the Transformation section, contains 18 binary records in the first file. Odd-numbered records have 10 fixed-point words; even-numbered records have a variable number of floating-point words. The tape symbol is LA in subroutine OPCAMI, LTAPE in subroutine DEFEN1 (which writes the tape), and LA in subroutine TFLAT (which reads the tape).

Record 1: Body Stations

Word	1	= 1
	2	Number of words in record 2.
	3	Code* (appears after record 18, at the end of this section).
	4 - 10	Not used.

Record 2: Body Stations

Word	1	First body station.
	2	Second body station.
	.	
	.	
	.	
	n	Last body station.

Record 3: Basic Body Meridian Lines

Word	1	= 3
	2	Number of words in record 4.
	3	Code*.
	4	Number of meridian lines.
	5 - 10	Not used.

Record 4: Basic Body Meridian Lines

Word	1	θ	First meridian line.
	2	Number of points in meridian line.	
	3	Word number of first point in meridian line.	
		Repeat for each additional meridian line.	
	i	X_1	Coordinates of first point in first meridian line.
	i + 1	Y_1	
	i + 2	Z_1	
		Repeat for each additional point in first meridian line and continue this way for each additional meridian line.	

Record 5: Wing Planform (Leading and Trailing Edge Points for Each Control Airfoil or Chord)

Word	1	= 5
	2	Number of words in record 6.
	3	Code*.
	4	Number of control airfoils or chords.
	5 - 10	Not used.

Record 6: Wing Planform (Leading and Trailing Edge Points for Each Control Airfoil or Chord).

Word	1	X_1	Coordinates of leading-edge point of first control airfoil or chord.
	2	Y_1	
	3	X_2	Coordinates of trailing-edge point of first control airfoil or chord.
	4	Y_2	
		Follow with leading- and trailing-edge points of remaining control airfoils or chords.	

Record 7: Basic Upper Wing Percent Chord Lines

Word	1	= 7
	2	Number of words in record 8.
	3	Code*.
	4	Number of percent chord lines.
	5 - 10	Not used.

Record 8: Basic Upper Wing Percent Chord Lines

Word	1	Percent.	First percent chord line.
	2	Number of points in percent chord line.	
	3	Word number of first point in percent chord line.	

Repeat for each additional percent chord line and follow with the coordinates of each point in each percent chord line.

NOTE: The format of record 8 is identical to that of record 4 except that words 1, 4, 7, ..., 3n - 2 contain percent instead of θ, where n = number of percent chord lines.

Record 9: Basic Lower Wing Percent Chord Lines

Word	1	= 9
	2	Number of words in record 10.
	3	Code*.
	4	Number of percent chord lines.
	5 - 10	Not used.

Record 10: Basic Lower Wing Percent Chord Lines

Same format as record 8.

Record 11: Upper Wing-Body Intersection Points

Word	1	= 11
	2	Number of words in record 12.
	3	Code*.
	4	Number of intersection points of upper wing percent chord lines with body.
	5 - 10	Not used.

Record 12: Upper Wing-Body Intersection Points

Word	1	X_1	Coordinates of the intersection of the first upper wing percent chord line with the body.
	2	Y_1	
	3	Z_1	

Repeat for each additional intersection point of the upper wing.

Record 13: Upper Wing Percent Chord Lines (Outside Body)

Word	1	= 13
	2	Number of words in record 14.
	3	Code*.
	4	Number of percent chord lines.
	5 - 10	Not used.

Record 14: Upper Wing Percent Chord Lines (Outside Body)

Same format as record 8.

Record 15: Lower Wing-Body Intersection Points

Word	1	= 15
	2	Number of words in record 16.
	3	Code*.
	4	Number of intersection points of lower wing chord percent lines with body.
	5 - 10	Not used.

Record 16: Lower Wing-Body Intersection Points

Word	1	X_1
	2	Y_1
	3	Z_1

Coordinates of the intersection of the first lower wing percent chord line with the body.

Repeat for each additional intersection point of the lower wing.

Record 17: Lower Wing Percent Chord Lines (Outside Body)

Word	1	= 17
	2	Number of words in record 18.
	3	Code*.
	4	Number of percent chord lines
	5 - 10	Not used.

Record 18: Lower Wing Percent Chord Lines (Outside Body)

Same format as record 8.

---

* CODE	= 0	If this and the next record have been successfully processed.
	< 0	An error occurred and the next record is a dummy (all zeros).
	> 0	This option was not requested; the next record is a dummy (all zeros).

Format 2 — Geometry transformation tape format

This tape, written by the Geometry Transformation section and read by the Geometry Paneling section of the program, contains 6 binary records in the first file. Odd-numbered records have 10 fixed-point words; even-numbered records have a variable number of floating-point words. The tape symbol is LD in subroutine OPCAMI, LD in subroutine TFLAT (which writes the tape), and ND4 in subroutines INPUTB and INPUTW (which read the tape).

Record 1: Body Meridian Lines

Word	1	= 1
	2	Number of words in record 2.
	3	Code* (appears after record 6, at the end of this section).
	4	Number of meridian lines.
	5 - 10	Not used.

Record 2: Body Meridian Lines

Word	1	θ	First meridian line.
	2	Number of points in meridian line.	First meridian line.
	3	Word number of first point in meridian line	First meridian line.
		Repeat for each additional meridian line.	
	i	X_1	Coordinates of first point in first meridian line.
	i + 1	Y_1	Coordinates of first point in first meridian line.
	i + 2	Z_1	Coordinates of first point in first meridian line.
		Repeat for each additional point in first meridian line and continue this way for each additional meridian line.	

Record 3: Wing Percent Chord Lines

Word	1	= 3
	2	Number of words in record 4.
	3	Code*.
	4	Number of percent chord lines.
	5 - 10	Not used.

Record 4: Wing Percent Chord Lines

Word 1	Percent.	First percent chord line.
2	Number of points in percent chord line.	
3	Word number of first point in percent chord line.	

Repeat for each additional percent chord line and follow with the coordinates of each point in each percent chord line.

NOTE: The format of record 4 is identical to that of record 2 except that words 1, 4, 7, ..., 3n - 2 contain percent instead of θ, where n = number of percent chord lines.

Record 5: Wing-Body Intersection Points

Word 1	= 5
2	Number of words in record 6.
3	Code*.
4	Number of intersection points of wing percent chord lines with body.
5 - 10	Not used.

Record 6: Wing-Body Intersection Points

Word 1	X_1	Coordinates of the intersection of the first wing percent chord line with the body.
2	Y_1	
3	Z_1	

Repeat for each additional intersection point.

---

* CODE =	0	If this and the next record have been successfully processed.
<	0	An error occurred and the next record is a dummy (all zeros).
>	0	This option was not requested; the next record is a dummy (all zeros).

Format 3 — Transformation data tape format

This tape, containing three records in the first file, is written by the Geometry Transformation section of the program; both the Geometry Paneling and Aerodynamics sections read the tape. The tape symbol is LB in subroutine OPCAMI, LB in subroutine TFLAT (which writes the tape), ND1 in subroutine INPUTB (which reads only the first record), and NTAPEB in subroutine INTAPE (which reads all records).

Record 1

Word	1	x	Components of a unit vector (in original coordinate system) along the x axis of the new coordinate system.
	2	y	
	3	z	
	4	z_A	z coordinate of the transformed and flattened wing.
	5	α	An original body station X is related to the new body station x by $x = \alpha X + \beta$.
	6	β	

Record 2

Word	1	NRX	Number of triplets (x_i, r_i, z_i) in record 3.
	2	INDEX	For INDEX = j. The triplet (x_i, r_i, z_i) corresponds to the intersection of the leading edge of the wing with the body surface (see record 3).

Record 3

After the body has been transformed to a new coordinate system such that the new x axis passes through the centers of body ends, the body radius and centroid is found at a number of body stations. Let x = body station, r = body radius, and z = z-coordinate of body centroid.

Word	1	x	x, r and z at first body station.
	2	r	
	3	z	

Repeat until the record contains NRX triplets.

Format 4 — Geometry Paneling tape format

This tape is written by the Geometry Paneling section for use in the Aerodynamics section. The tape symbol NTAPE2 is used in the Geometry Paneling section and NTAPEC is used in the Aerodynamic section. The sample format given below is that used for a case requiring both body and wing paneling. It should be noted that parts 1, 2, and 3 are deleted for a case that involves only wing paneling and that the tape is not written for a body-alone case. Body paneling data are written in parts 1, 2, and 3 and data for wing paneling in parts 4, 5, and 6; panel corner point coordinates are output in parts 1 and 4 and the additional geometry in parts 2 and 5. One-, two-, and three-records are written in parts 1 and 4 for panels of one, two, and three parts respectively. For these two tape parts, the initial record of each record set always has the format of record 1 as given below and the remaining records of the set have the format described for record 2. A single record is written per panel in parts 2 and 5.

Part 1

Record 1

Word	1	Body panel number.
	2	Number of body panel parts.
	3, 4, 5	Coordinates of inboard leading-edge corner point (first panel part).
	6, 7, 8	Coordinates of outboard leading-edge corner point (first panel part).
	9, 10, 11	Coordinates of inboard trailing-edge corner point (first panel part).

12, 13, 14	Coordinates of outboard trailing-edge corner point (first panel part).

Record 2

Word 1, 2, 3	Coordinates of inboard leading-edge corner point (second panel part).
4, 5, 6	Coordinates of outboard leading-edge corner point (second panel part).
7, 8, 9	Coordinates of inboard trailing-edge corner point (second panel part).
10, 11, 12	Coordinates of outboard trailing-edge corner point (second panel part).

These records are repeated for additional body panels as indicated above.

Part 2

Record 1

Word 1	Body panel number.
2, 3, 4	Coordinates of body panel centroid.
5, 6, 7	Coordinates of body panel control point.
8	Body panel area.
9	Body panel θ -inclination angle.

	10	Body panel α -incidence angle.
	11	Body panel (streamwise) chord length.
		This single record is repeated for each additional body panel.

Part 3

Record 1

Word	1	Number of body panels per column.
	2	Fractional value (XPER) used to calculate streamwise location of panel control point (see subroutine PANEL).
		This is the only record written for this part.

Part 4

Record 1

Word	1	Wing panel number.
	2	Number of wing panel parts.
	3, 4, 5	Coordinates of inboard leading-edge corner point (first panel part).
	6, 7, 8	Coordinates of outboard leading-edge corner point (first panel part).
	9, 10, 11	Coordinates of inboard trailing-edge corner point (first panel part).
	12, 13, 14	Coordinates of outboard trailing-edge corner point (first panel part).

Record 2

Word	1, 2, 3	Coordinates of inboard leading-edge corner point (second panel part).

4, 5, 6	Coordinates of outboard leading-edge corner point (second panel part).
7, 8, 9	Coordinates of inboard trailing-edge corner point (second panel part).
10, 11, 12	Coordinates of outboard trailing-edge corner point (second panel part).

Record 1 <u>or</u> records 1 <u>and</u> 2 are repeated for additional one- or two-part wing panels as indicated above.

<u>Part 5</u>

Record 1

Word 1	Wing panel number.
2, 3, 4	Coordinates of wing panel centroid.
5, 6, 7	Coordinates of wing panel control point.
8	Wing panel area.
9	Wing panel thickness slope.
10	Wing panel camber slope.
11	Wing panel (streamwise) chord length.

This one record is repeated for each additional wing panel.

<u>Part 6</u>

Record 1

Word 1	Number of wing panels per column.
2	Fractional value (XPER) used to calculate streamwise location of panel control point (see subroutine PANEL). This is the only record written for this part.

Appendix D — Subroutine Listings

Listings of the subroutines in the following index are given in this appendix.

Subroutine Index

Subroutine	Page No.	Subroutine	Page No.
AERO	455	CVEL	508
ALPHAB	456	DCPD	509
AMATE	457	DCPI	510
AREAP	458	DEFEN1	511
ARFREE	459	DEFEND	512
ARFUL	461	DISPTA	515
ARSECR	462	DMAXL	516
BCUTX	464	ELLIPR	517
BITURP	465	ENRYCH	518
BLDM	467	EVAL	522
BODCR	468	FASNCS	529
BODY	471	FATN1	533
BODY1	473	FDATE	538
BODY1M	476	FFSF	541
BODY1R	478	FFSR	545
BODY1S	480	FORCES	548
BSCALE	482	FROOTA	561
CAMBW	483	FTAN	564
CAMBWB	484	FVIO	568
CEGAR	485	GEOMD	570
CENTRD	486	IADARY	572
CHECK	487	IBFR	573
CHORDW	488	ICODE	575
CLEAR	490	IDLETE	577
CLOK	492	INIBFR	578
CNTRLB	493	INOUT	580
CNTRLW	495	INPUTB	582
CP	497	INPUTW	587
CRNRB	498	INTAPE	591
CRNRW	503	INTPOL	595

Subroutine	Page No.	Subroutine	Page No.
INTURP	596	TDUMP	673
INV4S	597	TFLAT	674
INVBB	611	TFLAT1	683
INVRW	612	TFLATM	684
INVW	613	TFLATW	686
IPACK	614	TFLATX	688
IRLEAS	615	THETAB	689
IRSERV	616	TRAPCT	690
KARMOR	617	TRAV	691
LENG	621	TROTPT	692
MDMATE	622	TVEL	693
MEAN	624	UFOOL	695
MERR	625	UN07	696
NWEED	626	UN08	697
OPCAM	627	UN09	698
OPCAMI	629	UPDATE	699
OPTIM3	630	UVECN	701
OPTMW	632	VCROS	702
OPTMWB	633	VDOTN	703
OUTB	634	VDOTP	704
OUTPTB	635	WBX	705
OUTPTW	640	WBXA	710
OUTW	646	WBXC	711
PANEL	647	WBXD	712
PARTV	648	WBXUL	714
POLXN	649	WBXX	717
QRAT	651	WING	722
READ	652	WING1	724
REDUCE	653	WING1A	728
RICH3A	655	WING1B	731
RICHNA	658	WING2	733
RITE	660	WING2P	735
SINVRT	662	WINGA	736
SLOPEW	664	WLDM	737

```
      SUBROUTINE AERO

C     CONTROL ROUTINE FOR AERODYNAMIC LINKS

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEC,NBODY,NWING,XMACH,SYM,KACE

      CALL AMATE

C     KACE = 1      WING ALONE
C     KACE = 2      BODY ALONE
C     KACE = 3      WING-BODY COMBINATION
      GO TO (200,500,300),KACE
  200 CALL INVW
      GO TO 400
  300 CONTINUE
      CALL INVBB
      CALL REDUCE
      CALL INVRW
      CALL PARTV
  400 CALL MOMATE
  500 CALL FORCES

      RETURN
      END
```

```
      SUBROUTINE ALPHAB
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/ NPLNB,NPLNW,JLEAD,JTRAIL,IMID,NPTS(16),X(16,90),Y(16
     1,90),Z(16,90),XCEPT(21),XCEPTB(21),XCEPTW(16),YCEPTW(16),ZCEPTW(16
     2),CODEBW(16),KPANEL(15,20),XCOR(16,21),YCOR(16,21),ZCOR(16,21),XIN
     3T(15,20,2),YINT(15,20,2),ZINT(15,20,2),XCEN(15,20),YCEN(15,20),ZCE
     4N(15,20),XCON(15,20),YCON(15,20),ZCON(15,20),AREA(15,20),ARAT(15,2
     50),THETA(15,20),ALPHA(15,20),CHORD(15,20)
      DIMENSION C(4),P(3,50),Q(3,5),LOC(3)

C     SUBROUTINE TO CALCULATE BODY PANEL ALPHA
C     ANGLES

C     DEFINE FUNCTION TO BE USED IN CALCULATIONS
C     ANGL CALCULATES ALPHA INCIDENCE ANGLES USING
C     ARCTANGENT ROUTINE
      ANGL(X1,X2,Y1,Y2,Z1,Z2,A)=ATAN(((Z2-Z1)*COS(A)-
     1(Y2-Y1)*SIN(A))/(X2-X1))

C     CALCULATE PANEL ALPHA ANGLES
 80   DO 90 I=1,NPER1                                                  ,1
      DO 87 J=1,NPLN1                                                  ,2
      KGO=KPANEL(I,J)                                                  ,3
      GO TO (83,85,83,85),KGO                                          ,4
 83   ALPHA(I,J)=ANGL(XCOR(I,J),XINT(I,J,1),YCOR(I,J),YINT(I,J,1),
     1ZCOR(I,J),ZINT(I,J,1),THETA(I,J))                                ,5
      GO TO 87                                                         ,6
 85   ALPHA(I,J)=ANGL(XINT(I,J,1),XCOR(I,J+1),YINT(I,J,1),YCOR(I,J+1),
     1ZINT(I,J,1),ZCOR(I,J+1),THETA(I,J))                              ,7
 87   CONTINUE                                                         ,8     ,9
   90 CONTINUE                                                         ,10    ,11
 100  RETURN                                                           ,12
       END                                                             ,13
```

```
      SUBROUTINE AMATE

C     CONTROL ROUTINE FOR COMPUTING AERODYNAMIC INFLUENCE COEFFICIENTS
C     MATRIX AND VELOCITY COMPONENTS

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEC,NBODY,NWING,XMACH,SYM,KACE

      COMMON /BLOCK/X(200,3,4),Y(200,3,4),Z(200,3,4),XBAR(200)
     1,YBAR(200),ZBAR(200),XC(200),YC(200),ZC(200),THETA(200),ALPHAS(200
     2),AREA(200),CHORD(200),NPART(200),NPANEL,ISYM,A(200),U(200),V(200)
     3,VPM(200),VV(200),W(200),WPM(200),WW(200),VPMM(200),WPMM(200),ALPH
     1AC(100),ALPHAT(100)


C     READ IN GEOMETRICAL DATA
      CALL INTAPE (THKW)

C     KACE = 1      WING ALONE
C     KACE = 2      BODY ALONE
C     KACE = 3      WING-BODY COMBINATION                                ,1
      GO TO (150,300,150),KACE                                           ,2
 150  IF (THKW) 200,250,200

C     COMPUTE VELOCITY COMPONENTS DUE TO WING SOURCES                    ,3
 200  NS=NBODY+1                                                         ,4
      CALL EVAL (NS,XBAR,1.)                                             ,5
      CALL TVEL

C     COMPUTE AERODYNAMIC INFLUENCE COEFFICIENTS MATRIX
C     AND VELOCITY COMPONENTS                                            ,6
 250  NS=1                                                               ,7
      CALL EVAL (NS,XC,0.)
                                                                         ,8
 300  REWIND NTAPEC
                                                                         ,9
      RETURN                                                             ,10
      END                                                                ,11
```

```
      SUBROUTINE AREAP(X,Y,Z,S,R,NC,NR)
      DIMENSION X(16,16),Y(16,16),Z(16,16),S(15,15),R(15,15)
      DIMENSION A(3),B(3),V(3)

C     SUBROUTINE TO CALCULATE BODY AND WING PANEL
C     AREAS

C     CALCULATE PANEL AREAS
 50   DO 60 I=1,NC
      DO 60 J=1,NR                                                  ,1
      A(1)=X(I,J)-X(I,J+1)                                          ,2
      A(2)=Y(I,J)-Y(I,J+1)                                          ,3
      A(3)=Z(I,J)-Z(I,J+1)                                          ,4
      B(1)=X(I+1,J+1)-X(I,J+1)                                      ,5
      B(2)=Y(I+1,J+1)-Y(I,J+1)                                      ,6
      B(3)=Z(I+1,J+1)-Z(I,J+1)                                      ,7
      CALL VCROS(A,B,V,D,K)                                         ,8
      S1=D/2.                                                       ,9
      A(1)=X(I,J)-X(I+1,J)                                          ,10
      A(2)=Y(I,J)-Y(I+1,J)                                          ,11
      A(3)=Z(I,J)-Z(I+1,J)                                          ,12
      B(1)=X(I+1,J+1)-X(I+1,J)                                      ,13
      B(2)=Y(I+1,J+1)-Y(I+1,J)                                      ,14
      B(3)=Z(I+1,J+1)-Z(I+1,J)                                      ,15
      CALL VCROS(A,B,V,D,K)                                         ,16
      S2=D/2.                                                       ,17
      S(I,J)=S1+S2                                                  ,18
 60   R(I,J)=S1/S(I,J)                                              ,19
 100  RETURN                                                        ,20   ,21   ,22
       END                                                          ,23
                                                                    ,24
```

```
ARFREE                                                                    07/31/65
7094 RELMOD ASSEMBLY.

        $IBLDR ARFREE                                                                  ARFR0000
ASSEMBLED TEXT.

        $TEXT  ARFREE                                                                  ARFR0001

                                  *ARFREE     MICHAEL SYNGE          6-9247      ASSEMBLED   26 JAN 65
                                  *     TO FORM THE FIRST CODE WORD OF AN EMPTY ARRAY.
                                  *
                                  *              CALLING SEQUENCE
                                  *
                                  *                  BUF(M-2)=ARFRE(M)
                                  *
                                  *              WHERE  BUF(M)  IS THE FIRST CELL OF THE ARRAY
                                  *
                                         ENTRY   ARFRE
                                   IRFRE ENTRY   ARFRE
                                  *

BINARY CARD (NOT PUNCHED)
      00000  0500 60 4 00003  10000   ARFRE CLA*    3,4
      00001  0361 00 0 00006  10001         ACL     =0700000
      00002  0602 00 0 00005  10001         SLW     TEMP
      00003  0500 00 0 00005  10001         CLA     TEMP
      00004  0020 00 4 00001  10000         TRA     1,4
                                  *
      00005  0 00000 0 00000  10000   TEMP
      00006  000000700000     10000         *LORG
                       00000  01111         END
         CONTROL DICTIONARY

        $CDICT ARFREE                                                                  ARFR0002

BINARY CARD (NOT PUNCHED)
            000007000000              PREFACE     START=0,LENGTH=7,TYPE=7094,CMPLX=5
            000004000005
            215126512525       ARFREE DECK        LOC=0,LENGTH=7
            000007000000
            215126512560       ARFRE  REAL        LOC=0,LENGTH=0
            000000000000
            315126512560       IRFRE  REAL        LOC=0,LENGTH=0
            000000000000

        $DKEND ARFREE                                                                  ARFR0003

NO MESSAGES FOR THIS ASSEMBLY
```

ARFREE 07/31/65

SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

CLASS	SYMBOL	VALUE	REFERENCES
	ARFRE	00000	
LCTR	BLCTR		
QUAL	UNQS		
LCTR	//		
	TEMP	00005	2,3

```
          ARFUL                                                                  07/31/65
        7C94 RELMOD ASSEMBLY.

                $IBLDR ARFUL                                                                  ARFU0000
        ASSEMBLED TEXT.

                $TEXT  ARFUL                                                                  ARFU0001

                                        *ARFUL      MICHAEL SYNGE          6-9247      ASSEMBLED  21 AUG 64
                                        *     TO FORM THE FIRST CODE WORD OF A FULL ARRAY.
                                        *
                                        *              CALLING SEQUENCE
                                        *
                                        *                  BUF(M-2)=ARFUL(M)
                                        *
                                        *              WHERE  BUF(M)  IS THE FIRST CELL OF THE ARRAY.
                                        *
                                              ENTRY   ARFUL
                                        *

BINARY CARD (NOT PUNCHED)
      00000  0500 60 4 00003  10000   ARFUL CLA*    3,4
      00001  0361 00 0 00006  10001         ACL     =0700000700000
      00002  0602 00 0 00005  10001         SLW     TEMP
      00003  0500 00 0 00005  10001         CLA     TEMP
      00004  0020 00 4 00001  10000         TRA     1,4
      00005  0 00000 0 00000  10000   TEMP
      00006  700000700000     10000         *LORG
                      00000  01111         END
        CONTROL DICTIONARY

                $CDICT ARFUL                                                                  ARFU0002

BINARY CARD (NOT PUNCHED)
            000007000000              PREFACE     START=0,LENGTH=7,TYPE=7094,CMPLX=5
            000004000005
            215126644360       ARFUL  DECK        LOC=0,LENGTH=7
            000007000000
            215126644360       ARFUL  REAL        LOC=0,LENGTH=0
            000000000000

                $DKEND ARFUL                                                                  ARFU0003

NO MESSAGES FOR THIS ASSEMBLY
        SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

  CLASS  SYMBOL   VALUE   REFERENCES

         ARFUL    00000
  LCTR    BLCTR
  QUAL    UNQS
  LCTR   //
         TEMP     00005   2,3
```

```
ARSECR                                                                07/31/65
094 RELMOD ASSEMBLY.

        $IBLDR ARSECR                                                                  ARSE0000
ASSEMBLED TEXT.

        $TEXT  ARSECR                                                                  ARSE0001

                                *ARSECR     MICHAEL SYNGE          6-9247      ASSEMBLED   26 JAN 65
                                *     TO FORM THE SECOND CODE WORD OF A FULL ARRAY.
                                *
                                *              CALLING SEQUENCE
                                *
                                *                  BUF(M-1)=ARSEC(N,NAME)
                                *
                                *              WHERE    N    = THE LENGTH OF THE ARRAY
                                *                       NAME = THE NAME OF THE ARRAY
                                *
                                       ENTRY   ARSEC
                                 IRSEC ENTRY   ARSEC
                                *

BINARY CARD (NOT PUNCHED)
      00000  0500 60 4 00003  10000   ARSEC CLA*    3,4
      00001  0767 00 0 00022  10000         ALS     18
      00002  0600 00 0 00010  10001         STZ     TEMP
      00003  0622 00 0 00010  10001         STD     TEMP
      00004  0500 00 4 00004  10000         CLA     4,4
      00005  0621 00 0 00010  10001         STA     TEMP
      00006  0500 00 0 00010  10001         CLA     TEMP
      00007  0020 00 4 00001  10000         TRA     1,4
                                *
      00010  0 00000 0 00000  10000   TEMP
                      00000  01111         END
         CONTROL DICTIONARY

        $CDICT ARSECR                                                                  ARSE0002

BINARY CARD (NOT PUNCHED)
            000011000000              PREFACE     START=0,LENGTH=9,TYPE=7094,CMPLX=5
            000004000005
            215162252351       ARSECR DECK        LOC=0,LENGTH=9
            000011000000
            215162252360       ARSEC  REAL        LOC=0,LENGTH=0
            000000000000
            315162252360       IRSEC  REAL        LOC=0,LENGTH=0
            000000000000

        $DKEND ARSECR                                                                  ARSE0003

NO MESSAGES FOR THIS ASSEMBLY
```

ARSECP
SYMBOL REFERENCE DATA

07/31/65

REFERENCES TO DEFINED SYMBOLS.

CLASS	SYMBOL	VALUE	REFERENCES
	ARSEC	00000	
LCTR	BLCTR		
QUAL	UNOS		
LCTR	//		
	TEMP	00010	2,3,5,6

```
      SUBROUTINE BCUTX(B,N,X,EP,NA,A)
C
      DIMENSION B(1),C(4),A(2,1),LQ(2),Q(3)
C
C     USED BY WBXC TO CUT THE BODY WITH A STATION PLANE
C     TO GET A 2-DIM. ARRAY OF PTS. (A) IN THE PLANE.
C         A = (Y1,Z1,Y2,Z2,.....)
C     NA = NO. PTS. IN A.
C     INPUTS ARE B,N,X,EP.  OUTPUTS ARE NA,A
      DATA C/-1.,3*0./
C
      J=0                                                                ,1
      C(4)=X                                                             ,2
 100  DO 1000 K=1,N                                                      ,3
      K1=3*K                                                             ,4
      K2=B(K1-1)                                                         ,5
      K3=B(K1)                                                           ,6
      CALL POLXN(C,B(K3),3,K2,EP,1,Q,LQ,L)                               ,7
      IF (L .NE. 1) GO TO 1000                                           ,8    ,9    ,10
      J=J+1                                                              ,11
      A(1,J)=Q(2)                                                        ,12
      A(2,J)=Q(3)
C                                                                        ,13
 1000 CONTINUE                                                           ,14   ,15
      NA=J                                                               ,16
 9000 RETURN                                                             ,17
      END                                                                ,18
```

```
      SUBROUTINE BITURP(X,Y,IXY,NXY,KD,XI,YI,NU)
      DIMENSION X(1),Y(1),NU(3),C(4),YIQ(2)
C
C     GIVEN NXY POINTS AND X=XI,INTERPOLATE TO FIND Y=YI.
C     X AND Y VALUES SPACED IXY APART.  IXY=1 IF CONSECUTIVE.
C     KD=1 IF LINEAR INTERPOLATION, KD=2 IF BIQUADRATIC.
C
      LOGICAL  LINT,LIF,LIL
C
      NU=0                                                              ,1
      LINT=.FALSE.                                                      ,2
      LIF=.FALSE.                                                       ,3
      LIL=.FALSE.
C
  100 IF (IXY)110,110,200                                               ,4
  110 NU=-1                                                             ,5
      GO TO 5000                                                        ,6
  200 IF (NXY-2)210,220,300                                             ,7
  210 NU=-2                                                             ,8
      GO TO 3200                                                        ,9
C     USE LINEAR INTERPOLATION IF ONLY 2 POINTS.
  220 LINT=.TRUE.                                                       ,10
C
  300 IX=IXY*(NXY-1)+1                                                  ,11
      JXY=IXY                                                           ,12
C
  400 IF (X(IX)-X)420,410,430                                           ,13
  410 NU=-3                                                             ,14
      GO TO 3200                                                        ,15
C
C     X-VALUES OF POINTS IN DECREASING ORDER.
C     WE WILL SEARCH FROM RIGHT TO LEFT INSTEAD OF
C       LEFT TO RIGHT.
  420 IA=IX+IXY                                                         ,16
      JXY=-JXY                                                          ,17
      GO TO 500                                                         ,18
C
  430 IA=1-IXY                                                          ,19
C     SEARCH FOR INTERVAL CONTAINING XI
  500 DO 600 I=1,NXY                                                    ,20
      IA=IA+JXY                                                         ,21
  510 IF (XI-X(IA))700,800,600                                          ,22
  600 CONTINUE                                                          ,23
  610 NU=2                                                              ,24   ,25
      GO TO 800                                                         ,26
C
  700 IF (I-2)710,900,1000                                              ,27
  710 NU=1                                                              ,28
C
C     XI FALLS EXACTLY ON A POINT OR IS OUTSIDE THE TABLE.
  800 YI=Y(IA)                                                          ,29
      GO TO 3100                                                        ,30
C
```

```
C     XI IN 1ST INTERVAL (OR LAST IF GOING FROM RIGHT)                    ,31
 900  LIF=.TRUE.
C                                                                         ,32
 1000 IF (I-NXY)1100,1010,1010
C     XI IN LAST INTERVAL (OR 1ST)                                        ,33
 1010 LIL=.TRUE.
C                                                                         ,34
 1100 I2=IA-JXY                                                           ,35
 1200 IF (KD-2)3000,1250,1250                                             ,36
 1250 IF (LINT) GO TO 3000
C
C     QUADRATIC OR BIQUADRATIC INTERPOLATION                              ,37   ,38   ,39
 1300 JA=IA                                                               ,40
 1400 DO 1800 IZ=1,2                                                      ,41
      JA=JA-JXY                                                           ,42
 1500 IF ((IZ.EQ.1 .AND.LIL) .OR. (IZ.EQ.2 .AND.LIF))GO TO 1800           ,43   ,44   ,45
 1600 CALL ORAT(X(JA),Y(JA),JXY,1,XI,YIQ(IZ),C,MU)                        ,46
      YI=YIQ(IZ)                                                          ,47
 1700 IF (MU)1710,1800,1710                                               ,48
 1710 NU=IZ+2
C     TWO POINTS MUST HAVE SAME X                                         ,49
      GO TO 3000
C                                                                         ,50
 1800 CONTINUE
C                                                                         ,51   ,52
 1900 IF (LIF.OR.LIL) GO TO 9000
C     BIQUADRATIC INTERPOLATION                                           ,53   ,54   ,55
 2000 KXY=JXY+JXY                                                         ,56
      I1=IA-KXY                                                           ,57
      I4=IA+JXY                                                           ,58
      YI=((X(IA)-XI)*YIQ+(XI-X(I2))*YIQ(2))/(X(IA)-X(I2))                 ,59
      GO TO 9000
C
C     LINEAR INTERPOLATION                                                ,60
 3000 YI=Y(I2)+(XI-X(I2))*((Y(IA)-Y(I2))/(X(IA)-X(I2)))
C                                                                         ,61
 3100 IF (NU)3200,9000,3200
C
C     ERROR.  REDUCE ERROR COUNTER AND WRITE MESSAGE                      ,62
 3200 LTAPE=NU(2)                                                         ,63
      NU(3)=NU(3)-1                                                       ,64
 3300 IF (NU(3).LE.0 .OR. LTAPE.LE.0) GO TO 9000                          ,65   ,66   ,67
 3400 WRITE (LTAPE,3410)NU(1),XI,YI,NXY,(X(I),Y(I),I=1,IX,IXY)            ,68   ,69   ,70   ,71   ,72   ,73

 3410 FORMAT(13HOBITURP ERROR I3,8H FOR X = F14.6,1H.14H  RETURNED Y =
     .F14.6,I6,17H POINTS (X,Y) ARE/(1H 8F14.6))
C
 9000 RETURN                                                              ,74
      END                                                                 ,75
```

```
      SUBROUTINE BLDM (NM,XP,ZP,RFAREA,AREA,XBAR,ZBAR,ALPHAM,THETAM
     1,CPM,CL,CD,CM)

C     COMPUTES COEFFICIENT OF LIFT, DRAG, AND MOMENT ON BODY

      DIMENSION AREA(1),XBAR(1),ZBAR(1),ALPHAM(1),THETAM(1),CPM(1)

      DRAG=0.                                                          ,1
      XLIFT=0.                                                         ,2
      XM=0.0
      DO 100 J=1,NM                                                    ,3
      F=CPM(J)*AREA(J)                                                 ,4
      XL=-F*COS(THETAM(J))                                             ,5
      XD=F*ALPHAM(J)                                                   ,6
      XM=XM-XL*(XBAR(J)-XP)+XD*(ZBAR(J)-ZP)                            ,7
      XLIFT=XLIFT+XL                                                   ,8
      DRAG=DRAG+XD                                                     ,9
  100 CONTINUE                                                         ,10
      CL=XLIFT/RFAREA                                                  ,11   ,12
      CD=DRAG/RFAREA                                                   ,13
      CM=XM/RFAREA                                                     ,14
      RETURN                                                           ,15
      END                                                              ,16
                                                                       ,17
```

```
      SUBROUTINE BODCR(B,N,X,EP,CHD,MAXE,S,E,NE,Y,Z,AREA,RAD,NU)
C
C     FINDS THE AREA,EQUIV. RADIUS, AND CENTROID OF A BODY SECTION.
C
      DIMENSION B(1),EP(5),S(2,N),E(MAXE),NU(3),CG(2)
C
C     THE BODY IS GIVEN AS A SET OF MERIDIAN LINES (SEE BODY1).
C       THE BODY IS SYMMETRICAL ABOUT A PLANE PARALLEL TO THE XZ PLANE.
C       WE HAVE THE MERIDIAN LINES FOR HALF THE BODY.
C     THE BODY SECTION IS THE INTERSECTION OF A STATION PLANE WITH THE BODY.
C    INPUTS ARE
C     B = BODY MERIDIAN LINE POINTS (WITH HEADER)
C     N = NO. OF MERIDIAN LINES
C     X = BODY STATION
C     EP = ARRAY OF TOLERANCES
C     CHD = CHORD HEIGHT TOLERANCE USED BY ENRYCH TO DETERMINE
C           SPACING OF INTERPOLATED POINTS BETWEEN MERIDIANS.
C     MAXE = LENGTH OF ARRAY E
C     NU(2) = TAPE NO. FOR ENRYCH ERROR MESSAGE
C     NU(3) = ERROR LIMITER AND COUNTER (ALSO OUTPUT)
C    OUTPUTS ARE
C     S = ARRAY OF N POINTS (Y,Z) IN SECTION
C     E = IS ESSENTIALLY A SCRATCH ARRAY FOR ENRICHED SECTION POINTS.
C         THE 1ST N CELLS ARE THE Y VALUES OF S.
C         THE NEXT N CELLS ARE THE Z VALUES OF S.
C         THE NEXT NE CELLS ARE THE DENSE ARRAY OF POINTS,
C         STORED (Y,Z,Y,Z,....) ON THE SECTION.
C       E IS NOT USED FOR DEGENERATE SECTIONS.
C       E IS USED BUT NO ENRICHING TAKES PLACE IF N .LT. 4 OR
C         IF CHD .LE. 0.
C     NE IS EXPLAINED UNDER E
C     Y = Y COORD. OF CENTROID OF WHOLE SECTION.
C     Z = Z COORD. OF CENTROID
C     AREA = AREA OF WHOLE SECTION
C     RAD = AVERAGE DISTANCE FROM CENTROID TO EACH MERIDIAN LINE.
C     NU = 0 IF SUCCESS
C          1 IF CUT THRU BODY DOES NOT INTERSECT ALL MERIDIAN LINES
C            OR ELSE N .LT. 2
C          2,3,4, OR 5 IF ENRYCH ERROR.
C          6 IF 1ST AND LAST PTS IN SECTION DO NOT HAVE SAME Y WITHIN EP(2)
C
      KEE = 0
      NU=1                                                              ,1
C     CUT BODY WITH PLANE X = X TO GET SECTION S OF NS POINTS           ,2
      CALL BCUTX(B,N,X,EP(1),NS,S)                                      ,3
      IF(NS .NE. N  .OR.  N .LT. 2) GO TO 8000                          ,4    ,5    ,6
      NU=6
C     CHECK THAT 1ST AND LAST PTS IN SECTION HAVE SAME Y VALUE          ,7
      IF(ABS(S(1,1)-S(1,N)) .GT. EP(2)) GO TO 8000                      ,8    ,9    ,10
      Y=(S(1,1)+S(1,N))/2.
C     CHECK FOR SECTION WITH ZERO AREA                                  ,11
      IF(ABS(S(2,1)-S(2,N)) .GT. EP(4)  .AND.  N .GT. 2) GO TO 500      ,12   ,13   ,14
      AREA=0.                                                           ,15
```

```
      GO TO 1500
C
C     PREPARE TO ENRICH SECTION.  WE MUST HAVE Y,Z IN SEPARATE
C     ARRAY SO USE 1ST PART OF E ARRAY
  500 DO 600 I=1,N                                                  ,16
      E(I)=S(1,I)                                                   ,17
      II=I+N                                                        ,18
  600 E(II)=S(2,I)                                                  ,19
C
      IE=II+1                                                       ,20   ,21
C     THE ENRICHED SECTION POINTS WILL START IN E(IE)
      NE=MAXE-II-2                                                  ,22
C     NE=AVAILABLE STORAGE FOR ENRICHED POINTS.  WE SUBTRACT 2 BECAUSE
C     CEGAR WILL LATER FILL IN 2 CELLS AT END OF ARRAY.
      CALL ENRYCH(E,E(N+1),N,CHD,E(IE),NE,NU)                       ,23
      IF(NU .EQ. 0) GO TO 1000                                      ,24
      KEE = 1                                                       ,25   ,26   ,27
      IF(NU .NE. 3) GO TO 8000                                      ,28
C     NOTE WE IGNORE ERROR 3 FROM ENRYCH, ASSUMING THAT CALLING PROGRAM
C     HAS SET NU(2), NU(3) SO THAT ENRYCH PRINTED ERROR MESSAGE.
C     IF ERROR FROM ENRYCH, EXPLANATORY MESSAGE WRITTEN AT END (KEE = 1)

C     FIND AREA, CENTROID OF ENRICHED SECTION
 1000 CALL CEGAR(NE/2,E(IE), AREA,CG)                               ,29   ,30   ,31
      AREA=AREA+AREA                                                ,32
C     IF AREA TOO SMALL, CG MAY BE INACCURATE
      IF(AREA .LE. EP(3)) GO TO 1500                                ,33
      Z=CG(2)                                                       ,34   ,35   ,36
      GO TO 2000                                                    ,37
C
 1500 CONTINUE                                                      ,38
C     FOR Z OF CENTROID, USE AVERAGE OF 1ST AND LAST POINTS.
      Z=(S(2,1)+S(2,N))/2.                                          ,39
C
      IF(AREA)1600,1600,2000                                        ,40
 1600 RAD=0.                                                        ,41
      GO TO 3000                                                    ,42
C
 2000 CONTINUE                                                      ,43

C     COMPUTE RADIUS OF SECTION
      SUMRAD=0.                                                     ,44
      DO 2500 I=1,NS                                                ,45
      RR=(S(1,I)-Y)**2  + (S(2,I)-Z)**2                             ,46
      R=SQRT(RR)                                                    ,47
      SUMRAD = SUMRAD + R                                           ,48
      IF(I .EQ. 1  .OR.  I .EQ. NS) GO TO 2500                      ,49
      SUMRAD = SUMRAD + R                                           ,50   ,51   ,52
 2500 CONTINUE                                                      ,53
      FNS=NS+NS-2                                                   ,54   ,55
      RAD=SUMRAD/FNS                                                ,56
 3000 CONTINUE                                                      ,57
      NU=0                                                          ,58
C                                                                   ,59
```

```
8000 CONTINUE                                                          ,60
     LTAPE=NU(2)                                                       ,61
     IF(.NOT.(NU .NE. 0  .AND.  LTAPE .GT. 0)) GO TO 9000              ,62   ,63   ,64
     WRITE(LTAPE,8100)NU(1),(J,(S(I,J),I=1,2),J=1,NS)                  ,65   ,66   ,67   ,68   ,69   ,70
                                                                       ,71   ,72   ,73   ,74
8100 FORMAT(12H0BODCR ERROR I3,5X
    1 57HFAILURE IN FINDING CENTROID AND RADIUS OF A BODY SECTION
    2 /30H BODY SECTION POINTS (Y,Z) ARE/(1H I3,2F20.6))
9000 CONTINUE                                                          ,75
     IF(KEE .EQ. 0  .OR.  LTAPE .LE. 0) GO TO 9900                     ,76   ,77   ,78
     WRITE (LTAPE,9100)                                                ,79   ,80
9100 FORMAT(50H0THE FOUR PTS WRITTEN BY ENRYCH ARE (Y,Z) OR (Z,Y) /
    163H COORD. OF BODY MERIDIAN LINES, BEFORE OR AFTER TRANSFORMATION,
    2/49H AT A GIVEN STATION.  IF CODE = 3, NO POINTS ARE
    3 27HINTERPOLATED IN AN INTERVAL /
    4 26H BUT THE PROGRAM CONTINUES )
9900 CONTINUE                                                          ,81
     IF(ABS(Z)-EP(5)) 9910,9910,9920                                   ,82
9910 Z=0.                                                              ,83
9920 IF(RAD-EP(5)) 9930,9930,9940                                      ,84
9930 RAD=0.                                                            ,85
9940 CONTINUE                                                          ,86
     RETURN                                                            ,87
     END                                                               ,88
```

```
      SUBROUTINE BODY
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/ NPLNB,NPLNW,JLEAD,JTRAIL,IMID,NPTS(16),X(16,90),Y(16
     1,90),Z(16,90),XCEPT(21),XCEPTB(21),XCEPTW(16),YCEPTW(16),ZCEPTW(16
     2),CODEBW(16),KPANEL(15,20),XCOR(16,21),YCOR(16,21),ZCOR(16,21),XIN
     3T(15,20,2),YINT(15,20,2),ZINT(15,20,2),XCEN(15,20),YCEN(15,20),ZCE
     4N(15,20),XCON(15,20),YCON(15,20),ZCON(15,20),AREA(15,20),ARAT(15,2
     50),THETA(15,20),ALPHA(15,20),CHORD(15,20)
      DIMENSION ND3(10855)
      EQUIVALENCE (ND3(1),NPLNB)

C     CONTROL PROGRAM FOR BODY PANELING SUBROUTINES
      DATA NAME/6HDEFINE/

C     DEFINE CONSTANTS
C     NBODY IS NUMBER OF BODY PANELS
C     KODEB,KODEI ARE SCRATCH TAPE ERROR CODES
C     IF KODEB OR KODEI ARE SET = 1 IN INPUTB
C     THEN TAPE ERROR ENCOUNTERED
 10   NBODY=0                                                          ,1
      KODEB=0                                                          ,2
      KODEI=0

C     INITIALIZE COMMON ARRAYS                                         ,3
      DO 15 I=1,10855                                                  ,4
 15   ND3(I)=0.

C     CALL SUBROUTINES                                                 ,5    ,6
 20   CALL INPUTB                                                      ,7
      IF (KODEB) 150,25,150                                            ,8
 25   IF (KODEI) 150,30,150                                            ,9
 30   CALL CRNRB                                                       ,10
      IF (KODEB) 150,40,80                                             ,11
 40   CALL AREAP(XCOR,YCOR,ZCOR,AREA,ARAT,NPER1,NPLN1)                 ,12
      CALL CENTRD (XCOR,YCOR,ZCOR,ARAT,XCEN,YCEN,ZCEN,
     1NPER1,NPLN1)                                                     ,13
      CALL THETAB                                                      ,14
      IF (KODEB) 150,50,80                                             ,15
 50   CALL CNTRLB                                                      ,16
      IF (KODEB) 150,60,80                                             ,17
 60   CALL ALPHAB                                                      ,18
      IF (KODEB) 150,90,80                                             ,19
 80   WRITE (NWRITE,900)                                               ,20   ,21
 90   CALL OUTPTB                                                      ,22
      IF (KODEB) 150,100,150                                           ,23
 100  RETURN                                                           ,24
 150  READ (NREAD,890) KARD                                            25    ,26   ,27
      IF (KARD-NAME) 150,160,150                                       ,28
 160  BACKSPACE NREAD                                                  ,29
```

```
      KODEB=-1                                                          ,30
      GO TO 100                                                         ,31
  890 FORMAT (A6)
  900 FORMAT (1H1,9X,13HERROR MESSAGE/10X,
     149HPANELLING DATA NOT COMPLETED DUE TO PROGRAM ERROR/10X,
     238HINCOMPLETE PANELLING DATA TO BE OUTPUT)
       END                                                              ,32
```

```
C     CALLED BY GEOMD
C     READS A BODY DEFINITION IN THE FORM OF CROSS-SECTIONS AT VARIOUS
C       STATION PLANES, AND THEN COMPUTES BODY MERIDIAN LINES (ALSO CALLED
C       PERCENT LINES) EACH OF WHICH CONSISTS OF 3-DIMENSIONAL POINTS ON THE
C       BODY SURFACE.
C
      SUBROUTINE BODY1(DAT,LI,LO,B,LSTA,NSTA,LBPL,MBPL,NBPL
     1,LTHETS,AXIS,TITLE,LGDEF,NU)
C
      DIMENSION DAT(2),B(1),TITLE(12),NU(3),AXIS(2)
      DATA   MAX,SR/150,5HBODY1/
      LOGICAL LGDEF(3,6)
C
C     INPUTS ARE DAT,LI,LO,NU(2)
C     INPUT-OUTPUTS ARE B,NU(3)
C     OUTPUTS ARE LSTA,NSTA,LBPL,MBPL,NBPL,LTHETS,AXIS,TITLE,NU(1),LGDEF
C
C     DEFINING STATION ARRAY STARTS IN B(LSTA)
C     NSTA = NO. OF STATIONS
C
C     BODY MERIDIAN LINE ARRAY (WITH HEADER) STARTS IN B(LBPL)
C     MBPL = NO. OF BODY PERCENT LINES (MERIDIAN LINES)
C     NBPL = NO OF CELLS IN ARRAY
C     A PERCENT LINE OF N POINTS HAS 3*N ELEMENTS
C             X1,Y1,Z1, X2,Y2,Z2, ........., XN,YN,ZN
C     THE FIRST 3*J CELLS OF AN ARRAY B OF J PERCENT LINES FORMS A TABLE OF
C     CONTENTS TO THE REST OF THE ARRAY.  ELEMENTS 1,4,7,...   OF B ARE LABELS
C     FOR PERCENT LINES 1,2,3,...   ELEMENTS 2,5,8,... ARE THE NUMBER OF
C     3-DIMENSIONAL POINTS IN PERCENT LINES 1,2,3,...   ELEMENTS 3,6,9,...   ARE
C     THE STARTING LOCATIONS IN B OF PERCENT LINES 1,2,3,...
C     FOR EXAMPLE, IF B(7) = 30., B(8) = 12. AND B(9) = 49., THEN THE PERCENT
C     LINE WITH THE LABEL 30. CONTAINS 12 POINTS AND STARTS IN B(49).
C
C     THE STANDARD SET OF THETAS STARTS IN B(LTHETS).
C       THERE ARE MBPL OF THEM
C     AXIS = Y,Z COORD. OF MAIN BODY AXIS
C     TITLE = BODY TITLE
C     LGDEF(1,1) = .TRUE. TO INDICATE BODY DEF. CALLED FOR.
C     LGDEF(2,1) = .TRUE. IF BODY SUCCESSFULLY DEFINED.
C
C     NU IS AN ERROR ARRAY WHICH IS PASSED DOWN TO LOWER LEVEL SUBROUTINES.
C     NU(1), AN OUTPUT, IS ZERO FOR SUCCESS, NON-ZERO FOR FAILURE.
C     NU(2), AN INPUT, MAY BE INTERPRETED AS A  TAPE NUMBER ON WHICH TO
C       WRITE AN ERROR MESSAGE, IF GREATER THAN ZERO.
C     NU(3), BOTH AN INPUT AND AN OUTPUT, IS AN ERROR COUNTER AND ERROR MESSAGE
C       LIMITER.  SOME LOWER LEVEL SUBROUTINES MAY SUBTRACT 1 FROM NU(3) IF AN
C       ERROR OCCURS AND THEN WRITE AN ERROR MESSAGE IF NU(3) AND NU(2) ARE BOTH
C       GREATER THAN ZERO.
C
C
      LGDEF(1,3)=.TRUE.
      READ(LI,100)TITLE                                                  ,1
  100 FORMAT(12A6)                                                       ,2    ,3    ,4
```

```
      WRITE(LO,110)DAT,TITLE                                            ,5     ,6     ,7
  110 FORMAT(16H1BODY DEFINITION 38X 2A6/1H012A6)
C
      READ(LI,200)BNS,BTHETA,AXIS,CHD                                   ,8     ,9     ,10
  200 FORMAT(6F10.0)
      NSTA=BNS                                                          ,11
      NTHETA=BTHETA                                                     ,12
      WRITE(LO,300)NSTA,NTHETA,AXIS,CHD                                 ,13    ,14    ,15
  300 FORMAT(10H0THERE ARE I3,13H STATIONS AND I3,
     1 15H MERIDIAN LINES/
     2 59H0THE MAIN BODY AXIS (PARALLEL TO THE X AXIS) IS LOCATED AT
     3/8X,3HY = F9.4,4X,3HZ = F9.4/
     4 26H0CHORD-HEIGHT TOLERANCE = F8.4//)
C
C     CHECK NSTA,NTHETA
      IF(NSTA .LT. 2 .OR. NSTA .GT. MAX .OR. NTHETA .LT. 2 .OR.
     1 NTHETA .GT. MAX)GO TO 8000
C
C     RESERVE STORAGE                                                   ,16    ,17    ,18
      LSTA=IRSERV(NSTA,B,LSTA)                                          ,19
      LTHETS=IRSERV(NTHETA,B,LTHETS)                                    ,20
      LTHETA=IRSERV(NTHETA,B,LTHETA)                                    ,21
      LRHO=IRSERV(NTHETA,B,LRHO)                                        ,22
      NTHET3=3*NTHETA                                                   ,23
      LPTS=IRSERV(NTHET3,B,LPTS)
C     THESE LAST 3 ARRAYS ARE SCRATCH STORAGE FOR BODY1R.
C     LEAVE BALANCE OF STORAGE FOR MERIDIAN LINES.
C       WE WILL RELEASE EXCESS LATER.                                   ,24
      MBPL=NTHETA                                                       ,25
      NBPL=IPACK(B)                                                     ,26
      MU=2
C     SEE IF ENOUGH ROOM                                                ,27
      IF(NBPL-NTHET3*(NSTA+1))8000,600,600                              ,28
  600 LBPL=IRSERV(NBPL,B,LBPL)                                          ,29
      IF(LBPL)8000,8000,700
C
C     READ STANDARD SET OF THETAS                                       ,30
  700 LTH=LTHETS+NTHETA-1                                               ,31
      READ(LI,200)(B(I),I=LTHETS,LTH)
C
C     READ EACH DEFINING SECTION.  FIND X,Y,Z,RHO,THETA FOR EACH PT.
C     WRITE DATA FOR EACH SECTION.  STORE PTS. IN B(LBPL) IN
C       MERIDIAN LINE ORDER.                                            ,32    ,33    ,34    ,35    ,36
      MU=3                                                              ,37
      CALL BODY1R(LI,LO,NTHETA,NSTA,AXIS,TITLE,DAT,B(LTHETS),
     1 B(LTHETA),B(LRHO),B(LPTS),B(LBPL),B(LSTA),NU)                    ,38
      IF(NU)8000,900,8000
C
C     PERFORM 3D INTERPOLATION ON EACH MERIDIAN LINE.
C     THE ENRICHED MERIDIAN LINES (PRECEDED BY HEADER) WILL BE
C       STORED STARTING IN B(LBPL).                                     ,39
  900 MAXP=NBPL/3                                                       ,40
      MU=4                                                              ,41
      CALL RICHNA(B(LBPL),MAXP,NTHETA,NSTA,B(LTHETS),.001
```

```
     1 ,CHD,O,NA,NU)
C     NA = TOTAL NO OF PTS                                                  ,42
      IF(NU)8000,1000,8000                                                  ,43
 1000 NBPL=3*(NA+NTHETA)
C
C     RELEASE UNUSED STORAGE                                                ,44
      MU=5                                                                  ,45
      IF(IRLEAS(B(LBPL),NBPL))8000,2000,8000
C
C     DELETE SCRATCH ARRAYS                                                 ,46
 2000 IF(IDLETE(B(LTHETA)))8000,2100,8000                                   ,47
 2100 IF(IDLETE(B(LRHO)))8000,2200,8000                                     ,48
 2200 IF(IDLETE(B(LPTS)))8000,3000,8000
C
C     WRITE BODY MERIDIAN LINES                                             ,49
 3000 CALL BODY1M(B(LBPL),NTHETA,AXIS,TITLE,DAT,LO,B(LSTA))
C
C     SUCCESS                                                               ,50
 4000 LGDEF(2,3)=.TRUE.                                                     ,51
      GO TO 9000
C
C     ERROR.  WRITE MESSAGE.                                                ,52
 8000 I=MERR(NU,NU+1,ML,SR)                                                 ,53
      NU=MU
C     IF MU = 3 OR 4, RELEASE BULK OF LBPL STORAGE SO WING CAN BE PROCESSED ,54
      IF(MU .LT. 3 .OR. MU .GT. 4)GO TO 9000                                ,55   ,56   ,57
      I=IRLEAS(B(LBPL),1000)
C                                                                           ,58
 9000 RETURN                                                                ,59
      END                                                                   ,60
```

```
C     USED BY BODY1 TO PRINT BODY MERIDIAN LINES.
C     ALSO COMPUTES RHO, THETA FOR EACH POINT.
C     THE RHO, THETA ORIGIN IS ON BODY AXIS.
C
      SUBROUTINE BODY1M(B,NB,AXIS,TITLE,DAYT,LO,STA)
C
      DIMENSION B(1),AXIS(2),TITLE(12),DAYT(2),STA(1),XYZ(3)
      LOGICAL NEWPCT,WTITLE
      DATA MAXLIN,DEGRAD,BLANK,STAR/45,0206712273406,1H ,3H  */
C
C     B IS ARRAY OF NB MERIDIAN LINES, HEADED BY INDEX.
C     STA IS ARRAY OF BODY STATIONS.  POINTS OF MERIDIAN LINES
C       WHICH HAVE X = A STATION ARE MARKED BY AN *
C     MAXLIN = MAX NO. OF LINES PER PAGE
C     MAXEL IS MAX NO. OF XYZ ELEMENTS THAT WILL GO ON A PAGE,
C       ASSUMING NO HEADINGS.  WE ALLOW FOR HEADINGS LATER.
C
      MAXEL=3*(MAXLIN-2)                                                ,1
      NELREM=0
C                                                                       ,2
      DO 4000 K=1,NB
C     WORKING ON KTH MERIDIAN LINE                                      ,3
      JK=3*K                                                            ,4
      NPTS=B(JK-1)                                                      ,5
      LOCST=B(JK)                                                       ,6
      LOCEND=LOCST+3*NPTS-1
C     THE X OF THE FIRST POINT IN THE MERIDIAN LINE IS
C       IN B(LOCST).
C     THE Z OF THE LAST POINT IS IN B(LOCEND).
C     FOR PAGE CONTROL WE WRITE A BLOCK OF POINTS FROM
C       B(LA) TO B(LB).                                                 ,7
      LA=LOCST                                                          ,8
      NEWPCT=.TRUE.                                                     ,9
      ISTA=1
C
C     WE KEEP COMING BACK TO HERE UNTIL ALL POINTS ARE WRITTEN          ,10
  200 NELREM=NELREM-9
C     THIS ALLOWS FOR HEADING                                           ,11
      IF(NELREM-6)300,400,400
C     NOT ENOUGH ROOM FOR 2 POINTS, SO WILL START NEW PAGE              ,12
  300 WTITLE=.TRUE.                                                     ,13
      NELREM=MAXEL-6                                                    ,14
      GO TO 500                                                         ,15
  400 WTITLE=.FALSE.
C                                                                       ,16
  500 LB=MINO (LOCEND,LA+NELREM-1)                                      ,17
      NL=(LB-LA+1)/3                                                    ,18
      IF(.NOT. WTITLE) GO TO 600                                        ,19  ,20  ,21
      WRITE(LO,550)TITLE,DAYT                                           ,22  ,23  ,24
  550 FORMAT(1H1 12A6/54X 2A6)
  600 IF(NEWPCT) GO TO 700
C     WRITE HEADING                                                     ,25  ,26  ,27
      WRITE(LO,650)K                                                    ,28  ,29  ,30
```

```
  650 FORMAT(19H0BODY MERIDIAN LINE I4,14H   (CONTINUED))
      GO TO 800                                                  ,31
  700 WRITE(LO,750)K                                             ,32   ,33   ,34
  750 FORMAT(19H0BODY MERIDIAN LINE I4)
      NUM=0                                                      ,35
  800 WRITE(LO,850)                                              ,36   ,37
  850 FORMAT(14X1HX10X1HY10X1HZ12X3HRHO7X5HTHETA)
C
C     READY TO WRITE POINTS, COMPUTING RHO-THETA AS WE GO
      DO 2000 L=1,NL                                             ,38
      NUM=NUM+1                                                  ,39
      CALL TRAV(XYZ,B(LA),3)                                     ,40
      DY=XYZ(2)-AXIS(1)                                          ,41
      DZ=XYZ(3)-AXIS(2)                                          ,42
      RHO=SQRT(DY*DY+DZ*DZ)                                      ,43
      IF(RHO)1100,1100,1200                                      ,44
 1100 THETA=0.                                                   ,45
      GO TO 1300                                                 ,46
 1200 THETA = ATN1(ABS(DY),DZ)*DEGRAD
C                                                                ,47
 1300 PNTR=BLANK
C     DOES THIS PT. LIE IN A DEFINING STATION PLANE              ,48
      IF(ABS(XYZ-STA(ISTA))-1.E-4) 1400,1400,1500
C     YES.  PRINT *                                              ,49
 1400 PNTR=STAR                                                  ,50
      ISTA=ISTA+1
C                                                                ,51
 1500 WRITE(LO,1550)NUM,XYZ,PNTR,RHO,THETA                       ,52   ,53   ,54
 1550 FORMAT(I4,4X,3F11.4,A3,2F11.4)
      LA=LA+3                                                    ,55
 2000 CONTINUE
C                                                                ,56   ,57
      NELREM=NELREM-3*NL                                         ,58
      IF(LB-LOCEND)3000,4000,4000                                ,59
 3000 NEWPCT=.FALSE.                                             ,60
      LA=LB+1                                                    ,61
      GO TO 200
C                                                                ,62
 4000 CONTINUE
C                                                                ,63   ,64
      RETURN                                                     ,65
      END                                                        ,66
```

```
C     USED BY BODY1 TO READ DEFINING SECTION DATA, PROCESS IT,
C       WRITE SECTION DATA, AND FORM MERIDIAN LINES.
C
      SUBROUTINE BODY1R(LI,LO,NPT,NSTA,BAXIS,TITLE,DAT,THETS,
     1 THETA,RHO,PT, BPL,STA,NU)
C
      DIMENSION BAXIS(2),TITLE(12),DAT(2),THETS(NPT),THETA(NPT)
     1,RHO(NPT), PT(3,NPT),BPL(3,NSTA,NPT),NU(3),SECAX(2),STA(NSTA)
C
      LOGICAL NEWPAG,NEWSTA
C
C     INPUTS ARE
C       LI = INPUT TAPE, LO = OUTPUT TAPE
C       NPT = NO. OF POINTS PER SECTION
C       NSTA = NO. OF STATIONS (DEFINING SECTIONS)
C       BAXIS = Y,Z OF BODY AXIS (AXIS PARALLEL TO X-AXIS)
C       TITLE, DAT = DATE
C       THETS = SET OF STANDARD THETA-VALUES (DEGREES)
C
C     SCRATCH ARRAYS ARE THETA, RHO AND PT
C
C     OUTPUTS ARE
C       BPL = XYZ COORD. OF POINTS ON MERIDIAN LINES
C       STA = ARRAY OF BODY STATIONS
C       NU(1) = 0 IF SUCCESS
C
      DATA LINPAG /45/
C     CHANGE LINPAG  IF 45 LINES/PAGE NOT OK
C     ASSUME 7 LINES HAVE BEEN WRITTEN ON PAGE (TITLE, ETC.)
C
      MAXLIN=LINPAG-2                                                   ,1
      LINREM=MAXLIN-5                                                   ,2
      NEWPAG=.FALSE.
C                                                                       ,3
      DO 2000 K=1,NSTA
C                                                                       ,4
      NEWSTA=.TRUE.
C     READ BODY SECTION DATA CARDS, COMPUTE PTS. ON SECTION             ,5
      CALL BODY1S(LI,BAXIS,NPT,THETS,SECAX,SCODE,THETA,RHO,PT,NU)
C     SECAX = Y,Z COORD. OF SECTION AXIS
C     NU=1 IF SCODE NOT BETWEEN 0. AND 6.
C     NU=2 IF SCODE = 3. (ELLIPSE) AND MAJOR OR MINOR AXIS N.G.         ,6
      IF(NU)8000,100,8000                                               ,7
  100 STA(K)=PT                                                         ,8
      LINES=NPT                                                         ,9
      NUM=0
C                                                                       ,10
  200 LINREM=LINREM-5                                                   ,11
      IF(LINREM-6)300,400,400                                           ,12
  300 NEWPAG=.TRUE.                                                     ,13
      LINREM=MAXLIN                                                     ,14
      GO TO 500                                                         ,15
  400 NEWPAG=.FALSE.
```

```
C                                                                       ,16
  500 NL=MINO(LINES,LINREM)                                             ,17
      IF(.NOT. NEWPAG)  GO TO 600                                       ,18   ,19   ,20
      WRITE(LO,550)TITLE,DAT                                            ,21   ,22   ,23
  550 FORMAT(1H1 12A6/54X 2A6)
C
  600 IF(NEWSTA) GO TO 700                                              ,24   ,25   ,26
      WRITE(LO,610)PT(1,1),SCODE                                        ,27   ,28   ,29
  610 FORMAT(18H0BODY STATION  X = F10.4,13H  (CONTINUED)
     1 8X,8HSCODE  = F4.0)
      GO TO 800
C                                                                       ,30
  700 WRITE(LO,710)PT(1,1),SCODE                                        ,31   ,32   ,33
  710 FORMAT(18H0BODY STATION  X = F10.4,21X,8HSCODE  = F4.0)
C
  800 WRITE(LO,810)SECAX                                                ,34   ,35   ,36
  810 FORMAT(41H RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (
     1 F9.4,1H, F9.4, 1H)/13H0MERIDIAN NO. 4X,3HRHO 7X
     2 5HTHETA 18X 1HY 10X 1HZ)
C
      DO 900 L=1,NL                                                     ,37
      NUM=NUM+1                                                         ,38
  900 WRITE(LO,910)NUM,RHO(NUM),THETA(NUM),(PT(I,NUM),I=2,3)            ,39   ,40   ,41   ,42   ,43   ,44
                                                                        ,45
  910 FORMAT(I9,F14.4,F11.4,F21.4,F11.4)
C
      LINES=LINES-NL                                                    ,46
      LINREM=LINREM-NL                                                  ,47
      IF(LINES)1100,1100,1000                                           ,48
 1000 NEWSTA=.FALSE.                                                    ,49
      GO TO 200
C
C     STORE SECTION POINTS AS MERIDIAN LINES                            ,50
 1100 DO 1200 J=1,NPT                                                   ,51
      DO 1200 I=1,3                                                     ,52
 1200 BPL(I,K,J)=PT(I,J)
C                                                                       ,53   ,5    ,55
 2000 CONTINUE
C                                                                       ,56   ,5
 8000 RETURN                                                            ,58
      END                                                               ,59
```

```
C     USED BY BODY1R TO READ BODY SECTION CARDS, COMPUTE PTS. ON SECTION
C
      SUBROUTINE BODY1S(LI,BAXIS,NPT,THETS,SECAX,SCODE,THETA,RHO,PT,NU)
C
      DIMENSION BAXIS(2),THETS(NPT),SECAX(2),THETA(NPT),RHO(NPT),PT(3,NP
     1T) ,RAD(2),V(2),PTSEC(2),YZ(2)
C
C    INPUTS ARE
C     LI = INPUT TAPE NO.
C     BAXIS = Y,Z OF MAIN BODY AXIS
C     NPT = NO. OF PTS IN SECTION
C     THETS = STANDARD SET OF THETA-VALUES TO USE IF SCODE = 1,2 OR 3
C    OUTPUTS ARE
C     SECAX = Y,Z OF SECTION AXIS
C     SCODE = CODE FOR TYPE OF SECTION
C     THETA, RHO = POLAR COORD. OF PTS ON SECTION (ORIGIN AT SECAX)
C     PT = XYZ COORD. OF PTS ON SECTION
C     NU = 0 IF SUCCESS
C          1 IF SCODE .LT. 0 OR .GT. 6
C          2 IF SCODE = 3 (ELLIPSE) AND A SEMI-AXIS IS = 0.
C
      DATA DEGRAD/0206712273406/                                    180/PI
C
      READ(LI,100)STA,YZ,SCODE,RAD                                  ,1    ,2    ,3
 100  FORMAT(6F10.0)
      MU=1
C     CHECK FOR VALID SCODE                                         ,4
      IF(SCODE .LT. 0. .OR. SCODE .GT. 6.) GO TO 8000               ,5    ,6    ,7
 200  ICODE=SCODE                                                   ,8
      DO 210 I=1,2                                                  ,9
 210  SECAX(I)=BAXIS(I)+YZ(I)
C
C     IF ICODE .LE. 0, SECTION SAME AS PREVIOUS EX. FOR STATION     ,10   ,11
      IF(ICODE)5000,5000,300                                        ,12
 300  IF(.NOT. (ICODE .EQ. 1 .OR. ICODE .EQ. 5))GO TO 400
C     ICODE = 1 OR 5.   READ RHO                                    ,13   ,14   ,15
      READ(LI,100)(RHO(I),I=1,NPT)                                  ,16   ,17   ,18   ,19   ,20
 400  IF(ICODE .LT.4 .OR. ICODE .GT. 5)GO TO 500
C     ICODE = 4 OR 5.  READ THETA                                   ,21   ,22   ,23
      READ(LI,100)(THETA(I),I=1,NPT)                                ,24   ,25   ,26   ,27   ,28
      GO TO 1000                                                    ,29
 500  IF(ICODE .GT. 3)GO TO 1000
C     ICODE = 1,2 OR 3.  STORE THETAS IN THETA                      ,30   ,31   ,32
      CALL TRAV(THETA,THETS, NPT)
C
C     THE TABLE BELOW SHOWS WHETHER RHO AND THETA HAVE BEEN FOUND
C     FOR EACH POINT ON THE SECTION
C
C     ICODE RHO THETA
C       0   YES  YES
C       1   YES  YES
C       2   NO   YES
C       3   NO   YES
```

```
C        4   NO   YES
C        5   YES  YES
C        6   NO   NO
C                                                                          ,33
 1000 GO TO (2000,1100,1200,1100,2000,4000),ICODE
C
C     CIRCLE (ICODE=2 OR 4)                                                ,34
 1100 DO 1110 I=1,NPT                                                      ,35
 1110 RHO(I)=RAD                                                           ,36   ,37
      GO TO 2000
C
C     ELLIPSE (ICODE=3)                                                    ,38
 1200 DO 1400 I=1,NPT                                                      ,39
      THETR=THETA(I)/DEGRAD                                                ,40
      CALL ELLIPR(RAD,RAD(2),SIN(THETR),COS(THETR),RHO(I),X,Y,NU)          ,41
      MU=2                                                                 ,42
      IF(NU)8000,1300,8000                                                 ,43
 1300 PT(2,I)=X+SECAX(1)                                                   ,44
      PT(3,I)=Y+SECAX(2)                                                   ,45
 1400 CONTINUE                                                             ,46   ,47
      GO TO 5000
C
C     ICODE = 1,2,4 OR 5.  FIND Y,Z FOR EACH PT.                           ,48
 2000 DO 2010 I=1,NPT                                                      ,49
      THETR=THETA(I)/DEGRAD                                                ,50
      PT(2,I)=RHO(I)*SIN(THETR)+SECAX(1)                                   ,51
 2010 PT(3,I)=RHO(I)*COS(THETR)+SECAX(2)                                   ,52   ,53
      GO TO 5000
C
C     Y,Z GIVEN (ICODE=6).  READ Y,Z AND FIND RHO,THETA                    ,54
 4000 READ(LI,100)((PT(I,J),I=2,3),J=1,NPT)                                ,55   ,56   ,57   ,58   ,59   ,60
                                                                           ,61   ,62
      DO 4400 I=1,NPT                                                      ,63
      PTSEC(1)=PT(2,I)                                                     ,64
      PTSEC(2)=PT(3,I)                                                     ,65
      RHO(I)=UVECN(SECAX,PTSEC,V,0,2)                                      ,66
      IF(RHO(I))4200,4200,4300                                             ,67
 4200 THETA(I)=THETS(I)                                                    ,68
      GO TO 4400                                                           ,69
 4300 THETA(I)=ATN1(ABS(V),V(2))*DEGRAD                                    ,70
 4400 CONTINUE
C
C     WE NOW HAVE RHO,THETA,Y,Z FOR EACH PT.  STORE STA AS X.              ,71   ,72
 5000 DO 5010 I=1,NPT                                                      ,73
 5010 PT(1,I)=STA                                                          ,74   ,75
      MU=0
C                                                                          ,76
 8000 NU=MU                                                                ,77
      RETURN                                                               ,78
      END                                                                  ,79
```

```
      SUBROUTINE BSCALE(B,NB,XEND,NU)
      DIMENSION B(1)
C
C     SCALE X-VALUES OF BODY MERIDIAN LINES SUCH THAT
C     EACH LINE STARTS AT X=0, AND ENDS WITH X=XEND.
C
C     X = (X-XF)*XEND/(XL-XF)
C     WHERE XF = X OF 1ST POINT OF A LINE, XL = X OF LAST POINT
C
      J=B(3)
C     1ST PT OF 1ST LINE STARTS IN B(J)
C                                                                    ,1
      DO 1000 K=1,NB
C                                                                    ,2
      XF=B(J)                                                        ,3
      NP=B(3*K-1)                                                    ,4
      L=J+3*(NP-1)                                                   ,5
      XL=B(L)                                                        ,6
      DEL=XL-XF                                                      ,7
      IF(DEL)200,100,200                                             ,8
  100 NU=K                                                           ,9
      GO TO 2000
C                                                                    ,10
  200 DO 300 I=1,NP                                                  ,11
      B(J)=((B(J)-XF)/DEL)*XEND                                      ,12
  300 J=J+3
C                                                                    ,13   ,14
 1000 CONTINUE                                                       ,15   ,16
      NU=0                                                           ,17
 2000 RETURN                                                         ,18
      END                                                            ,19
```

```
      SUBROUTINE CAMBW (NW,NTAPEX,A,CLW,ALPHAW)
      DIMENSION A(1),ALPHAW(1),CLW(1)

C     FOR WING ONLY CASE, COMPUTES NORMAL VELOCITY COMPONENTS ON WING

      DO 100 J=1,NW                                                    ,1
      ALPHAW(J)=0.0                                                    ,2
  100 CONTINUE                                                         ,3    ,4
      DO 200 J=1,NW                                                    ,5
      READ (NTAPEX) (A(I),I=1,NW)                                      ,6    ,7    ,8    ,9    ,10
      DO 200 I=1,NW                                                    ,11
      ALPHAW(I)=ALPHAW(I)+A(I)*CLW(J)                                  ,12
  200 CONTINUE
                                                                       ,13   ,14   ,15
      RETURN                                                           ,16
      END                                                              ,17
```

```
      SUBROUTINE CAMBWB (NB,NW,NTAPEX,NTAPEY,A,CLW,ALPHAB,ALPHAW)

C     FOR WING-BODY CASES, COMPUTES NORMAL VELOCITY COMPONENTS ON WING

      DIMENSION A(1),ALPHAW(1),ALPHAB(1),CLW(1)

      DO 100 J=1,NW                                                    ,1
      ALPHAW(J)=0.                                                     ,2
  100 CONTINUE
                                                                       ,3    ,4
      DO 200 J=1,NW                                                    ,5
      READ (NTAPEX) (A(I),I=1,NW)                                      ,6    ,7    ,8    ,9    ,10
      DO 200 I=1,NW                                                    ,11
      ALPHAW(I)=ALPHAW(I)+A(I)*CLW(J)                                  ,12
  200 CONTINUE
                                                                       ,13   ,14   ,15
      DO 300 J=1,NB                                                    ,16
      READ (NTAPEY) (A(I),I=1,NW)                                      ,17   ,18   ,19   ,20   ,21
      DO 300 I=1,NW                                                    ,22
      ALPHAW(I)=ALPHAW(I)+A(I)*ALPHAB(J)                               ,23
  300 CONTINUE
                                                                       ,24   ,25   ,26
      RETURN                                                           ,27
      END                                                              ,28
```

```
      SUBROUTINE CEGAR(N,P,AREA,CGRAV)
C
C          FINDS AREA AND CENTROID OF AREA OF A POLYGON
C
      DIMENSION P(2,1),CGRAV(2)
      DOUBLE PRECISION A,CG(2)
C
      A=0.D0                                                            ,1
      CG(1)=A                                                           ,2
      CG(2)=A                                                           ,3
      P(1,N+1)=P(1,1)                                                   ,4
      P(2,N+1)=P(2,1)
C                                                                       ,5
      DO 100J=1,2                                                       ,6
      K=(J+1)/J                                                         ,7
      DO 80I=1,N                                                        ,8
   80 CG(J)=CG(J)+(P(J,I+1)**2+P(J,I)**2+P(J,I+1)*P(J,I))*(P(K,I+1)-P(K,
     1I))                                                               ,9     ,10
  100 CONTINUE
C                                                                       ,11    ,12
      DO 120L=1,N                                                       ,13
  120 A=A+(P(2,L+1)+P(2,L))*(P(1,L+1)-P(1,L))
C                                                                       ,14    ,15
      IF (A) 160,140,160                                                ,16
  140 CG(1)=0.                                                          ,17
      CG(2)=0.                                                          ,18
      GO TO 180                                                         ,19
  160 CG(1)=CG(1)/(-A*3.)                                               ,20
      CG(2)=CG(2)/(A*3.)                                                ,21
      A=ABS (A)/2.                                                      ,22
  180 CONTINUE                                                          ,23
      AREA=A                                                            ,24
      CGRAV(1)=CG(1)                                                    ,25
      CGRAV(2)=CG(2)                                                    ,26
      RETURN                                                            ,27
       END                                                              ,28
```

```
      SUBROUTINE CENTRC (X,Y,Z,R,XC,YC,ZC,NC,NR)
      DIMENSION X(16,16),Y(16,16),Z(16,16),R(15,15),XC(15,15),
     1YC(15,15),ZC(15,15)

C     SUBROUTINE TO CALCULATE BODY AND WING PANEL
C     CENTROID COORDINATES

C     DEFINE FUNCTION TO BE USED TO CALCULATE
C     PANEL CENTROID COORDINATE
      CENT(A1,A2,A3,P1,P2)=A1+P1*(A2+P2*(A3-A2)-A1)

C     DEFINE CONSTANTS
      P1=2./3.                                                          ,1
      P2=.5

C     CALCULATE PANEL CENTROID COORDINATES                              ,2
 50   DO 60 I=1,NC                                                      ,3
      DO 60 J=1,NR                                                      ,4
      XC1=CENT(X(I,J+1),X(I,J),X(I+1,J+1),P1,P2)                        ,5
      YC1=CENT(Y(I,J+1),Y(I,J),Y(I+1,J+1),P1,P2)                        ,6
      ZC1=CENT(Z(I,J+1),Z(I,J),Z(I+1,J+1),P1,P2)                        ,7
      XC2=CENT(X(I+1,J),X(I,J),X(I+1,J+1),P1,P2)                        ,8
      YC2=CENT(Y(I+1,J),Y(I,J),Y(I+1,J+1),P1,P2)                        ,9
      ZC2=CENT(Z(I+1,J),Z(I,J),Z(I+1,J+1),P1,P2)                        ,10
      XC(I,J)=XC2+R(I,J)*(XC1-XC2)                                      ,11
      YC(I,J)=YC2+R(I,J)*(YC1-YC2)                                      ,12
 60   ZC(I,J)=ZC2+R(I,J)*(ZC1-ZC2)                                      ,13   ,14   ,15
 100  RETURN                                                            ,16
      END                                                               ,17
```

```
      SUBROUTINE CHECK (X,N,K,L,N6)
      DIMENSION X(1)

C     SUBROUTINE TO CHECK IF SEQUENCE IS
C     MONOTONIC INCREASING OR DECREASING
   10 IF (K .LT. -2 .OR. K .EQ. 0 .OR. K .GT. 2) GO TO 80                  ,1    ,2    ,3
      K1=(K+IABS(K))/(2*K)                                                 ,4
      K2=K1-1                                                              ,5
      N1=N-1                                                               ,6
   30 DO 60 I=1,N1                                                         ,7    ,8
      L1=K1*(I+1)+K2*(N-I)                                                 ,9
      L2=L1-K1+K2                                                          ,10
      IF (IABS(K)-1) 80,40,50                                              ,11
   40 IF (X(L1)-X(L2)) 90,60,60                                            ,12
   50 IF (X(L1)-X(L2)) 90,90,60                                            ,13
   60 CONTINUE                                                             ,14   ,15
      L=0                                                                  ,16
      GO TO 100                                                            ,17
   80 WRITE (N6,900)                                                       ,18   ,19
      WRITE (N6,910) K                                                     ,20   ,21   ,22
      L=1                                                                  ,23
      GO TO 100                                                            ,24
   90 WRITE (N6,900)                                                       ,25   ,26
      WRITE (N6,920) (X(I),I=1,N)                                          ,27   ,28   ,29   ,30   ,31
      L=1                                                                  ,32
      GO TO 100                                                            ,33
  100 RETURN                                                               ,34
  900 FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CHECK)
  910 FORMAT (1H ,9X,28HINVALID VALUE FOR K      K = ,I3)
  920 FORMAT (1H ,9X,25HSEQUENCE IS NOT MONOTONIC//
     1(F20.5))
      END                                                                  ,35
```

```
      SUBROUTINE CHORDW
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/IJ,NPTS(16),X(90,16),Y(90,16),Z(90,16),KPNT,VALUE(5),
     1YCEPT(16),SLOPE,KPANEL(2,15),XCOR(16,16),YCOR(16,16),ZCOR(16,16),X
     2INT(2,15),YINT(2,15),ZINT(2,15),XCEN(15,15),YCEN(15,15),ZCEN(15,15
     3),XCON(15,15),YCON(15,15),ZCON(15,15),AREA(15,15),ARAT(15,15),THET
     4A(15,15),ALPHA(15,15),CHORD(15,15),XFOIL(16,25,2),ZFOIL(16,25,2),X
     5NUM(16,2),XTAB1(25),ZTAB1(25),XTAB2(25),ZTAB2(25)
      DIMENSION C(4),P(3,50),Q(3,5),LOC(5)

C     SUBROUTINE TO CALCULATE WING PANEL STREAMWISE
C     CHORD LENGTHS

C     DEFINE CONSTANTS
      E=.000001                                                         ,1
      F=.001                                                            ,2
      M=3                                                               ,3
      MAX=2                                                             ,4
      N=4                                                               ,5
      C(1)=0.                                                           ,6
      C(2)=-1.                                                          ,7
      C(3)=0.

C     CALCULATE PANEL CHORD LENGTHS                                     ,8
 50   DO 70 I=1,NPLN1                                                   ,9
      DO 60 J=1,NPER1

C     SETUP ARRAY                                                       ,10
      P(1,1)=XCOR(I,J)                                                  ,11
      P(2,1)=YCOR(I,J)                                                  ,12
      P(3,1)=ZCOR(I,J)                                                  ,13
      P(1,2)=XCOR(I+1,J)                                                ,14
      P(2,2)=YCOR(I+1,J)                                                ,15
      P(3,2)=ZCOR(I+1,J)                                                ,16
      P(1,3)=XCOR(I+1,J+1)                                              ,17
      P(2,3)=YCOR(I+1,J+1)                                              ,18
      P(3,3)=ZCOR(I+1,J+1)                                              ,19
      P(1,4)=XCOR(I,J+1)                                                ,20
      P(2,4)=YCOR(I,J+1)                                                ,21
      P(3,4)=ZCOR(I,J+1)                                                ,22
      C(4)=YCON(I,J)

C     ATTEMPT INTERSECTION                                              ,23
      CALL POLXN (C,P,M,N,F,MAX,Q,LOC,NINT)                             ,24
      IF (NINT-2) 80,60,90                                              ,25
 60   CHORD(I,J)=ABS(Q(1,1)-Q(1,2))                                     ,26   ,27
 70   CONTINUE                                                          ,28   ,29
      GO TO 100
```

```
C     ERROR MESSAGES                                                    ,30
 80   WRITE (NWRITE,90C)                                                ,31   ,32
      GO TO 95                                                          ,33
 90   WRITE (NWRITE,91C)                                                ,34   ,35
      GO TO 95                                                          ,36
 95   KODEW=1                                                           ,37
      GO TO 100                                                         ,38
 100  RETURN                                                            ,39
 900  FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CHORD/10X,
     124HSUBROUTINE POLXN FAILURE/10X,
     133HPANEL CHORD LENGTH NOT CALCULATED/10X,
     237HFEWER THAN TWC INTERSECTIONS COMPUTED)
 910  FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CHORD/10X,
     124HSUBROUTINE POLXN FAILURE/10X,
     133HPANEL CHORD LENGTH NOT CALCULATED/10X,
     236HMORE THAN TWO INTERSECTIONS COMPUTED)
      END                                                               ,40
```

```
CLEAR                                                                    10/04/65
7094 RELMOD ASSEMBLY.

        $IBLDR CLEAR                                                                    CLEA0000

ASSEMBLED TEXT.

        $TEXT  CLEAR                                                                    CLEA0001

                                        *CLEAR      MICHAEL SYNGE          6-9247      ASSEMBLED  21 SEP 64
                                        *  TO CLEAR ANY NUMBER OF STORAGE BUFFERS
                                        *
                                        *     CALLING SEQUENCE -
                                        *
                                        *              CALL CLEAR (BUFA,BUFB,.......)
                                        *
                                              ENTRY   CLEAR
                                        *

BINARY CARD ID. CLEA0002
        00000  0634 00 4 00004  10001   CLEAR SXA     EXIT,4
        00001  0500 00 4 00003  10000         CLA     3,4
        00002  4320 00 0 00021  10001         ANA     =D777777700000
        00003  0100 00 0 00006  10001         TZE     NEXT
        00004  0774 00 4 00000  10000   EXIT  AXT     0,4
        00005  0020 00 4 00001  10000         TRA     1,4
                                        *
        00006  0500 60 4 00003  10000   NEXT  CLA*    3,4
        00007  0621 00 0 00020  10001         STA     LENG
        00010  0500 00 4 00003  10000         CLA     3,4
        00011  0621 00 0 00016  10001         STA     BUF
        00012  0634 00 4 00014  10001         SXA     XR,4
        00013  0C74 00 4 03000  10011         TSX     $INIBFR,4
        00014  0774 00 4 00000  10000   XR    AXT     **,4
        00015  1 77777 4 02001  10011         TXI     CLEAR+1,4,-1
        00016  0000 00 0 00000  10000   BUF   HTR     **
        00017  0000 00 0 00020  10001         HTR     LENG
                                        *
        00020  0 00000 0 00000  10000   LENG
        00021  777777700000     10000         *LORG
                         00000  01111         END

        CONTROL DICTIONARY

        $CDICT CLEAR                                                                    CLEA0003

BINARY CARD ID. CLEA0004
            000022000000                PREFACE     START=0,LENGTH=18,TYPE=7094,CMPLX=5
            000004000005
            234325215160        CLEAR   DECK        LOC=0,LENGTH=18
            000022000000
            234325215160        CLEAR   REAL        LOC=0,LENGTH=0
            000000000000
            314531222651        INIBFR  VIRTUAL     SECT. 3
            200000000000
```

NO MESSAGES FOR THIS ASSEMBLY

SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

CLASS	SYMBOL	VALUE	REFERENCES
	BUF	00016	11
	CLEAR	00000	15
	EXIT	00004	0
	LENG	00020	7,17
	NEXT	00006	3
LCTR	BLCTR		
QUAL	UNQS		13
LCTR	//		
	XR	00014	12

REFERENCES TO VIRTUAL SYMBOLS.

	INIBFR	3	13

```
      SUBROUTINE CLOK(TIME,DATE)
C     DUMMY ROUTINE TO RETURN TIME, DATE AS BLANKS.
C     IT MAY BE REPLACED BY A ROUTINE WHICH RETURNS TWO
C     HOLLERITH WORDS CONTAINING THE TIME AND DATE.
      DATA T/1H /
      TIME = T                                                          ,1
      DATE = T                                                          ,2
      RETURN                                                            ,3
      END                                                               ,4
```

```
      SUBROUTINE CNTRLB
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/ NPLNB,NPLNW,JLEAD,JTRAIL,IMID,NPTS(16),X(16,90),Y(16
     1,90),Z(16,90),XCEPT(21),XCEPTB(21),XCEPTW(16),YCEPTW(16),ZCEPTW(16
     2),CODEBW(16),KPANEL(15,20),XCOR(16,21),YCOR(16,21),ZCOR(16,21),XIN
     3T(15,20,2),YINT(15,20,2),ZINT(15,20,2),XCEN(15,20),YCEN(15,20),ZCE
     4N(15,20),XCON(15,20),YCON(15,20),ZCON(15,20),AREA(15,20),ARAT(15,2
     50),THETA(15,20),ALPHA(15,20),CHORD(15,20)
      DIMENSION C(4),P(3,10),Q(3,5),LOC(5)

C     SUBROUTINE TO CALCULATE BODY PANEL
C     CONTROL POINT COORDINATES AND PANEL CHORD LENGTHS

C     DEFINE FUNCTIONS TO BE USED IN CALCULATIONS
C     XLENG CALCULATE LENGTH OF STREAMWISE CHORD
C     THROUGH PANEL CENTROID
      XLENG(X1,X2,Y1,Y2,Z1,Z2)=SQRT((X2-X1)**2+(Y2-Y1)**2+
     1(Z2-Z1)**2)

C     DEFINE CONSTANTS
 10   M=3                                                                  ,1
      MAX=2                                                                ,2
      N4=4                                                                 ,3
      N5=5                                                                 ,4
      E=0.001                                                               5
      F=0.000001

C     CALCULATE PANEL CONTROL POINT COORDINATES
C     AND PANEL CHORD LENGTHS                                              ,6
 50   DO 90 I=1,NPER1                                                      ,7
      DO 80 J=1,NPLN1                                                      ,8
      IF (KODEC) 150,53,57                                                  ,9
 53   YTEMP=YCEN(I,J)                                                      ,10
      ZTEMP=ZCEN(I,J)                                                      ,11
      GO TO 60                                                             ,12
 57   YTEMP=YCOR(I,J)+YPER*(YCOR(I+1,J+1)-YCOR(I,J))                       13
      ZTEMP=ZCOR(I,J)+YPER*(ZCOR(I+1,J+1)-ZCOR(I,J))

C     SETUP ARRAY
 60   P(1,1)=XCOR(I,J+1)                                                   ,14
      P(2,1)=YCOR(I,J+1)                                                   ,15
      P(3,1)=ZCOR(I,J+1)                                                   ,16
      P(1,2)=XCOR(I,J)                                                     ,17
      P(2,2)=YCOR(I,J)                                                     ,18
      P(3,2)=ZCOR(I,J)                                                     ,19
      P(1,3)=XCOR(I+1,J)                                                   ,20
      P(2,3)=YCOR(I+1,J)                                                   ,21
      P(3,3)=ZCOR(I+1,J)                                                   ,22
      P(1,4)=XCOR(I+1,J+1)                                                  23
                                                                           ,24
```

```
      P(2,4)=YCOR(I+1,J+1)                                              ,25
      P(3,4)=ZCOR(I+1,J+1)                                              ,26
      P(1,5)=XCOR(I,J+1)                                                 27
      P(2,5)=YCOR(I,J+1)                                                ,28
      P(3,5)=ZCOR(I,J+1)                                                ,29
      IF (ABS(ZCOR(I+1,J+1)-ZCOR(I,J+1))-F) 63,63,67                    ,30
 63   C(1)=0.                                                           ,31
      C(2)=-1.                                                          ,32
      C(3)=0.                                                           ,33
      C(4)=YTEMP                                                        ,34
      GO TO 70                                                          ,35
 67   C(1)=0.                                                           ,36
      C(2)=-(YCOR(I+1,J+1)-YCOR(I,J+1))/(ZCOR(I+1,J+1)-ZCOR(I,J+1))     ,37
      C(3)=-1.                                                          ,38
      C(4)=ZTEMP-C(2)*YTEMP

C     ATTEMPT INTERSECTION                                              ,39
 70   CALL POLXN (C,P,M,N5,F,MAX,Q,LOC,NINT)                            ,40
      IF (NINT-2) 143,73,145                                            ,41
 73   CHORD(I,J)=XLENG(Q(1,1),Q(1,2),Q(2,1),Q(2,2),Q(3,1),Q(3,2))       ,42
      IF (Q(1,1)-Q(1,2)) 75,150,77                                      ,43
 75   XCON(I,J)=Q(1,1)+XPER*(Q(1,2)-Q(1,1))                             ,44
      YCON(I,J)=Q(2,1)+XPER*(Q(2,2)-Q(2,1))                             ,45
      ZCON(I,J)=Q(3,1)+XPER*(Q(3,2)-Q(3,1))                             ,46
      GO TO 80                                                          ,47
   77 XCON(I,J)=Q(1,2)+XPER*(Q(1,1)-Q(1,2))                             ,48
      YCON(I,J)=Q(2,2)+XPER*(Q(2,1)-Q(2,2))                             ,49
      ZCON(I,J)=Q(3,2)+XPER*(Q(3,1)-Q(3,2))                             ,50
 80   CONTINUE                                                          ,51   ,52
 90   CONTINUE                                                          ,53   ,54
      GO TO 150

C     ERROR MESSAGES                                                    ,55
 143  WRITE (NWRITE,900)                                                ,56   ,57
      GO TO 147                                                          58
 145  WRITE (NWRITE,910)                                                ,59   ,60
      GO TO 147                                                         ,61
 147  KODEB=1                                                           ,62
      GO TO 150                                                          63
 150  RETURN                                                            ,64
 900  FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE CNTRLB/10X,
     134HPANEL CONTROL POINT NOT CALCULATED/10X,
     124HSUBROUTINE POLXN FAILURE/10X,
     237HFEWER THAN TWO INTERSECTIONS COMPUTED)
 910  FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE CNTRLB/10X,
     134HPANEL CONTROL POINT NOT CALCULATED/10X,
     124HSUBROUTINE POLXN FAILURE/10X,
     236HMORE THAN TWO INTERSECTIONS COMPUTED)
      END                                                               ,65
```

```
      SUBROUTINE CNTRLW
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/IJ,NPTS(16),X(90,16),Y(90,16),Z(90,16),KPNT,VALUE(5),
     1YCEPT(16),SLOPE,KPANEL(2,15),XCOR(16,16),YCOR(16,16),ZCOR(16,16),X
     2INT(2,15),YINT(2,15),ZINT(2,15),XCEN(15,15),YCEN(15,15),ZCEN(15,15
     3),XCON(15,15),YCON(15,15),ZCON(15,15),AREA(15,15),ARAT(15,15),THET
     4A(15,15),ALPHA(15,15),CHORD(15,15),XFOIL(16,25,2),ZFOIL(16,25,2),X
     5NUM(16,2),XTAB1(25),ZTAB1(25),XTAB2(25),ZTAB2(25)

C     SUBROUTINE TO CALCULATE WING PANEL CONTROL POINT
C     COORDINATES
 10   IF (KSTART-1) 150,60,20

C     CALCULATE PANEL CONTROL POINT COORDINATES ON
C     INBOARD COLUMN IF BODY AND WING INTERSECT IN
C     NON-STREAMWISE LINE                                                ,1
 20   DO 55 J=1,NPER1                                                    ,2
      I=1                                                                ,3
      IF (KODEC) 150,23,27                                               ,4
 23   YCON(I,J)=YCEN(I,J)                                                ,5
      GO TO 30                                                           ,6
 27   YCON(I,J)=YCOR(I+1,J)+YPER*(AMAX1(YCOR(I,J),YCOR(I,J+1))-
     1YCOR(I+1,J))                                                       ,7
 30   IF (YCOR(I,J)-YCON(I,J)) 33,33,35                                  ,8
 33   YPERL=(YCON(I,J)-YCOR(I+1,J))/(YCOR(I,J)-YCOR(I+1,J))              ,9
      XLEAD=XCOR(I+1,J)+YPERL*(XCOR(I,J)-XCOR(I+1,J))                    ,10
      ZLEAD=ZCOR(I+1,J)+YPERL*(ZCOR(I,J)-ZCOR(I+1,J))                    ,11
      GO TO 40                                                           ,12
 35   IF (YCOR(I,J+1)-YCON(I,J)) 37,37,140                               ,13
 37   YPERL=(YCON(I,J)-YCOR(I,J))/(YCOR(I,J+1)-YCOR(I,J))                ,14
      XLEAD=XCOR(I,J)+YPERL*(XCOR(I,J+1)-XCOR(I,J))                      ,15
      ZLEAD=ZCOR(I,J)+YPERL*(ZCOR(I,J+1)-ZCOR(I,J))                      ,16
 40   IF (YCOR(I,J+1)-YCON(I,J)) 43,43,45                                ,17
 43   YPERT=(YCON(I,J)-YCOR(I+1,J+1))/(YCOR(I,J+1)-YCOR(I+1,J+1))        ,18
      XTRAIL=XCOR(I+1,J+1)+YPERT*(XCOR(I,J+1)-XCOR(I+1,J+1))             ,19
      ZTRAIL=ZCOR(I+1,J+1)+YPERT*(ZCOR(I,J+1)-ZCOR(I+1,J+1))             ,20
      GO TO 50                                                           ,21
 45   IF (YCOR(I,J)-YCON(I,J)) 47,47,140                                 ,22
 47   YPERT=(YCON(I,J)-YCOR(I,J+1))/(YCOR(I,J)-YCOR(I,J+1))              ,23
      XTRAIL=XCOR(I,J+1)+YPERT*(XCOR(I,J)-XCOR(I,J+1))                   ,24
      ZTRAIL=ZCOR(I,J+1)+YPERT*(ZCOR(I,J)-ZCOR(I,J+1))                   ,25
 50   XCON(I,J)=XLEAD+XPER*(XTRAIL-XLEAD)                                ,26
      ZCON(I,J)=ZLEAD+XPER*(ZTRAIL-ZLEAD)                                ,27
   55 CONTINUE                                                           ,28   ,29
 60   IF (KEND-1) 150,70,70

C     CALCULATE PANEL CONTROL POINT COORDINATES FOR
C     MAIN WING REGION AND WING-TIP REGION                               ,30
 70   DO 90 I=KSTART,KEND                                                ,31
```

```
      DO 90 J=1,NPER1                                                   ,32
      IF (KODEC) 150,73,77                                              ,33
73    YCON(I,J)=YCEN(I,J)                                               ,34
      YPER=(YCON(I,J)-YCOR(I,J))/(YCOR(I+1,J)-YCOR(I,J))                ,35
      GO TO 80                                                          ,36
77    YCON(I,J)=YCOR(I,J)+YPER*(YCOR(I+1,J)-YCOR(I,J))                  ,37
80    XLEAD=XCOR(I,J)+YPER*(XCOR(I+1,J)-XCOR(I,J))                      ,38
      XTRAIL=XCOR(I,J+1)+YPER*(XCOR(I+1,J+1)-XCOR(I,J+1))               ,39
      ZLEAD=ZCOR(I,J)+YPER*(ZCOR(I+1,J)-ZCOR(I,J))                      ,40
      ZTRAIL=ZCOR(I,J+1)+YPER*(ZCOR(I+1,J+1)-ZCOR(I,J+1))               ,41
      XCON(I,J)=XLEAD+XPER*(XTRAIL-XLEAD)                               ,42
90    ZCON(I,J)=ZLEAD+XPER*(ZTRAIL-ZLEAD)                               ,43   ,44   ,45
      IF (KEND-NPLN1) 100,150,150                                       ,46
100   I=NPLN1                                                           ,47
      DO 135 J=1,NPER1                                                  ,48
      IF (KODEC) 150,103,107                                            ,49
103   YCON(I,J)=YCEN(I,J)                                               ,50
      GO TO 110                                                         ,51
107   YCON(I,J)=YCOR(I,J)+YPER*(AMIN1(YCOR(I+1,J),
     1YCOR(I+1,J+1))-YCOR(I,J))                                         ,52
110   IF (YCON(I,J)-YCOR(I+1,J)) 113,113,115                            ,53
113   YPERL=(YCON(I,J)-YCOR(I,J))/(YCOR(I+1,J)-YCOR(I,J))               ,54
      XLEAD=XCOR(I,J)+YPERL*(XCOR(I+1,J)-XCOR(I,J))                     ,55
      ZLEAD=ZCOR(I,J)+YPERL*(ZCOR(I+1,J)-ZCOR(I,J))                     ,56
      GO TO 120                                                         ,57
115   IF (YCON(I,J)-YCOR(I+1,J+1)) 117,117,140                          ,58
117   YPERL=(YCON(I,J)-YCOR(I+1,J))/(YCOR(I+1,J+1)-YCOR(I+1,J))         ,59
      XLEAD=XCOR(I+1,J)+YPERL*(XCOR(I+1,J+1)-XCOR(I+1,J))               ,60
      ZLEAD=ZCOR(I+1,J)+YPERL*(ZCOR(I+1,J+1)-ZCOR(I+1,J))               ,61
120   IF (YCON(I,J)-YCOR(I+1,J+1)) 123,123,125                          ,62
123   YPERT=(YCON(I,J)-YCOR(I,J+1))/(YCOR(I+1,J+1)-YCOR(I,J+1))         ,63
      XTRAIL=XCOR(I,J+1)+YPERT*(XCOR(I+1,J+1)-XCOR(I,J+1))              ,64
      ZTRAIL=ZCOR(I,J+1)+YPERT*(ZCOR(I+1,J+1)-ZCOR(I,J+1))              ,65
      GO TO 130                                                         ,66
125   IF (YCON(I,J)-YCOR(I+1,J)) 127,127,140                            ,67
127   YPERT=(YCON(I,J)-YCOR(I+1,J+1))/(YCOR(I+1,J)-YCOR(I+1,J+1))       ,68
      XTRAIL=XCOR(I+1,J+1)+YPERT*(XCOR(I+1,J)-XCOR(I+1,J+1))            ,69
      ZTRAIL=ZCOR(I+1,J+1)+YPERT*(ZCOR(I+1,J)-ZCOR(I+1,J+1))            ,70
130   XCON(I,J)=XLEAD+XPER*(XTRAIL-XLEAD)                               ,71
135   ZCON(I,J)=ZLEAD+XPER*(ZTRAIL-ZLEAD)                               ,72   ,73
      GO TO 150                                                         ,74
140   WRITE (NWRITE,900)                                                ,75   ,76
      GO TO 150                                                         ,77
150   RETURN                                                            ,78
900   FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE CNTRLW/10X,
     157HPANEL CONTROL POINT Y-COORDINATE NOT CORRECTLY CALCULATED)
      END                                                               ,79
```

```
      SUBROUTINE CP(NP,BETA2,CPCALC,U,V,W,CPP)

C     COMPUTES NON-LINEAR OR LINEAR COEFFICIENT OF PRESSURE

      DIMENSION U(1),V(1),W(1),CPP(1)

      IF (CPCALC) 100,200,100

C     NON-LINEAR CP CALCULATION                                          ,1
  100 CONTINUE                                                           ,2
      DO 150 J=1,NP                                                      ,3
      CPP(J)=-2.*U(J)+BETA2*U(J)**2-V(J)**2-W(J)**2                      ,4
  150 CONTINUE                                                           ,5    ,6
      GO TO 999

C     LINEAR CP CALCULATIONS                                             ,7
  200 CONTINUE                                                           ,8
      DO 250 J=1,NP                                                      ,9
      CPP(J)=-2.*U(J)                                                    ,10
  250 CONTINUE
                                                                         ,11   ,12
  999 RETURN                                                             ,13
      END                                                                ,14
```

```
      SUBROUTINE CRNRB
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/ NPLNB,NPLNW,JLEAD,JTRAIL,IMID,NPTS(16),X(16,90),Y(16
     1,90),Z(16,90),XCEPT(21),XCEPTB(21),XCEPTW(16),YCEPTW(16),ZCEPTW(16
     2),CODEBW(16),KPANEL(15,20),XCOR(16,21),YCOR(16,21),ZCOR(16,21),XIN
     3T(15,20,2),YINT(15,20,2),ZINT(15,20,2),XCEN(15,20),YCEN(15,20),ZCE
     4N(15,20),XCON(15,20),YCON(15,20),ZCON(15,20),AREA(15,20),ARAT(15,2
     50),THETA(15,20),ALPHA(15,20),CHORD(15,20)
      DIMENSION C(4),D(4),P(3,50),R(3,50),Q(3,5),LOC(3)
      DIMENSION ISAVE(21)

C     SUBROUTINE TO CALCULATE BODY PANEL CORNER
C     POINT COORDINATES

C     DEFINE CONSTANTS
      EPS=0.04                                                          ,1
      E=0.001                                                           ,2
      F=0.0001                                                          ,3
      M=3                                                               ,4
      MAX=1                                                             ,5
      N=2

C     CALCULATE X-, Y- AND Z-COORDINATES OF PANEL
C     CORNER POINTS     PREVIOUSLY DETERMINED BODY
C     PERCENT LINES ARE INTERSECTED WITH SERIES
C     OF PLANES PARALLEL TO X-Z PLANE                                   ,6
      IF (KOPTB) 300,3,13                                               ,7
    3 DO 10 I=1,NPER                                                    ,8    ,9
      NPOINT=NPTS(I)                                                    ,10
      DO 5 J=1,NPOINT                                                   ,11   ,12
      P(1,J)=X(I,J)                                                     ,13
      P(2,J)=Y(I,J)                                                     ,14
    5 P(3,J)=Z(I,J)                                                     ,15   ,16
      DO 10 J=1,NPLANE                                                  ,17   ,18
      C(1)=1.                                                           ,19
      C(2)=0.                                                           ,20
      C(3)=0.                                                           ,21
      C(4)=-XCEPT(J)                                                    ,22
      CALL POLXN (C,P,M,NPOINT,E,MAX,Q,LOC,NINT)                        ,23
      IF (NINT) 250,260,7                                               ,24
    7 XCOR(I,J)=Q(1,1)                                                  ,25
      YCOR(I,J)=Q(2,1)                                                  ,26
   10 ZCOR(I,J)=Q(3,1)

C     DEFINE KSTART     KSTART=1 INDICATES BODY HAS
C     NO FORE CLOSING                                                   ,27   ,28   ,29
   13 KSTART=1

C     DEFINE KEND      KEND=NPLN1 INDICATES BODY HAS
```

```
C       NO AFT CLOSING                                                  ,30
        KEND=NPLN1

C       REDEFINE X-, Y- AND Z-COORDINATES OF BODY PANEL
C       CORNER POINTS FOR BODY PANELS IN BODY-WING
C       INTERSECTION REGION                                             ,31
   27   IF (KOPTB) 300,30,110                                           ,32
   30   IF (NPLNW) 300,110,35                                           ,33
   35   IF (KINT) 300,300,40                                            ,34
   40   NPLNW1=NPLNW-1

C       CHECK IF BODY-WING INTERSECTION POINT IS
C       COINCIDENT (WITHIN A TOLERANCE) WITH A
C       BODY MERIDIAN LINE                                              ,45
        DO 50 J=1,NPLNW                                                 ,36   ,37
        DO 47 I=1,NPER                                                  ,38   ,39
        K=JLEAD+J                                                       ,40
        IF (ZCOR(I,K)-(ZCEPTW(J)+EPS/2.)) 43,43,47                      ,41
 43     ISAVE(J)=I-1                                                    ,42
        GO TO 50                                                        ,43
 47     CONTINUE                                                        ,44   ,45
        GO TO 270                                                       ,46
 50     CONTINUE                                                        ,47   ,48
        DO 55 J=1,NPLNW1                                                ,49   ,50
        IF (ISAVE(J+1)-ISAVE(J)) 280,55,280                             ,51
 55     CONTINUE                                                        ,52   ,53
        IMID=ISAVE(1)

C       CHECK IF BODY-WING INTERSECTION LINE IS
C       COINCIDENT (WITHIN A TOLERANCE) WITH A
C       SINGLE BODY MERIDIAN LINE                                       ,54
 60     DO 65 J=1,NPLNW                                                 ,55   ,56
        K=JLEAD+J                                                       ,57
        IF (ABS(ZCOR(IMID+1,K)-ZCEPTW(J))-EPS) 65,65,70                 ,58
 65     CONTINUE                                                        ,59   ,60
        GO TO 110                                                       ,61
 70     NPER=NPER+1                                                     ,62
        NPER1=NPER-1                                                    ,63
        ISTART=IMID+2                                                   ,64
        DO 75 I=ISTART,NPER                                             ,65   ,66
        DO 75 J=1,NPLANE                                                ,67   ,68
        L=NPER-(I-ISTART)                                               ,69
        K=L-1                                                           ,70
        XCOR(L,J)=XCOR(K,J)                                             ,71
        YCOR(L,J)=YCOR(K,J)                                             ,72
 75     ZCOR(L,J)=ZCOR(K,J)                                             ,73   ,74   ,75
        I=IMID+1

C       DEFINE ADDITIONAL MERIDIAN LINE
C       AVERAGE ADJACENT POINTS FORE AND AFT OF
C       INTERSECTION REGION                                             ,76
        DO 80 J=1,NPLANE                                                ,77   ,78
        XCOR(I,J)=XCOR(I-1,J)+.5*(XCOR(I+1,J)-XCOR(I-1,J))              ,79
        YCOR(I,J)=YCOR(I-1,J)+.5*(YCOR(I+1,J)-YCOR(I-1,J))              ,80
```

```
 80   ZCOR(I,J)=ZCOR(I-1,J)+.5*(ZCOR(I+1,J)-ZCOR(I-1,J))

C     USE BODY-WING INTERSECTION POINTS IN BODY-
C     WING INTERSECTION REGION                                          ,81    ,82
      DO 90 J=1,NPLNW                                                   ,83    ,84
      K=JLEAD+J                                                         ,85
      XCOR(I,K)=XCEPTW(J)                                               ,86
      YCOR(I,K)=YCEPTW(J)                                               ,87
   90 ZCOR(I,K)=ZCEPTW(J)                                               ,88    ,89
 100  GO TO 110

C     CALCULATE ADDITIONAL PANEL POINTS     XINT,YINT,
C     ZINT ARE POINTS ON PANEL LEADING AND TRAILING
C     EDGES USED TO DEFINE (STREAMWISE) PANEL PARTS
C     KPANEL IS CODE DEFINING PANEL TYPE                                ,90
  110 DO 220 I=1,NPER1                                                  ,91    ,92
      DO 220 J=1,NPLN1                                                  ,93    ,94
      KPANEL(I,J)=1                                                     ,95
      P(1,1)=XCOR(I,J+1)                                                ,96
      P(2,1)=YCOR(I,J+1)                                                ,97
      P(3,1)=ZCOR(I,J+1)                                                ,98
      P(1,2)=XCOR(I+1,J+1)                                              ,99
      P(2,2)=YCOR(I+1,J+1)                                              ,100
      P(3,2)=ZCOR(I+1,J+1)                                              ,101
      R(1,1)=XCOR(I,J)                                                  ,102
      R(2,1)=YCOR(I,J)                                                  ,103
      R(3,1)=ZCOR(I,J)                                                  ,104
      R(1,2)=XCOR(I+1,J)                                                ,105
      R(2,2)=YCOR(I+1,J)                                                ,106
      R(3,2)=ZCOR(I+1,J)

C     LOCATE INBOARD POINT                                              ,107
      IF (ABS(ZCOR(I+1,J+1)-ZCOR(I,J+1))-F) 153,153,155                 ,108
 153  C(1)=0.                                                           ,109
      C(2)=-1.                                                          ,110
      C(3)=0.                                                           ,111
      C(4)=YCOR(I,J)                                                    ,112
      GO TO 157                                                         ,113
 155  C(1)=0.                                                           ,114
      C(2)=-(YCOR(I+1,J+1)-YCOR(I,J+1))/(ZCOR(I+1,J+1)-ZCOR(I,J+1))     ,115
      C(3)=-1.                                                          ,116
      C(4)=ZCOR(I,J)-C(2)*YCOR(I,J)

C     ATTEMPT TRAILING-EDGE INTERSECTION                                ,117
 157  CALL POLXN (C,P,M,N,F,MAX,Q,LOC,NINT)                             ,118
      IF (NINT) 250,170,160                                             ,119
 160  XINT(I,J,1)=Q(1,1)                                                ,120
      YINT(I,J,1)=Q(2,1)                                                ,121
      ZINT(I,J,1)=Q(3,1)                                                ,122
      GO TO 190                                                         ,123
 170  KPANEL(I,J)=KPANEL(I,J)+1                                         ,124
      IF (ABS(ZCOR(I+1,J)-ZCOR(I,J))-F) 173,173,175                     ,125
 173  D(1)=0.                                                           ,126
      D(2)=-1.                                                          ,127
```

```
      D(3)=0.                                                          ,128
      D(4)=YCOR(I,J+1)                                                 ,129
      GO TO 177                                                        ,130
 175  D(1)=0.                                                          ,131
      D(2)=-(YCOR(I+1,J)-YCOR(I,J))/(ZCOR(I+1,J)-ZCOR(I,J))            ,132
      D(3)=-1.                                                         ,133
      D(4)=ZCOR(I,J+1)-D(2)*YCOR(I,J+1)

C     ATTEMPT LEADING-EDGE INTERSECTION                                ,134
 177  CALL POLXN (D,R,M,N,F,MAX,Q,LOC,NINT)                            ,135
      IF (NINT) 250,260,180                                            ,136
 180  XINT(I,J,1)=Q(1,1)                                               ,137
      YINT(I,J,1)=Q(2,1)                                               ,138
      ZINT(I,J,1)=Q(3,1)

C     LOCATE OUTBOARD POINT                                            ,139
 190  IF (ABS(ZCOR(I+1,J+1)-ZCOR(I,J+1))-F) 193,193,195               ,140
 193  C(1)=0.                                                          ,141
      C(2)=-1.                                                         ,142
      C(3)=0.                                                          ,143
      C(4)=YCOR(I+1,J)                                                 ,144
      GO TO 197                                                        ,145
 195  C(1)=0.                                                          ,146
      C(2)=-(YCOR(I+1,J+1)-YCOR(I,J+1))/(ZCOR(I+1,J+1)-ZCOR(I,J+1))    ,147
      C(3)=-1.                                                         ,148
      C(4)=ZCOR(I+1,J)-C(2)*YCOR(I+1,J)

C     ATTEMPT TRAILING-EDGE INTERSECTION                               ,149
 197  CALL POLXN (C,P,M,N,F,MAX,Q,LOC,NINT)                            ,150
      IF (NINT) 250,210,200                                            ,151
 200  XINT(I,J,2)=Q(1,1)                                               ,152
      YINT(I,J,2)=Q(2,1)                                               ,153
      ZINT(I,J,2)=Q(3,1)                                               ,154
      GO TO 220                                                        ,155
 210  KPANEL(I,J)=KPANEL(I,J)+2                                        ,156
      IF (ABS(ZCOR(I+1,J)-ZCOR(I,J))-F) 213,213,215                    ,157
 213  D(1)=0.                                                          ,158
      D(2)=-1.                                                         ,159
      D(3)=0.                                                          ,160
      D(4)=YCOR(I+1,J+1)                                               ,161
      GO TO 217                                                        ,162
 215  D(1)=0.                                                          ,163
      D(2)=-(YCOR(I+1,J)-YCOR(I,J))/(ZCOR(I+1,J)-ZCOR(I,J))            ,164
      D(3)=-1.                                                         ,165
      D(4)=ZCOR(I+1,J+1)-D(2)*YCOR(I+1,J+1)

C     ATTEMPT LEADING-EDGE INTERSECTION                                ,166
 217  CALL POLXN (D,R,M,N,F,MAX,Q,LOC,NINT)                            ,167
      IF (NINT) 250,260,218                                            ,168
 218  XINT(I,J,2)=Q(1,1)                                               ,169
      YINT(I,J,2)=Q(2,1)                                               ,170
      ZINT(I,J,2)=Q(3,1)                                               ,171
 220  CONTINUE                                                         ,172  ,173  ,174
      GO TO 300
```

```
C     ERROR MESSAGES                                                    ,175
 250  WRITE (NWRITE,900)                                                ,176  ,177
      GO TO 295                                                         ,178
 260  WRITE (NWRITE,910)                                                ,179  ,180
      GO TO 295                                                         ,181
 270  WRITE (NWRITE,920)                                                ,182  ,183
      GO TO 295                                                         ,184
 280  IMAX=ISAVE(1)                                                     ,185
      DO 285 J=2,NPLNW                                                  ,186  ,187
 285  IMAX=MAX0(IMAX,ISAVE(J))                                          ,188  ,189
      WRITE (NWRITE,930) IMAX                                           ,190  ,191  ,192
      GO TO 295                                                         ,193
 295  KODEB=1                                                           ,194
      GO TO 300                                                         ,195
 300  RETURN                                                            ,196
 900  FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CRNRB/10X,
     140HPANEL (PART) CORNER POINT NOT CALCULATED/10X,
     124HSUBROUTINE POLXN FAILURE/10X,
     152HINTERSECTION NOT CALCULATED DUE TO ZERO COEFFICIENTS)
 910  FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CRNRB/10X,
     140HPANEL (PART) CORNER POINT NOT CALCULATED/10X,
     124HSUBROUTINE POLXN FAILURE/10X,
     127HINTERSECTION DOES NOT EXIST)
 920  FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CRNRB/10X,
     156HBODY-(PLANAR) WING INTERSECTION NOT CORRECTLY CALCULATED)
 930  FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CRNRB/10X,
     140HBODY-(PLANAR) WING INTERSECTION CROSSES ,I2,
     221HTH BODY MERIDIAN LINE)
       END                                                              ,197
```

```
      SUBROUTINE CRNRW
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/IJ,NPTS(16),X(90,16),Y(90,16),Z(90,16),KPNT,VALUE(5),
     1YCEPT(16),SLOPE,KPANEL(2,15),XCOR(16,16),YCOR(16,16),ZCOR(16,16),X
     2INT(2,15),YINT(2,15),ZINT(2,15),XCEN(15,15),YCEN(15,15),ZCEN(15,15
     3),XCON(15,15),YCON(15,15),ZCON(15,15),AREA(15,15),ARAT(15,15),THET
     4A(15,15),ALPHA(15,15),CHORD(15,15),XFOIL(16,25,2),ZFOIL(16,25,2),X
     5NUM(16,2),XTAB1(25),ZTAB1(25),XTAB2(25),ZTAB2(25)
      DIMENSION C(4),P(3,50),Q(3,5),LOC(3)

C     SUBROUTINE TO CALCULATE WING PANEL CORNER
C     POINT COORDINATES

C     CALCULATE Y-AXIS INTERCEPT OF OUTBOARD CUTTING
C     PLANE
      IF (KOPTW) 300,10,80                                          ,1
 10   IF (IJ) 300,13,30                                             ,2
 13   IF (SLOPE) 20,14,20                                           ,3
 14   IF (KPNT) 17,17,18                                            ,4
 17   YCEPT(NPLANE)=VALUE(2)                                        ,5
      GO TO 27                                                      ,6
 18   YCEPT(NPLANE)=VALUE(4)                                        ,7
      GO TO 27                                                      ,8
 20   IF (KPNT-1) 23,23,25                                          ,9
 23   YCEPT(NPLANE)=VALUE(2)-VALUE(1)/SLOPE                         ,10
      GO TO 27                                                      ,11
 25   YCEPT(NPLANE)=VALUE(4)-VALUE(3)/SLOPE                         ,12
 27   IF (KINT) 300,40,30

C     DEFINE INBOARD EDGE PANEL CORNER POINTS FOR
C     INBOARD COLUMN OF PANELS TO BE BODY-WING
C     INTERSECTION POINTS IF BODY AND WING INTER-
C     SECT     OTHERWISE SKIP TO STATEMENT 40                       ,13
 30   DO 35 J=1,NPER                                                ,14
      XCOR(1,J)=XI(J)                                               ,15
      YCOR(1,J)=YI(J)                                               ,16
 35   ZCOR(1,J)=ZI(J)

C     CALCULATE X-, Y- AND Z-COORDINATES OF PANEL
C     CORNER POINTS     PREVIOUSLY DETERMINED
C     WING PERCENT LINES ARE INTERSECTED WITH
C     SERIES OF CUTTING PLANES                                      ,17   ,18
 40   DO 70 J=1,NPER                                                ,19
      NPOINT=NPTS(J)                                                ,20
      DO 43 I=1,NPOINT                                              ,21
      P(1,I)=X(I,J)                                                 ,22
      P(2,I)=Y(I,J)                                                 ,23
 43   P(3,I)=Z(I,J)                                                 ,24   ,25
      DO 50 I=1,NPLN1                                               ,26
```

```
      K=I+KINT                                                          ,27
      C(1)=0.                                                           ,28
      C(2)=1.                                                           ,29
      C(3)=0.                                                           ,30
      C(4)=-YCEPT(I)                                                    ,31
      E=0.001                                                           ,32
      M=3                                                               ,33
      MAX=1                                                             ,34
      CALL POLXN (C,P,M,NPOINT,E,MAX,Q,LOC,NINT)                        ,35
      IF (NINT) 230,240,47                                              ,36
 47   XCOR(K,J)=Q(1,1)                                                  ,37
      YCOR(K,J)=Q(2,1)                                                  ,38
 50   ZCOR(K,J)=Q(3,1)                                                  ,39   ,40
      IF (SLOPE) 57,53,57                                               ,41
 53   C(4)=-YCEPT(NPLANE)                                               ,42
      GO TO 60                                                          ,43
 57   C(1)=1.                                                           ,44
      C(2)=-SLOPE                                                       ,45
      C(3)=0.                                                           ,46
      C(4)=SLOPE*YCEPT(NPLANE)                                          ,47
 60   CALL POLXN (C,P,M,NPOINT,E,MAX,Q,LOC,NINT)                        ,48
      IF (NINT) 230,240,65                                              ,49
 65   L=NPLANE+KINT                                                     ,50
      XCOR(L,J)=Q(1,1)                                                  ,51
      YCOR(L,J)=Q(2,1)                                                  ,52
      ZCOR(L,J)=Q(3,1)                                                  ,53
 70   CONTINUE                                                          ,54   ,55
 75   NPLANE=NPLANE+KINT                                                ,56
      NPLN1=NPLANE-1

C     DEFINE KSTART     IF KSTART = 1,WING-BODY
C     INTERSECTION IS STRAIGHT LINE IN STREAMWISE
C     DIRECTION    IF KSTART = 2,OTHERWISE                              ,57
 80   DO 85 J=1,NPER1                                                   ,58
      IF (ABS(YCOR(1,J+1)-YCOR(1,J))-.001) 85,85,87                     ,59
 85   CONTINUE                                                          ,60   ,61
      KSTART=1                                                          ,62
      GO TO 100                                                         ,63
 87   KSTART=2

C     DEFINE KEND     IF KEND EQUAL TO NPLN1
C     WING HAS STREAMWISE OUTBOARD EDGE     IF
C     KEND EQUAL TO NPLN1 OTHERWISE                                     ,64
 100  DO 110 J=1,NPER1                                                  ,65
      IF (ABS(YCOR(NPLANE,J+1)-YCOR(NPLANE,J))-.001) 110,110,120        ,66
 110  CONTINUE                                                          ,67   ,68
      KEND=NPLN1                                                        ,69
      GO TO 130                                                         ,70
 120  KEND=NPLN1-1

C     CALCULATE ADDITIONAL PANEL POINTS FOR
C     PANELS IN BODY-WING INTERSECTION
C     XINT,YINT,ZINT ARE POINTS ON PANEL LEADING AND
C     TRAILING EDGES USED TO DEFINE (STREAMWISE) PANEL
```

```
C       PARTS                                                           ,71
 130    IF (KSTART-1) 30C,170,140                                       ,72
 140    DO 160 J=1,NPER1                                                ,73
        P(1,1)=XCOR(2,J+1)                                              ,74
        P(2,1)=YCOR(2,J+1)                                              ,75
        P(3,1)=ZCOR(2,J+1)                                              ,76
        P(1,2)=XCOR(1,J+1)                                              ,77
        P(2,2)=YCOR(1,J+1)                                              ,78
        P(3,2)=ZCOR(1,J+1)

C       LOCATE POINT
        C(1)=0.                                                         ,79
        C(2)=-1.                                                        ,80
        C(3)=0.                                                         ,81
        C(4)=YCOR(1,J)                                                  ,82
        E=0.001                                                         ,83
        M=3                                                             ,84
        MAX=1                                                           ,85
        N=2                                                             ,86

C       ATTEMPT TRAILING-EDGE INTERSECTION
        CALL POLXN (C,P,M,N,E,MAX,Q,LOC,NINT)                           ,87
        IF (NINT) 230,15C,145                                           ,88
 145    XINT(1,J)=Q(1,1)                                                ,89
        YINT(1,J)=Q(2,1)                                                ,90
        ZINT(1,J)=Q(3,1)                                                ,91
        KPANEL(1,J)=1                                                   ,92
        GO TO 160                                                       ,93
 150    P(1,1)=XCOR(2,J)                                                ,94
        P(2,1)=YCOR(2,J)                                                ,95
        P(3,1)=ZCOR(2,J)                                                ,96
        P(1,2)=XCOR(1,J)                                                ,97
        P(2,2)=YCOR(1,J)                                                ,98
        P(3,2)=ZCOR(1,J)                                                ,99
        C(4)=YCOR(1,J+1)                                                ,100
        CALL POLXN (C,P,M,N,E,MAX,Q,LOC,NINT)                           ,101
        IF (NINT) 230,24C,155                                           ,102
 155    XINT(1,J)=Q(1,1)                                                ,103
        YINT(1,J)=Q(2,1)                                                ,104
        ZINT(1,J)=Q(3,1)                                                ,105
        KPANEL(1,J)=2                                                   ,106
 160    CONTINUE                                                        ,107
        DO 165 J=1,NPER1                                                ,108  ,109
        IF (YINT(1,J)-YCOR(2,J)) 165,165,250                            ,110
 165    CONTINUE                                                        ,111
 170    IF (KEND-NPLN) 180,210,300                                      ,112  ,113
 180    DO 200 J=1,NPER1                                                ,114
        P(1,1)=XCOR(NPLN1,J)                                            ,115
        P(2,1)=YCOR(NPLN1,J)                                            ,116
        P(3,1)=ZCOR(NPLN1,J)                                            ,117
        P(1,2)=XCOR(NPLANE,J)                                           ,118
        P(2,2)=YCOR(NPLANE,J)                                           ,119
        P(3,2)=ZCOR(NPLANE,J)                                           ,120
        C(1)=0.                                                         ,121
                                                                        ,122
```

```
      C(2)=-1.                                                          ,123
      C(3)=0.                                                           ,124
      C(4)=YCOR(NPLANE,J+1)                                             ,125
      E=0.001                                                           ,126
      M=3                                                               ,127
      MAX=1                                                             ,128
      N=2                                                               ,129
      CALL POLXN (C,P,M,N,E,MAX,Q,LOC,NINT)                             ,130
      IF (NINT) 230,190,185                                             ,131
  185 XINT(2,J)=Q(1,1)                                                  ,132
      YINT(2,J)=Q(2,1)                                                  ,133
      ZINT(2,J)=Q(3,1)                                                  ,134
      KPANEL(2,J)=1                                                     ,135
      GO TO 200                                                         ,136
  190 P(1,1)=XCOR(NPLN1,J+1)                                            ,137
      P(2,1)=YCOR(NPLN1,J+1)                                            ,138
      P(3,1)=ZCOR(NPLN1,J+1)                                            ,139
      P(1,2)=XCOR(NPLANE,J+1)                                           ,140
      P(2,2)=YCOR(NPLANE,J+1)                                           ,141
      P(3,2)=ZCOR(NPLANE,J+1)                                           ,142
      C(4)=YCOR(NPLANE,J)

C     ATTEMPT LEADING-EDGE INTERSECTION                                 ,143
      CALL POLXN (C,P,M,N,E,MAX,Q,LOC,NINT)                             ,144
      IF (NINT) 230,240,195                                             ,145
  195 XINT(2,J)=Q(1,1)                                                  ,146
      YINT(2,J)=Q(2,1)                                                  ,147
      ZINT(2,J)=Q(3,1)                                                  ,148
      KPANEL(2,J)=2                                                     ,149
  200 CONTINUE                                                          ,150  ,151
      DO 205 J=1,NPER1                                                  ,152
      IF (YINT(2,J)-YCOR(NPLN1,J)) 260,205,205                          ,153
  205 CONTINUE                                                          ,154  ,155
  210 IF (KOPTW) 300,300,270

C     ERROR MESSAGES                                                    ,156
  230 WRITE (NWRITE,900)                                                ,157  ,158
      GO TO 265                                                         ,159
  240 WRITE (NWRITE,910)                                                ,160  ,161
      GO TO 265                                                         ,162
  250 WRITE (NWRITE,920)                                                ,163  ,164
      GO TO 265                                                         ,165
  260 WRITE (NWRITE,930)                                                ,166  ,167
      GO TO 265                                                         ,168
  265 KODEW=1                                                           ,169
      GO TO 300                                                         ,170
  270 IF (KEND-NPLN1) 280,290,300                                       ,171
  280 J=NPLANE                                                          ,172
      SLOPE=(XCOR(NPER,J)-XCOR(1,J))/(YCOR(NPER,J)-YCOR(1,J))           ,173
      GO TO 300                                                         ,174
  290 SLOPE=0.                                                          ,175
  300 RETURN                                                            ,176
  900 FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CRNRW/10X,
     141HPANEL (PART-) CORNER POINT NOT CALCULATED/10X,
```

```
      124HSUBROUTINE POLXN FAILURE/10X,
      152HINTERSECTION NOT CALCULATED DUE TO ZERO COEFFICIENTS)
910    FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CRNRW/10X,
      141HPANEL (PART-) CORNER POINT NOT CALCULATED/10X,
      124HSUBROUTINE POLXN FAILURE/10X,
      127HINTERSECTION DOES NOT EXIST)
920    FORMAT (1H1,9X,16HDATA ERROR/10X,
      135HINCORRECT WING PANEL (FIRST COLUMN))
930    FORMAT (1H1,9X,16HDATA ERROR/10X,
      134HINCORRECT WING PANEL (LAST COLUMN))
       END                                                          ,177
```

```
      SUBROUTINE CVEL(M,N,NTAPEX,A,B,C,CL,U,V,W)
C     COMPUTES VELOCITY COMPONENTS FOR A GIVEN PANEL PRESSURE DIFFERENCE
      DIMENSION A(1),B(1),C(1),U(1),V(1),W(1),CL(1)
      DO 100 J=1,M                                                      ,1
      U(J)=0.0                                                          ,2
      V(J)=0.0                                                          ,3
      W(J)=0.0                                                          ,4
  100 CONTINUE
                                                                        ,5     ,6
      DO 200 J=1,N                                                      ,7
      READ (NTAPEX) (A(I),B(I),C(I),I=1,M)                              ,8     ,9     ,10    ,11    ,12
      DO 200 I=1,M                                                      ,13
      U(I)=U(I)+A(I)*CL(J)                                              ,14
      V(I)=V(I)+B(I)*CL(J)                                              ,15
      W(I)=W(I)+C(I)*CL(J)                                              ,16
  200 CONTINUE
                                                                        ,17    ,18    ,19
      RETURN                                                            ,20
      END                                                               ,21
```

```
      SUBROUTINE DCPD(NM,NTAPEX,A,ALPHAM,CLM)

C     COMPUTES WING PANEL PRESSURE DIFFERENCE

      DIMENSION A(1),ALPHAM(1),CLM(1)


      DO 100 J=1,NM                                                    ,1
      CLM(J)=0.0                                                       ,2
  100 CONTINUE
                                                                       ,3    ,4
      DO 200 J=1,NM                                                     5
      READ (NTAPEX) (A(I),I=1,NM)                                      ,6    ,7    ,8    ,9    ,10
      DO 200 I=1,NM                                                    ,11
      CLM(I)=CLM(I)+A(I)*ALPHAM(J)                                     ,12
  200 CONTINUE
                                                                       ,13   ,14   ,15
      RETURN                                                           ,16
      END                                                              ,17
```

```
      SUBROUTINE DCPI (NB,NW,NTAPEX,A,ALPHAB,CLW,CLB)

C     COMPUTES BODY PANEL PRESSURE DIFFERENCE

      DIMENSION A(1),ALPHAB(1),CLW(1),CLB(1)

      DO 100 J=1,NB                                                    ,1
      CLB(J)=0.                                                        ,2
  100 CONTINUE                                                         ,3    ,4
                                                                       ,5
      DO 200 J=1,NB                                                    ,6    ,7    ,8    ,9    ,10
      READ (NTAPEX) (A(I),I=1,NB)                                      ,11
      DO 200 I=1,NB                                                    ,12
      CLB(I)=CLB(I)+A(I)*ALPHAB(J)                                     ,13   ,14   ,15
  200 CONTINUE                                                         ,16
      DO 300 J=1,NW                                                    ,17   ,18   ,19   ,20   ,21
      READ (NTAPEX) (A(I),I=1,NB)                                      ,22
      DO 300 I=1,NB                                                    ,23
      CLB(I)=CLB(I)+A(I)*CLW(J)
  300 CONTINUE                                                         ,24   ,25   ,26
                                                                       ,27
      RETURN                                                           ,28
      END
```

```
      SUBROUTINE DEFEN1(B,J,K,LTAPE,IREC)
      DIMENSION  B(1),IREC(10),JREC(10),DUM(9)
      DATA  DUM/9*0./,JREC/0,9,8*0/
C
C     USED BY DEFEND TO WRITE DEFINITION TAPE
C
      IF (IREC(3))10,20,10
C                                                                  ,1
 10   JREC=IREC                                                    ,2
      JREC(3)=IREC(3)                                              ,3
      WRITE(LTAPE)JREC                                             ,4    ,5    ,6
      WRITE(LTAPE)DUM                                              ,7    ,8    ,9
      GO TO 100
C                                                                  ,10
 20   WRITE(LTAPE)IREC                                             ,11   ,12   ,13
      WRITE(LTAPE)(B(I),I=J,K)                                     ,14   ,15   ,16   ,17   ,18
 100  RETURN                                                       ,19
      END                                                          ,20
```

```
      SUBROUTINE DEFEND(B,LK,LO,LTAPE,LGDEF,LGTAPE,INTWB)
      DIMENSION B(1),LK(3,9),LOK(3,9),IREC(10),KR(6),LGDEF(3,6)
      LOGICAL  LGDEF,LGTAPE
      EQUIVALENCE  (LOK(1,1),LUWPL),(LOK(2,1),MUWPL),(LOK(3,1),NUWPL),
     X(LOK(1,2),LLWPL),(LOK(2,2),MLWPL),(LOK(3,2),NLWPL),
     X(LOK(1,3),LUWPLX),(LOK(2,3),MUWPLX),(LOK(3,3),NUWPLX),
     X(LOK(1,4),LLWPLX),(LOK(2,4),MLWPLX),(LOK(3,4),NLWPLX),
     X(LOK(1,5),LBPL),(LOK(2,5),MBPL),(LOK(3,5),NBPL),
     X(LOK(1,6),LSTA),(LOK(3,6),NSTA),
     X(LOK(1,7),LWPF),(LOK(2,7),MWPF),(LOK(3,7),NWPF),
     X(LOK(1,8),LUWB),(LOK(3,8),NUWB),
     X(LOK(1,9),LLWB),(LOK(3,9),NLWB)
      DATA  IREC/10*0/
C
C     SPECIAL SUBROUTINE USED BY GEOMD
C     WRITES DEFINITION TAPE FOR USE BY PANEL LINK
C
 90   IF (LGTAPE) GO TO 200                                          ,1    ,2    ,3
      WRITE (LO,110)LTAPE                                            ,4    ,5    ,6
 110  FORMAT(50H1DID NOT WRITE DEFINITION TAPE.  LOGICAL TAPE NO.
     .3HWAS I10)
      GO TO 9000
C
C     WE ALWAYS WRITE 18 RECORDS ON TAPE LTAPE, BUT IT MAY TAKE 2 CALLS OF
C     DEFEND TO DO IT.  IF A WING-BODY INTERSECTION IS CALLED FOR, WE WRITE
C     THE BODY AND WING RECORDS BEFORE FINDING THE INTERSECTIONS BECAUSE THE
C     ORIGINAL WING STORAGE IS RE-USED.   THE REMAINING RECORDS ARE WRITTEN
C     LATER, ON A SECOND CALL.  INTWB IS A CODE WHICH CONTROLS THIS.

C     THE 3RD WORD OF COD-NUMBERED RECORDS IS =0 IF OK, =1 IF NOT WANTED,
C     =-1 IF WANTED BUT ERROR OCCURRED.  WE WRITE DUMMY RECORDS IF NOT =0.
C     SET UP THIS WORD IN KR ARRAY.                                  ,7
 200  DO 500 I=1,6                                                   ,8
      IF (LGDEF(1,I).AND.LGDEF(2,I)) GO TO 400                       ,9    ,10   ,11
 300  IF (LGDEF(1,I)) GO TO 310                                      ,12   ,13   ,14
      KR(I)=1                                                        ,15
      GO TO 500                                                      ,16
 310  KR(I)=-1                                                       ,17
      GO TO 500
C                                                                    ,18
 400  KR(I)=0                                                        ,19
      LGDEF(3,I)=.TRUE.                                              ,20
 500  CONTINUE
C                                                                    ,21   ,22
      DO 600 J=1,9                                                   ,23
      DO 600 I=1,3                                                   ,24
 600  LOK(I,J)=LK(I,J)
C                                                                    ,25   ,26   ,27
      IF (INTWB)650,650,2000                                         ,28
 650  REWIND LTAPE
C
C     BODY STATIONS                                                  ,29
      IREC(1)=1                                                      ,30
```

```
      IREC(2)=NSTA                                                      ,31
      IREC(3)=KR(3)
C                                                                       ,32
      CALL DEFEN1(B,LSTA,LSTA+NSTA-1,LTAPE,IREC)
C
C     BASIC BODY PERCENT LINES                                          ,33
      IREC(1)=3                                                         ,34
      IREC(2)=NBPL                                                      ,35
      IREC(3)=KR(3)                                                     ,36
      IREC(4)=MBPL                                                      ,37
      CALL DEFEN1(B,LBPL,LBPL+NBPL-1,LTAPE,IREC)
C
C     WING PLANFORM (RECORDS 5,6)                                       ,38
      IREC(1)=5                                                         ,39
      IREC(2)=NWPF                                                      ,40
      IREC(3)=KR(4)                                                     ,41
      IREC(4)=MWPF                                                      ,42
      CALL DEFEN1(B,LWPF,LWPF+NWPF-1,LTAPE,IREC)
C
C     BASIC WING PERCENT LINES (RECORDS 7,8,9,10)                       ,43
      DO 1000 IA=1,2                                                    ,44
      IREC(1)=IA+IA+5                                                   ,45
      IREC(2)=LOK(3,IA)                                                 ,46
      IREC(3)=KR(IA)                                                    ,47
      IREC(4)=LOK(2,IA)                                                 ,48
 1000 CALL DEFEN1(B,LOK(1,IA),LOK(1,IA)+LOK(3,IA)-1,LTAPE,IREC)
C                                                                       ,49    ,50
 2000 IF (INTWB)9000,2100,2100
C
C     UPPER WING-BODY INTERSECTION POINTS (RECORDS 11,12)               ,51
 2100 IREC(1)=11                                                        ,52
      IREC(2)=NUWB                                                      ,53
      IREC(3)=KR(5)                                                     ,54
      IREC(4)=MUWPLX                                                    ,55
      CALL DEFEN1 (B,LLWB,LUWB+NUWB-1,LTAPE,IREC)
C
C     UPPER WING OUTSIDE BODY (RECORDS 13,14)                           ,56
      IREC(1)=13                                                        ,57
      IREC(2)=NUWPLX                                                    ,58
      CALL DEFEN1 (B,LUWPLX,LUWPLX+NUWPLX-1,LTAPE,IREC)
C
C     LOWER WING-BODY INTERSECTION POINTS (RECORDS 15,16)               ,59
      IREC(1)=15                                                        ,60
      IREC(2)=NLWB                                                      ,61
      IREC(3)=KR(6)                                                     ,62
      IREC(4)=MLWPLX                                                    ,63
      CALL DEFEN1 (B,LLWB,LLWB+NLWB-1,LTAPE,IREC)
C
C     LOWER WING OUTSIDE BODY (RECORDS 17,18)                           ,64
      IREC(1)=17                                                        ,65
      IREC(2)=NLWPLX                                                    ,66
      CALL DEFEN1 (B,LLWPLX,LLWPLX+NLWPLX-1,LTAPE,IREC)
C                                                                       ,67
      END FILE LTAPE                                                    ,68
```

```
                                                                                09/14/65
          DEFEND
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)
                                                                                  ,69
9000 RETURN                                                                       ,70
     END
```

```
      FUNCTION DISPTA(P,Q,J)
C
C     FIND DISTANCE FROM POINT P TO Q
C     J = DIMENSION OF P AND Q
      DIMENSION P(1),Q(1)
C
      D=0.                                                          ,1
      DO 10 I=1,J                                                   ,2
      U=P(I)-Q(I)                                                   ,3
   10 D=D+U*U                                                       ,4     ,5
      DISPTA=SQRT (D)                                               ,6
      RETURN                                                        ,7
      END                                                           ,8
```

```
      SUBROUTINE DMAXL(P,N,DIST,J,U,NU)
      DIMENSION P(3,1),U(3),V(3),UC(3)
C
C     GIVEN AN ARRAY P OF N POINTS IN 3-SPACE.
C     FIND WHICH POINT IN P IS FARTHEST FROM THE LINE THRU END POINTS,
C     AND ITS DISTANCE FROM THE LINE.  IF MORE THAN ONE POINTS ARE
C     EQUALLY FAR, CHOOSE 1ST SUCH POINT IN P.
C     THE DESIRED POINT IS P(1,J)
C     NU=0 IF SUCCESS, = -1 IF N .LT. 2, =-2 IF END POINTS OF P
C       ARE COINCIDENT, =1 IF N=2.
C     INPUTS ARE P,N.  OUTPUTS ARE DIST,J,U,NU
C
      NU=0                                                              ,1
      DP=0.                                                             ,2
      J=0                                                               ,3
      IF(N-2)100,200,300                                                ,4
 100  NU=-1                                                             ,5
      GO TO 2000
C                                                                       ,6
 200  NU=1                                                              ,7
      GO TO 2000
C                                                                       ,8
 300  NM=N-1                                                            ,9
      IF(UVECN(P,P(1,N),U,0,3))400,400,500
C     U IS A UNIT VECTOR FROM P(1,1) TO P(1,N)                          ,10
 400  NU=-2                                                             ,11
      GO TO 2000
C                                                                       ,12
 500  DO 1000 K=2,NM
C                                                                       ,13
      DO 600 I=1,3                                                      ,14
 600  V(I)=P(I,K)-P(I,1)
C     V IS A VECTOR FROM P(1,1) TO THE K-TH POINT                       ,15   ,16
C                                                                       ,17
      KE=0
      CALL VCROS(U,V,UC,D,KE)
C     UC IS A VECTOR, THE CROSS-PRODUCT U X V.                          ,18
C     D IS THE MAGNITUDE OF UC AND IS DISTANCE FROM LINE TO POINT       ,19
      IF(D-DP)1000,1000,700                                             ,20
 700  J=K
      DP=D                                                              ,21
C                                                                       ,22   ,23
 1000 CONTINUE                                                          ,24
 2000 DIST=DP                                                           ,25
 9000 RETURN                                                            ,26
      END
```

```
      SUBROUTINE ELLIPR(A,B,SIN,COS,RHO,X,Y,NU)
C
C     GIVEN A RECTANGULAR ELLIPSE WITH MAJOR, MINOR SEMI-AXES A,B
C     AND CENTER AT THE ORIGIN.
C     GIVEN THE SIN AND COS OF AN ANGLE THETA MEASURED
C     CLOCKWISE FROM THE +Y AXIS.
C
C     FIND RHO, THE LENGTH OF A RADIUS VECTOR FROM ORIGIN ALONG THETA
C     TO THE ELLIPSE.
C     FIND (X,Y), THE CORRESPONDING POINT ON THE ELLIPSE.
C
      IF(A*B)200,100,200                                                ,1
  100 NU=1                                                              ,2
      GO TO 9000                                                        ,3
  200 U=(SIN/A)**2+(COS/B)**2                                           ,4
      IF(U)300,300,400                                                  ,5
  300 NU=2                                                              ,6
      GO TO 9000                                                        ,7
  400 RHO=1./SQRT(U)                                                    ,8
      X=RHO*SIN                                                         ,9
      Y=RHO*COS                                                         ,10
      NU=0                                                              ,11
 9000 RETURN                                                            ,12
      END                                                               ,13
```

```
      SUBROUTINE ENRYCH(A,B,NI,EPS,C,NC,NU)
      DIMENSION A(500),B(500), C(10000),D(4),E(4),F(4),G(4),W(200),Z(200ENRI0150
     1),NU(3)
C                                                                       ENRI0110
C     J. E. BROWN                          ASSEMBLED 12-15-64
C     USES OPTIM3 TO CONVERT SPARSE ARRAY A, B                          ENRI0120
C     TO DENSE ARRAY C.                                                 ENRI0130
C                                                                       ENRI0140
C     A = ARRAY OF X VALUES
C     B = ARRAY OF Y VALUES
C     NI = NUMBER OF INPUT POINTS
C     EPS = CHORD HEIGHT TOLERANCE
C     C = (X,Y,X,Y,....)
C     NC (INPUT) = LENGTH OF C
C     NC (OUTPUT) =NUMBER OF ELEMENTS PLACED IN C
C     NU = 0 IF SUCCESS
C     NU = 1 IF OPTIM3 ERROR WAS =0 (NOT POSSIBLE)
C     NU = 2 IF 4 CONSEC. POINTS NOT MONOTONIC IN EITHER X OR Y
C     NU = 3 IF OPTIM3 GENERATED 199 POINTS BETWEEN 2 GIVEN PTS AND NOT DONE.
C     NU = 4 IF LENGTH OF C (=NC) WAS TOO SMALL
C     NU = 5 IF NC LESS THAN 2*NI

C     NU(2), AN INPUT, IS THE LOGICAL NUMBER OF AN OUTPUT TAPE ON WHICH TO WRITE
C     ERROR COMMENTS, IF ANY.  NU(3), BOTH AN INPUT AND OUTPUT, IS AN ERROR
C     MESSAGE LIMITER AND COUNTER.  IF AN ERROR OCCURS FOR WHICH A MESSAGE IS
C     NORMALLY WRITTEN, NU(3) IS FIRST REDUCED BY ONE.  THE MESSAGE IS THEN
C     WRITTEN ONLY IF BOTH NU(2) AND NU(3) ARE GREATER THAN ZERO.
C                                                                       ENRI0190
      NU=0                                                                      ,1
      NOM=NC                                                                    ,2
      NC=0                                                                      ,3
      NII=NI+NI
                                                                                ,4
      IF (NII-NOM)1,1,3                                                         ,5
    1 IF (NI-4)500,2,2                                                          ,6
    2 IF (EPS)500,500,10                                                        ,7
    3 NU=5                                                                      ,8
      GO TO 1000
                                                                                ,9
   10 I1=1                                                              ENRI0260  ,10
      I2=1                                                              ENRI0270  ,11
      I3=1                                                              ENRI0280  ,12
      KK=1                                                              ENRI0290
                                                                                ,13
   20 DO 320 I=I1,I2                                                    ENRI0300
                                                                                ,14
   30 DO 35 J=1,4                                                       ENRI0310  ,15
      I4=I+J-1                                                          ENRI0320  ,16
      D(J)=A(I4)                                                        ENRI0330  ,17
   35 E(J)=B(I4)                                                        ENRI0340
                                                                                ,18    ,19
```

```
      L1=0                                                              ENRI0350   ,20
   40 IF (ABS(D(4)-D(1))-ABS(E(4)-E(1)))100,50,50                       ENRI0360   ,21
   50 IF (D(2)-D(1))80,95,60                                            ENRI0370   ,22
   60 IF (D(3)-D(2))95,95,70                                            ENRI0380   ,23
   70 IF (D(4)-D(3))95,95,200                                           ENRI0390   ,24
   80 IF (D(3)-D(2))90,95,95                                            ENRI0400   ,25
   90 IF (D(4)-D(3))210,95,95                                           ENRI0410   ,26
   95 IF (L1)160,97,160                                                 ENRI0420   ,27
   97 L1=1                                                              ENRI0430   ,28
  100 IF (E(2)-E(1))130,150,110                                         ENRI0440   ,29
  110 IF (E(3)-E(2))150,150,120                                         ENRI0450   ,30
  120 IF (E(4)-E(3))150,150,220                                         ENRI0460   ,31
  130 IF (E(3)-E(2))140,150,150                                         ENRI0470   ,32
  140 IF (E(4)-E(3))230,150,150                                         ENRI0480   ,33
  150 IF (L1)160,152,160                                                ENRI0490   ,34
  152 L1=1                                                              ENRI0500   ,35
      GO TO 50                                                          ENRI0510
C     NOT MONOTONIC IN EITHER X OR Y                                    ENRI0520   ,36
  160 NU=2                                                                         ,37
      L=1                                                               ENRI0540   ,38
      GO TO 316                                                         ENRI0550
C                                                                       ENRI0560   ,39
  200 DO 205 J=1,4                                                      ENRI0570   ,40
      F(J)=D(J)                                                         ENRI0580   ,41
  205 G(J)=E(J)                                                         ENRI0590   ,42   ,43
      MX=1                                                              ENRI0600   ,44
      GO TO 240                                                         ENRI0610   ,45
  210 DO 215 J=1,4                                                      ENRI0620   ,46
      F(J)=-D(J)                                                        ENRI0630   ,47
  215 G(J)=E(J)                                                         ENRI0640   ,48   ,49
      MX=2                                                              ENRI0650   ,50
      GO TO 240                                                         ENRI0660   ,51
  220 DO 225 J=1,4                                                      ENRI0670   ,52
      F(J)=E(J)                                                         ENRI0680   ,53
  225 G(J)=D(J)                                                         ENRI0690   ,54   ,55
      MX=3                                                              ENRI0700   ,56
      GO TO 240                                                         ENRI0710   ,57
  230 DO 235 J=1,4                                                      ENRI0720   ,58
      F(J)=-E(J)                                                        ENRI0730   ,59
  235 G(J)=D(J)                                                         ENRI0740   ,60   ,61
      MX=4                                                              ENRI0750   ,62
  240 CONTINUE                                                          ENRI0760
                                                                                   ,63
      CALL OPTIM3 (F,G,EPS,W,Z,KK,L,MU)                                 ENRI0770
C     MU = 0 IF F(1)=F(4)  (NOT POSSIBLE)
C     MU = 1 IF SUCCESS
C     MU = 2 IF 139 PTS. GENERATED AND NOT DONE                                  ,64
  250 IF(MU-1)251,260,251                                               ENRI0780   ,65
  251 NU=MU+1                                                                      ,66
      GO TO 316                                                         ENRI0800
                                                                                   ,67
  260 IF(L-1)275,275,265                                                ENRI0810   ,68
  265 IF ((F(I3+1)-W(L))/(W(L)-W(L-1))-.2) 270,270,275                             ,69
  270 L=L-1                                                             ENRI0830   ,70
```

```
  275 TA=1.0                                                        ENRI0840   ,71
      IF(MX-3)280,285,285                                           ENRI0850   ,72
  280 NA=1                                                          ENRI0860   ,73
      NB=0                                                          ENRI0870   ,74
      IF(MX-2)300,290,300                                           ENRI0880   ,75
  285 NA=0                                                          ENRI0890   ,76
      NB=1                                                          ENRI0900   ,77
      IF(MX-4)300,290,300                                           ENRI0910   ,78
  290 TA=-1                                                         ENRI0920
                                                                                 ,79
  300 NOM=NOM-L-L                                                                ,80
      IF (NOM)365,305,305
                                                                                 ,81
  305 DO 315 J=1,L                                                               ,82
      NO=NO+2                                                       ENRI0940   ,83
      JA=NO-NA                                                      ENRI0950   ,84
      JB=NO-NB                                                      ENRI0960   ,85
      C(JA)=W(J)*TA                                                 ENRI0970   ,86
  315 C(JB)=Z(J)                                                    ENRI0980
                                                                                 ,87   ,88
      GO TO 320                                                     ENRI0990

C     ERROR AT 160 OR 250.  WRITE AND GO ON, WITHOUT ADDING NEW POINTS. ENRI1000   ,89
  316 NU(3)=NU(3)-1                                                              ,90
      IF (NU(3))321,321,317                                                      ,91
  317 LTAPE=NU(2)                                                                ,92
      IF (LTAPE)321,321,318
                                                                                 ,93
  318 WRITE (LTAPE,319)NU(1),EPS,NI,D,E                                          ,94   ,95   ,96
  319 FORMAT(18H0ENRYCH ERROR CODE I2,34H FOR THE FOUR POINTS BELOW.  EP
     1S = E12.4, 6H, NI = I4/4H X =4F20.9/4H Y =4F20.9)             ENRI1030
  321 NOM=NOM-2                                                                  ,97
      IF (NOM)365,323,323                                                        ,98
  323 NO=NO+1                                                                    ,99
      C(NO)=D(I3)                                                   ENRI1050   ,100
      NO=NO+1                                                       ENRI1060   ,101
      C(NO)=E(I3)                                                   ENRI1070
                                                                                 ,102
  320 CONTINUE                                                      ENRI1080
                                                                                 ,103  ,104
  330 IF (I3-2)340,350,360                                          ENRI1090   ,105
  340 I2=NI-3                                                       ENRI1100   ,106
      KK=2                                                          ENRI1110   ,107
      I3=2                                                          ENRI1120   ,108
      GO TO 20                                                      ENRI1130   ,109
  350 I1=I2                                                         ENRI1140   ,110
      KK=3                                                          ENRI1150   ,111
      I3=3                                                          ENRI1160   ,112
      GO TO 20                                                      ENRI1170   ,113
  360 IF (NOM-2)365,370,370                                                      ,114
  365 NU=4                                                                       ,115
      GO TO 1000
                                                                                 ,116
  370 C(NO+1)=A(NI)                                                              ,117
```

```
EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      C(NO+2)=B(NI)                                          ENRI1190    ,118
      NO=NO+2                                                ENRI1200    ,119
  420 GO TO 1000
                                                                         ,120
  500 DO 510 I=1,NI                                                      ,121
      J=I+I                                                              ,122
      C(J-1)=A(I)                                                        ,123
  510 C(J)=B(I)                                                          ,124  ,125
      NO=NII
                                                                         ,126
 1000 RETURN                                                             ,127
      END                                                    ENRI1220    ,128
```

```
      SUBROUTINE EVAL (NI,XCPT,THKW)

C     COMPUTES AERODYNAMIC INFLUENCE COEFFICIENTS MATRIX AND VELOCITY
C     COMPONENTS MATRIX

      ACOSH(U)=2.*ALOG(SQRT(ABS((U+1.)/2.))+SQRT(ABS((U-1.)/2.)))

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KALE

      COMMON /BLOCK/X(200,3,4),Y(200,3,4),Z(200,3,4),XBAR(200)
     1,YBAR(200),ZBAR(200),XC(200),YC(200),ZC(200),THETA(200),ALPHAS(200
     2),AREA(200),CHORD(200),NPART(200),NPANEL,ISYM,A(200),U(200),V(200)
     3,VPM(200),VV(200),W(200),WPM(200),WW(200),VPMM(200),WPMM(200),ALPH
     1AC(100),ALPHAT(100)

      DIMENSION BBETAM(4),R(4),C(4),Q(4),CPM(4),RPM(4),CPMM(4),RPMM(4)
     1,XCPT(200)

      EPS=0.00001                                                       ,1
      BETA=SQRT(XMACH*XMACH-1.0)                                        ,2
      PI=3.1415926                                                      ,3
      CONSTA=BETA/4.0                                                   ,4
      CONSTB=CONSTA/PI

C     INFLUENCING PANEL I
                                                                        ,5
      DO 900 I=NI,NPANEL                                                ,6
      COST=COS(THETA(I))                                                ,7
      SINT=SIN(THETA(I))                                                ,8
      TANA= TAN(ALPHAS(I))                                              ,9
      ABR=BETA*TANA                                                     ,10
      IF (ABS(ABR)-EPS) 80,80,85                                        ,11
 80   ABR=0.0                                                           ,12
      SA=1.0                                                            ,13
      GO TO 90                                                          ,14
 85   SA=SIGN(1.,ABR)                                                   ,15
 90   APM2=ABR*ABR                                                      ,16
      NPT=NPART(I)

C     INFLUENCED PANEL J
                                                                        ,17
      DO 850 J=1,NPANEL                                                 ,18
      U(J)=0.0                                                          ,19
      V(J)=0.0                                                          ,20
      W(J)=0.0                                                          ,21
      VPM(J)=0.                                                         ,22
      WPM(J)=0.                                                         ,23
      VPMM(J)=0.                                                        ,24
      WPMM(J)=0.                                                        ,25
      SINA=SIN(ALPHAS(J))                                               ,26
```

```
      COSA=COS(ALPHAS(J))                                                ,27
      DO 850M=1,NPT                                                      ,28
      BBETAM(1)=(X(I,M,2)-X(I,M,1))/(BETA*((Y(I,M,2)-Y(I,M,1))*COST
     1+(Z(I,M,2)-Z(I,M,1))*SINT))                                        ,29
      BBETAM(3)=(X(I,M,4)-X(I,M,3))/(BETA*((Y(I,M,4)-Y(I,M,3))*COST
     1+(Z(I,M,4)-Z(I,M,3))*SINT))                                        ,30
      BBETAM(2)=BBETAM(1)                                                ,31
      BBETAM(4)=BBETAM(3)

C     CORNER POINTS K OF INFLUENCING PANEL I
                                                                         ,32
      DO 800K=1,4                                                        ,33
      R(K)=0.0                                                           ,34
      C(K)=0.0                                                           ,35
      Q(K)=0.0                                                           ,36
      RPM(K)=0.                                                          ,37
      CPM(K)=0.                                                          ,38
      RPMM(K)=0.                                                         ,39
      CPMM(K)=0.                                                         ,40
      BPM=BBETAM(K)                                                      ,41
      IF (ABS(BPM)-EPS) 105,105,100                                      ,42
 100  SM=SIGN(1.0,BPM)                                                   ,43
      GO TO 110                                                          ,44
 105  SM=1.0                                                             ,45
      BPM=0.0                                                            ,46
 110  IF (ABS(ABS(BPM)-1.0)-EPS) 115,115,120                             ,47
 115  BPM=SM                                                             ,48
 120  BPM=SM*BPM                                                         ,49
      BPM2=BPM*BPM                                                       ,50
      CONSTC=1.+APM2*BPM2

C     CONFIGURATION SYMMETRY CONDITION
                                                                         ,51
      DO 800ISIDE=1,ISYM                                                 ,52
      SN=(-1.0)**ISIDE                                                   ,53
      DELTAT=THETA(I)+THETA(J)*SN                                        ,54
      SINDT=SIN(DELTAT)                                                  ,55
      COSDT=COS(DELTAT)                                                  ,56
      DELTAY=SN*(-YC(J))-Y(I,M,K)                                        ,57
      DELTAZ=ZC(J)-Z(I,M,K)                                              ,58
      XPM=XCPT(J)-X(I,M,K)                                               ,59
      YPM=(DELTAY*COST+DELTAZ*SINT)*SM                                   ,60
      ZBR=DELTAZ*COST-DELTAY*SINT                                        ,61
      IF (ABS(ZBR)-EPS) 140,140,145                                      ,62
 140  ZBR=0.0                                                            ,63
 145  SZ=SIGN(1.,ZBR)                                                    ,64
      ZPM2=ZBR*ZBR                                                       ,65
      IF (ABS(YPM)-EPS) 150,150,155                                      ,66
 150  YPM=0.0                                                            ,67
 155  SY=SIGN(1.,YPM)                                                    ,68
      XIPM=XPM/BETA                                                      ,69
      XIPM2=XIPM*XIPM                                                    ,70
      YPM2=YPM*YPM                                                       ,71
```

```
      IF (THKW) 700,190,700                                             ,72
190   CONTINUE                                                          ,73
      IF (ABR) 195,499,198
C     VELOCITY COMPONENTS INDUCED BY PANELS NON-PARALLEL TO X- AXIS     ,74
                                                                        ,75
195   APM=-ABR                                                          ,76
      ZPM=-ZBR                                                          ,77
      GO TO 199                                                         ,78
198   APM=ABR                                                           ,79
      ZPM=ZBR                                                           ,80
199   CONTINUE                                                          ,81
      IF (XIPM-SQRT(YPM2+ZPM2)-EPS) 200,200,250                         ,82
200   IF (BPM-1./SQRT(1.-APM2)) 205,550,550                             ,83
205   TERMA=SQRT(1.-BPM2*(1.-APM2))                                     ,84
      TERMB=ZPM-APM*BPM*YPM                                             ,85
      TERMC=YPM+APM*BPM*ZPM                                             ,86
      TERMD=1.+APM2*BPM2                                                ,87
      IF (((BPM*TERMC+ABS(TERMB)*TERMA)/TERMD)-XIPM)210,210,550         ,88
210   IF (ZPM-APM*XIPM+EPS) 225,215,215                                 ,89
215   IF (YPM-BPM*XIPM*(1.-APM*TERMA)/TERMD)550,550,220                 ,90
220   P=CONSTA                                                          ,91
      IF (ABS(ZPM-APM*XIPM)-EPS) 221,221,222                            ,92
221   P=P*SA                                                            ,93
222   D=CONSTA/CONSTC*(APM*BPM+TERMA)*SA                                ,94
223   S=CONSTA/CONSTC*BPM*(1.-APM*TERMA)                                ,95
      GO TO 795                                                         ,96
225   IF (YPM-BPM*XIPM*(1.+APM*TERMA)/TERMD)550,550,230                 ,97
230   P=-CONSTA                                                         ,98
      IF (TERMB) 235,240,235                                            ,99
235   S=-CONSTA/CONSTC*BPM*(1.+APM*TERMA)                               ,100
      D=CONSTA/CONSTC*(-APM*BPM2+TERMA)*SA                              ,101
      GO TO 795                                                         ,102
240   S=-CONSTA/CONSTC*APM*TERMA*BPM                                    ,103
      D=CONSTA/CONSTC*TERMA*SA                                          ,104
      GO TO 795                                                         ,105
250   TERMA=BPM*YPM-XIPM                                                ,106
      IF (ABS(TERMA)-EPS) 255,255,260                                   ,107
255   TERMA=0.0                                                         ,108
260   TERMA2=TERMA*TERMA                                                ,109
      TERMB=APM*BPM*YPM-ZPM                                             ,110
      IF (ABS(TERMB)-EPS) 262,262,265                                   ,111
262   TERMB=0.0                                                         ,112
      STERMB=1.0                                                        ,113
      GO TO 270                                                         ,114
265   STERMB=SIGN(1.0,-TERMB)                                           ,115
270   TERMB2=TERMB*TERMB                                                ,116
      TERMC=ZPM-APM*XIPM                                                ,117
      IF (ABS(TERMC)-EPS)275,275,280                                    ,118
275   TERMC=0.0                                                         ,119
      STERMC=1.0                                                        ,120
      GO TO 285                                                         ,121
280   STERMC=SIGN(1.0,TERMC)                                            ,122
285   TERMC2=TERMC*TERMC
```

```
      TERMD=1.-APM2                                                     ,123
      TERME=TERMA2+BPM2*TERMC2-TERMB2                                   ,124
      TERMF=YPM+APM*BPM*ZPM                                             ,125
      IF (XIPM-SQRT(YPM2+ZPM2)) 550,550,288                             ,126
 288  IF (TERMC) 320,290,320                                            ,127
 290  IF (TERMA) 300,305,310                                            ,128
 300  IF (YPM) 310,305,315                                              ,129
 305  FTR1=PI/2.                                                        ,130
      GO TO 325                                                         ,131
 310  FTR1=0.0                                                          ,132
      GO TO 325                                                         ,133
 315  FTR1=PI*STERMC                                                    ,134
      GO TO 325                                                         ,135
 320  FTR1=(YPM*(TERMA-APM*TERMB)+BPM*TERMC2)/SQRT
     1((TERMC2+TERMD*YPM2)*TERME)                                       ,136
      IF (FTR1+1.0) 315,315,322                                         ,137
 322  IF (FTR1-1.0) 324,310,310                                         ,138
 324  FTR1=ACOS(FTR1)*STERMC                                            ,139
 325  IF (BPM-1./SQRT(TERMD)) 335,330,330                               ,140
 330  FTR2=ACOSH((BPM*XIPM-TERMF)/SQRT(TERME))                          ,141
      GO TO 340                                                         ,142
 335  FTR2=(BPM*XIPM-TERMF)/SQRT(TERME)                                 ,143
      IF (FTR2+1.0) 339,339,336                                         ,144
 336  IF (FTR2-1.0) 337,338,338                                         ,145
 337  FTR2=-ACOS(FTR2)                                                  ,146
      GO TO 340                                                         ,147
 338  FTR2=0.0                                                          ,148
      GO TO 340                                                         ,149
 339  FTR2=-PI                                                          ,150
 340  IF (YPM) 370,345,370                                              ,151
 345  IF (TERMC) 355,350,350                                            ,152
 350  FTR3=FTR1                                                         ,153
      GO TO 365                                                         ,154
 355  IF (ZPM) 350,350,360                                              ,155
 360  FTR3=PI+FTR1                                                      ,156
 365  FTR5=-APM2*BPM2*FTR3                                              ,157
      GO TO 415                                                         ,158
 370  IF (TERMB) 400,375,400                                            ,159
 375  IF (TERMA) 380,390,385                                            ,160
 380  IF (YPM) 385,390,395                                              ,161
 385  FTR3=0.0                                                          ,162
      GO TO 415                                                         ,163
 390  FTR3=PI/2.0                                                       ,164
      GO TO 415                                                         ,165
 395  FTR3=PI*STERMB                                                    ,166
      GO TO 415                                                         ,167
 400  FTR3=((-XIPM*TERMF+BPM*(YPM2+ZPM2))/SQRT((YPM2+ZPM2)*TERME))      ,168
      IF (FTR3+1.0) 395,395,405                                         ,169
 405  IF (FTR3-1.0) 410,385,385                                         ,170
 410  FTR3=ACOS (FTR3)*STERMB                                           ,171
 415  FTR3=FTR3*SIGN(1.,APM)                                            ,172
      FTR4=SQRT(ABS(BPM2*TERMD-1.))                                     ,173
      IF (YPM) 420,430,420                                              ,174
 420  FTR5=FTR3-FTR1*CONSTC                                             ,175
```

```
      IF (YPM) 430,430,422                                              ,176
422   IF (TERMC) 424,430,430                                            ,177
424   IF (TERMB) 426,428,430                                            ,178
426   FTR5=FTR5-2.*PI*CONSTC                                            ,179
      GO TO 430                                                         ,180
428   FTR5=FTR5-PI*(2.+ABR2*BPM2)                                       ,181
430   IF (YPM2+ZPM2) 435,440,435                                        ,182
435   FTR6=ACOSH(XIPM/SQRT(YPM2+ZPM2))                                  ,183
      GO TO 445                                                         ,184
440   FTR6=1000.                                                        ,185
445   IF (TERMC) 455,450,455                                            ,186
450   IF (YPM) 455,460,455                                              ,187
455   FTR7=ACOSH((XIPM-APM*ZPM)/SQRT(TERMC2+TERMD*YPM2))*SQRT(TERMD)    ,188
      GO TO 465                                                         ,189
460   FTR7=1000.                                                        ,190
465   FTR8=FTR6-CONSTC*FTR7                                             ,191
      S=CONSTB/CONSTC*(APM*BPM*FTR4*FTR2+BPM*FTR3+FTR8/APM)             ,192
      D=CONSTB/CONSTC*(FTR4*FTR2-BPM*FTR6+FTR5/APM)*(-SA)               ,193
      P=CONSTB*FTR1                                                     ,194
      IF (TERMC) 795,470,795                                            ,195
470   P=P*SA                                                            ,196
      GO TO 795

C     VELOCITY COMPONENTS INDUCED BY PANELS PARALLEL TO X- AXIS
                                                                        ,197
499   CONTINUE                                                          ,198
      ZPM=ZBR                                                           ,199
      TERMA=(XIPM-BPM*YPM)**2                                           ,200
      TERMB=YPM2+ZPM2                                                   ,201
      TERMC=SQRT(TERMB)                                                 ,202
      TERMD=BPM2-1.0                                                    ,203
      TERME=-TERMD                                                      ,204
      IF (BPM2-1.0) 500,560,560                                         ,205
500   IF (XIPM-TERMC)520,535,510                                        ,206
510   D=CONSTB*(SQRT(TERME)*ACOS((BPM*XIPM-YPM)/SQRT
     1(TERMA+TERMD*ZPM2))+BPM*ACOSH(XIPM/TERMC)+
     2YPM/TERMB*SQRT(XIPM2-TERMB))                                      ,207
      GO TO 575                                                         ,208
520   IF (YPM) 550,550,530                                              ,209
530   IF (XIPM-(BPM*YPM+SQRT(TERME)*ABS(ZPM))) 550,537,540              ,210
535   IF (YPM-BPM*XIPM)550,537,540                                      ,211
537   CT=.5                                                             ,212
      GO TO 545                                                         ,213
540   CT=1.0                                                            ,214
545   D=CONSTA*SQRT(TERME)*CT                                           ,215
      S=CONSTA*BPM*SZ*CT                                                ,216
      P=CONSTA*SZ*CT                                                    ,217
      GO TO 795                                                         ,218
550   D=0.0                                                             ,219
      S=0.0                                                             ,220
      P=0.0                                                             ,221
      GO TO 795                                                         ,222
560   IF (XIPM-TERMC) 550,550,570                                       ,223
570   D=CONSTB*(-SQRT(TERMD)*ACOSH((BPM*XIPM-YPM)
```

```
     1/SQRT(TERMA+TERMD*ZPM2))+BPM*ACOSH(XIPM/TERMC)
     2+YPM/TERMB*SQRT(XIPM2-TERMB))                                    ,224
 575  IF (ZPM) 580,585,580                                             ,225
 580  FACTOR=      (-XIPM*YPM+BPM*TERMB)/SQRT(TERMB*
     1(TERMA+TERMD*ZPM2))                                              ,226
      IF (FACTOR+1.) 600,600,582                                       ,227
 582  IF (FACTOR-1.) 584,590,590                                       ,228
 584  FACTOR=ACOS(FACTOR)*SZ                                           ,229
      GO TO 625                                                        ,230
 585  IF (YPM) 590,597,595                                             ,231
 590  FACTOR=0.0                                                       ,232
      GO TO 625                                                        ,233
 595  IF (XIPM-BPM*YPM) 590,597,600                                    ,234
 597  FACTOR=PI/2.                                                     ,235
      GO TO 625                                                        ,236
 600  FACTOR=PI*SZ                                                     ,237
 625  S=CONSTB*(BPM*FACTOR-ZPM/TERMB*SQRT(XIPM2-TERMB))                ,238
      P=CONSTB*FACTOR                                                  ,239
      GO TO 795                                                        ,240
 700 CONTINUE

C     VELOCITY COMPONENTS INDUCED BY WING SOURCES
                                                                       ,241
      ZPM=ZBR                                                          ,242
      TERMA=(XIPM-BPM*YPM)**2                                          ,243
      TERMB=YPM2+ZPM2                                                  ,244
      TERMC=SQRT(TERMB)                                                ,245
      TERMD=BPM2-1.0                                                   ,246
      TERME=-TERMD                                                     ,247
      IF (XIPM-TERMC) 704,704,708                                      ,248
 704 IF (TERMD) 752,550,550                                            ,249
 708  FTR1=ACOSH(XIPM/TERMC)                                           ,250
      IF (TERMD) 712,716,720                                           ,251
 712 FTR2=ACOS((BPM*XIPM-YPM)/SQRT(TERMA+TERMD*ZPM2))/SQRT(TERME)      ,252
      GO TO 724                                                        ,253
 716 FTR2=SQRT(XIPM2-TERMB)/(XIPM-YPM)                                 ,254
      GO TO 724                                                        ,255
 720 FTR2=ACOSH((BPM*XIPM-YPM)/SQRT(TERMA+TERMD*ZPM2))/SQRT(TERMD)     ,256
 724 IF (ZPM) 728,732,728                                              ,257
 728  FTR3=      (-XIPM*YPM+BPM*TERMB)/SQRT(TERMB*(TERMA+TERMD*ZPM2))  ,258
      IF (FTR3+1.0) 748,748,730                                        ,259
 730  IF (FTR3-1.0) 731,736,736                                        ,260
 731  FTR3=ACOS(FTR3)*SZ                                               ,261
      GO TO 750                                                        ,262
 732 IF (YPM) 736,736,740                                              ,263
 736 FTR3=0.                                                           ,264
      GO TO 750                                                        ,265
 740 IF (XIPM-BPM*YPM) 736,744,748                                     ,266
 744 FTR3=PI/2.                                                        ,267
      GO TO 750                                                        ,268
 748 FTR3=PI*SZ                                                        ,269
 750  P=FTR2/PI                                                        ,270
      S=-(FTR1-BPM*FTR2)/PI                                            ,271
```

```
EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      D=FTR3/PI                                                          ,272
      GO TO 795                                                          ,273
  752 IF (YPM) 550,550,756                                               ,274
  756 IF (XIPM-TERMC) 760,764,768                                        ,275
  760 IF (XIPM-(BPM*YPM+SQRT(TERME)*ABS(ZPM))) 550,768,772               ,276
  764 IF (YPM-BPM*XIPM) 550,768,772                                      ,277
  768 CT=.5                                                              ,278
      GO TO 776                                                          ,279
  772 CT=1.                                                              ,280
  776 P=1./SQRT(TERME)*CT                                                ,281
      S=BPM/SQRT(TERME)*CT                                               ,282
      D=CT*SZ
                                                                         ,283
  795 CONTINUE
                                                                         ,284
      R(K)=R(K)+S*SINDT                                                  ,285
      C(K)=C(K)+D*COSDT*SM                                               ,286
      Q(K)=Q(K)-P*SM                                                     ,287
      RPM(K)=RPM(K)+S                                                    ,288
      CPM(K)=CPM(K)+D*SM                                                 ,289
      RPMM(K)=RPMM(K)+S*(-SN)                                            ,290
      CPMM(K)=CPMM(K)+D*SM*(-SN)                                         ,291
  800 CONTINUE                                                           ,292  ,293  ,294
                                                                         ,295
      U(J)=U(J)+(Q(1)-Q(2)-Q(3)+Q(4))/BETA                               ,296
      V(J)=V(J)+(R(1)-R(2)-R(3)+R(4))                                    ,297
      W(J)=W(J)+(C(1)-C(2)-C(3)+C(4))                                    ,298
      VPM(J)=VPM(J)+(RPM(1)-RPM(2)-RPM(3)+RPM(4))                        ,299
      WPM(J)=WPM(J)+(CPM(1)-CPM(2)-CPM(3)+CPM(4))                        ,300
      VPMM(J)=VPMM(J)+(RPMM(1)-RPMM(2)-RPMM(3)+RPMM(4))                  ,301
      WPMM(J)=WPMM(J)+(CPMM(1)-CPMM(2)-CPMM(3)+CPMM(4))
  850 CONTINUE                                                           ,302  ,303  ,304
                                                                         ,305
      DO 860 J=1,NPANEL                                                  ,306
      VV(J)=VPMM(J)*COST-WPMM(J)*SINT                                    ,307
      WW(J)=WPM(J)*COST+VPM(J)*SINT                                      ,308
      A(J)=W(J)+V(J)
  860 CONTINUE                                                           ,309  ,310
                                                                         ,311  ,312  ,313  ,314  ,315
      WRITE (NTAPEA)(A(J),J=1,NPANEL)                                    ,316  ,317  ,318  ,319  ,320
      WRITE (NTAPEB)(U(J),VV(J),WW(J),J=1,NPANEL)
  900 CONTINUE
                                                                         ,321  ,322
                                                                         ,323
      END FILE NTAPEA                                                    ,324
      END FILE NTAPEB                                                    ,325
      REWIND NTAPEA
      REWIND NTAPEB
                                                                         ,326
                                                                         ,327
      RETURN                                                             ,328
      END
```

```
          FASNCS
        ASSEMBLED TEXT.

                $TEXT  FASNCS                                                                  FASN0001

                                        *ASNCS   IBJOB LIBRARY / ASIN AND ACOS                                  ASIN0010
                                        *        ADAPTED FROM SHARE SUBROUTINE B1 CL ASC1                       ASIN0020
                                        *        CONVERTED TO IBMAP / 1-9-63 / E. WRIGHT - TRANSPORT - BOEING   ASIN0030
                                        *        METHODS MEMO 7090-49                                           ASIN0040
                                        *                                                                       ASIN0050
                                        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ASIN0060
                                        *                                                                       ASIN0070
                                        *     A=ASIN (B)                                                        ASIN0080
                                        *     A=ACOS (B)                                                        ASIN0090
                                        *                                                                       ASIN0100
                                        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ASIN0110
                                                                                                                ASIN0120
                                                                                                                ASIN0130
                                                ENTRY   ASIN                                                    ASIN0140
                                                ENTRY   ACOS                                                    ASIN0150
                                                                                                                ASIN0160
                                                                                                                ASIN0170
                                        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ASIN0180

BINARY CARD (NOT PUNCHED)
        00000  0600 00 0 04013  10011  ASIN   STZ     BEANIE        ASIN ENTRY                                  ASIN0190
        00001  0020 00 C 00003  10001         TRA     PETUN                                                     ASIN0200
        00002  4625 00 0 04013  10011  ACOS   STL     BEANIE        ACOS ENTRY                                  ASIN0210
        00003  1 00000 C 00006  10001  PETUN  SAVEN                                                             ASIN0220
        00004  0774 00 4 00000  10000
        00005  0020 00 4 00001  10000
        00006  0634 00 4 07000  10011
        00007  0634 00 4 00004  10001
        00010  0634 00 4 00121  10001         SXA     LK.DR,4                                                   ASIN0230
        00011  0500 60 4 00003  10000         CLA*    3,4           GET ARGUMENT X                              ASIN0240
        00012  0601 00 C 04015  10011  ASIN3  STO     X             STORE X                                     ASIN0250
        00013  0500 00 C 00123  10001         CLA    =1.0                                                       ASIN0260
        00014  0306 00 0 04015  10011         FSM     X             1 - ABS X                                   ASIN0270
        00015  0601 00 0 04014  10011         STO     TEMP                                                      ASIN0280
        00016  4120 00 C 00073  10001         TMI     ERROR                                                     ASIN0290
        00017  0500 00 C 04015  10011         CLA     X                                                         ASIN0300
        00020  0520 00 C 04013  10011         ZET     BEANIE        WAS ENTRY ASIN OR ACOS                      ASIN0310
        00021  0020 00 0 00034  10001         TRA     ACOS1                                                     ASIN0320
        00022  0560 00 C 00124  10001  ASIN6  LDQ   =0171400000000  2(-8) = .00390625                           ASIN0330

BINARY CARD (NOT PUNCHED)
        00023  0760 00 C 00003  10000         SSP                                                               ASIN0340
        00024  0040 00 0 00027  10001         TLQ     ASIN10                                                    ASIN0350
        00025  0500 00 0 04015  10011         CLA     X             X EQUALS ARCSIN X                           ASIN0360
                          00026               RETURN  PETUN                                                     ASIN0370
        00027  0500 00 0 04015  10011  ASIN10 CLA     X                                                         ASIN0380
        00030  0560 00 C 00125  10001         LDQ    =1.57079633    PI/2                                        ASIN0390
        00031  0765 00 C 00000  10000         LRS     0             + OR - PI/2                                 ASIN0400
        00032  0760 00 C 00002  10000         CHS                   -X                                          ASIN0410
        00033  0020 00 C 00037  10001         TRA     ASIN23                                                    ASIN0420
        00034  0560 00 C 00126  10001  ACOS1  LDQ    =0             ZERO                                        ASIN0430
        00035  0120 00 0 00037  10001         TPL     ASIN23                                                    ASIN0440
```

```
          FASNCS
        ASSEMBLED TEXT.

      00036  0560 00 C 00127  10001          LDQ    =3.14159265    PI                                      ASIN0450
      00037  4600 00 C 04012  10011  ASIN23  STQ    A              A                                       ASIN0460
      00040  0601 00 C 04013  10011          STO    BEANIE         PLUS OR MINUS (B)                       ASIN0470
      00041  00CC0000C000     00010          CALL   SQRT(TEMP)     SQUARE ROOT (1-ABS X)                   ASIN0480
      00041  0074 00 4 06000  10011
      00042  1 00001 C 01003  10011
      00043  0 00121 C 00072  10100
      00044  0 00000 0 04014  10011

BINARY CARD (NOT PUNCHED)
      00045  0560 00 C 04013  10011          LDQ    BEANIE                                                 ASIN0490
      00046  0763 00 C 00000  10000          LLS    0                                                      ASIN0500
      00047  0601 00 C 04013  10011          STO    BEANIE         PLUS OR MINUS B SQRT (1-ABS X)          ASIN0510
      00050  0560 00 0 04015  10011          LDQ    X              X                                       ASIN0520
      00051  0763 00 C 00010  10000          LLS    8              GET CHARACTERISIC                       ASIN0530
      00052  0734 00 4 00000  10000          PAX    0,4            PLACE IN IR4                            ASIN0540
      00053  4754 0C C 00000  10000          ZAC                                                           ASIN0550
      00054  0765 00 4 00200  10000          LRS    128,4          SHIFT(128 - POWER)MODULO 256            ASIN0560
      00055  4600 00 0 04015  10011          STQ    X              X IN FIXED POINT                        ASIN0570
      00056  0500 00 C 00101  10001          CLA    A8             A8                                      ASIN0580
      00057  0774 CC 4 00010  10000          AXT    8,4                                                    ASIN0590
      00060  0131 00 C 00000  10000  ASIN41  XCA                                                           ASIN0600
      00061  4200 00 C 04015  10011          MPR    X              TIMES X                                 ASIN0610
      00062  0400 00 4 00112  10001          ADD    A0+1,4         PLUS A(I)                               ASIN0620
      00063  2 00001 4 00060  10001          TIX    ASIN41,4,1                                             ASIN0630
      00064  0765 00 C 00010  10000          LRS    8                                                      ASIN0640
      00065  0760 00 C 00010  10000          RND                                                           ASIN0650
      00066  0400 00 C 00130  10001          ADD    =129B8         FLOAT                                   ASIN0660
      00067  0131 00 C 00000  10000          XCA                                                           ASIN0670

BINARY CARD (NOT PUNCHED)
      00070  0260 00 C 04013  10011          FMP    BEANIE         TIMES B SQRT (1-ABS X)                  ASIN0680
      00071  0300 00 C 04012  10011          FAD    A              PLUS A                                  ASIN0690
                      00072                  RETURN PETUN          EXIT                                    ASIN0700
                                     *                                                                     ASIN0710
                                     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ASIN0720
                                     *                                                                     ASIN0730
      00073  000C00000000     00010  ERROR  CALL   .FXEM.(BAIL)   CALL ERROR EXECUTION MONITOR            ASIN0740
      00073  0074 0C 4 05000  10011
      00074  1 00001 C 01003  10011
      00075  0 00121 C 00136  10100
      00076  0 00000 C 00112  10001
      00077  0500 00 C 00123  10001          CLA    =1.0           SET ARGUMENT TO ONE                     ASIN0750
      00100  0020 00 0 00012  10001          TRA    ASIN3                                                  ASIN0760
                                     *                                                                     ASIN0770
                                     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ASIN0780
                                     *                                                                     ASIN0790
      00101  000076257520     10000   A8    OCT    76257520       A8                                      ASIN0800
      00102  400507371460     10000          DEC    -85848880      A7                                      ASIN0810
      00103  0 01472 4 40066  10000          PZE    16438,4,826    A6                                      ASIN0820
      00104  402641537160     10000          DEC    -377929328     A5                                      ASIN0830
      00105  004173275020     10000          OCT    4173275020     A4                                      ASIN0840
      00106  406374356540     10000          DEC    -871488864     A3                                      ASIN0850
      00107  013313544557     10000          OCT    13313544557    A2                                      ASIN0860
```

```
          FASNCS
       ASSEMBLED TEXT.

      00110  433360112561     10000          OCT     -033360112561 A1                                    ASIN0870
      00111  311037552421     10000   A0     OCT     311037552421  A0                                    ASIN0880

BINARY CARD (NOT PUNCHED)
      00112  000000000101     10000  BAIL    DEC     65                                                  ASIN0881
      00113  0 00002 C 00115  10001          PZE     MESA,,2                                             ASIN0882
      00114  0 00002 C 00117  10001          PZE     MESB,,2                                             ASIN0883
      00115  002151276027     10000  MESA    BCI     2,0ARG GT ONE                                       ASIN0884
      00116  636046452560     10000
      00117  002151276013     10000  MESB    BCI     2,0ARG = 1                                          ASIN0885
      00120  600160606060     10000
                                     *                                                                   ASIN0890
                                     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ASIN0900
                                     *                                                                   ASIN0910
      00121  0000C000C000     10000  LK.DR   LDIR                                                        ASIN0920
      00122  262162452362     10000
      00123  20140000C000     10000          LORG                                                        ASIN0930
      00124  17140000C000     10000
      00125  201622077325     10000
      00126  000C00000000     10000
      00127  202622077325     10000
      00130  201C0000C000     10000
                                     *                                                                   ASIN0940
                                     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ASIN0950
                                     *                                                                   ASIN0960
                                      ERASE CONTRL  ER                                                   ASIN0970
      00131  10000000C131     00001          USE     ER                                                  ASIN0980
      00131  20000000C012     00001          BSS     10            RESERVE FOR SQRT                      ASIN0990
      00143  20000000C001     00001  A       BSS     1                                                   ASIN1000
      00144  20000000C001     00001  BEANIE  BSS     1                                                   ASIN1010

BINARY CARD (NOT PUNCHED)
      00145  200000000001     00001  TEMP    BSS     1                                                   ASIN1020
      00146  20000000C001     00001  X       BSS     1                                                   ASIN1030
      00147  20000000C006     00001          BSS     6                                                   ASIN1040
                                     *                                                                   ASIN1050
                                     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ASIN1060
                                                                                                         ASIN1070
                      00000  01111          END                                                           ASIN1080
```

```
          FASNCS
        CONTROL DICTIONARY

                $CDICT FASNCS                                                    FASN0002

BINARY CARD (NOT PUNCHED)
          000155000000                PREFACE     START=0,LENGTH=109,TYPE=7094,CMPLX=5
          000004000005
          262162452362        FASNCS DECK         LOC=0,LENGTH=109
          000155000000
          216231456060        ASIN   REAL         LOC=0,LENGTH=0
          000000000000
          212346626060        ACOS   REAL         LOC=2,LENGTH=0
          000000000002
          255121622560        ERASE  REAL         LOC=131,LENGTH=20
          000024000131
          332667254433        .FXEM. VIRTUAL      SECT. 5,CALL
          200000100000
          625051636060        SQRT   VIRTUAL      SECT. 6,CALL
          200000100000
          627062434623        SYSLOC VIRTUAL      SECT. 7
          200000000000

                $DKEND FASNCS                                                    FASN0003

NO MESSAGES FOR THIS ASSEMBLY
```

```
FATN1                                                                    07/31/65
ASSEMBLED TEXT.

        *TEXT  FATN1                                                                    FATN0001
                                        *     SINGLE-VALUED ARCTANGENT FUNCTION / ATN1                          ATN10010
                                        * SHARE SUBROUTINE B1 GM ATN1                                           ATN10040
                                        * METHODS MEMO 7090-45                                                  ATN10050
                                        *                                                                       ATN10060
                                        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ATN10070
                                        *                                                                       ATN10080
                                        *     A=ATN1 (Y,X)                                                      ATN10090
                                        *                                                                       ATN10100
                                        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ATN10110
                                                                                                                ATN10120
                                              ENTRY   ATN1                                                      ATN10130
                                                                                                                ATN10140
BINARY CARD (NOT PUNCHED)
        00000  000000000000     10000   LNKG  LDIR                                                              ATN10150
        00001  262163450160     10000                                                                           ATN10160
                                        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ATN10170
        00002  1 00000 0 00010   10001   STRT  SAVEN   2,I                                                      ATN10180
        00003  0774 00 2 00000  10000
        00004  0774 00 4 00000  10000
        00005  0441 00 0 00007  10001
        00006  0020 00 4 00001  10000
        00007  0 00000 0 00000  10000
        00010  0604 00 0 00007  10001
        00011  0634 00 4 04000  10011
        00012  0634 00 4 00004  10001
        00013  0634 00 2 00003  10001
        00014  0634 00 4 00000  10001          SXA     LNKG,4                                                   ATN10190
        00015  0500 60 4 00004  10000          CLA*    4,4                                                      ATN10200
        00016  0131 00 0 00000  10000          XCA                                                              ATN10210
        00017  0500 60 4 00003  10000          CLA*    3,4                                                      ATN10220
        00020  0601 00 0 03023  10011          STO     COMMON                                                   ATN10230
        00021  0020 00 0 00023  10001          TRA     BGN                                                      ATN10240
        00022  0020 00 0 00002  10001   ATN1   TRA     STRT                                                     ATN10250
BINARY CARD (NOT PUNCHED)
        00023  4600 00 0 03022  10011   BGN    STO     COMMON-1          STORE X                                ATN10260
        00024  0100 00 0 02023  10011          TZE     ATN1+19                                                  ATN10270
        00025  0771 00 0 00033  10000          ARS     27                CHAR. IN ACC EQUALS P                  ATN10280
        00026  0760 00 0 00003  10000          SSP                                                              ATN10290
        00027  0601 00 0 03020  10011          STO     COMMON-3          CHAR. OF Y                             ATN10300
        00030  0500 00 0 03022  10011          CLA     COMMON-1                                                 ATN10310
        00031  0100 00 0 02025  10011          TZE     ATN1+21                                                  ATN10320
        00032  0771 00 0 00033  10000          ARS     27                CHAR. OF X IN ACC EQUALS Q             ATN10330
        00033  0760 00 0 00003  10000          SSP                                                              ATN10340
        00034  0402 00 0 03020  10011          SUB     COMMON-3          Q-P                                    ATN10350
        00035  0402 00 0 02147  10011          SUB     ATN1+103          Q-P-1                                  ATN10360
        00036  0601 00 0 03017  10011          STO     COMMON-4                                                 ATN10370
        00037  0760 00 0 00003  10000          SSP                       ABS VALUE OF ABOVE                     ATN10380
        00040  0402 00 0 02164  10011          SUB     ATN1+116          I Q-P-1 I-128                          ATN10390
        00041  4120 00 0 02027  10011          TMI     ATN1+23           XFER IF IN RANGE                       ATN10400
```

```
00042  0500 00 0 03020  10011   CLA   COMMON-3      Q-P                          ATN10410
00043  0402 00 0 02164  10011   SUB   ATN1+116      Q-P-128                      ATN10420
00044  0120 00 0 02025  10011   TPL   ATN1+21       XFER IF Y/X WOULD OFLOW      ATN10430
00045  4754 00 0 00000  10000   PXD   0,0           ZERO ACC, IF Y/X IS ZERO     ATN10440

BINARY CARD (NOT PUNCHED)
00046  0020 00 0 02122  10011   TRA   ATN1+82       SET Z EQUALS ZERO            ATN10450
00047  0500 00 0 02146  10011   CLA   ATN1+102      PI/2, IF Y/X IS INF.,        ATN10460
00050  0020 00 0 02122  10011   TRA   ATN1+82       SET Z EQUALS PI/2            ATN10470
00051  0500 00 0 03022  10011   CLA   COMMON-1      X                            ATN10480
00052  0300 00 0 02145  10011   FAD   ATN1+101      NORMALIZE                    ATN10490
00053  0601 00 0 03021  10011   STO   COMMON-2                                   ATN10500
00054  0500 00 0 03023  10011   CLA   COMMON        Y                            ATN10510
00055  0241 00 0 03021  10011   FDP   COMMON-2      Y/X                          ATN10520
00056  4600 00 0 03021  10011   STQ   COMMON-2                                   ATN10530
00057  0500 00 0 03021  10011   CLA   COMMON-2                                   ATN10540
00060  0760 00 0 00003  10000   SSP                 ABS VAL OF Y/X               ATN10550
00061  0601 00 0 03021  10011   STO   COMMON-2      EQUALS W                     ATN10560
00062  0760 00 0 00000  10000   CLM                                              ATN10570
00063  0601 00 0 03013  10011   STO   COMMON-8      SET SW1 TO ZERO              ATN10580
00064  0500 00 0 03017  10011   CLA   COMMON-4      Q-P-1                        ATN10590
00065  0760 00 0 00003  10000   SSP                 ABS VALUE                    ATN10600
00066  0402 00 0 02166  10011   SUB   ATN1+118      I Q-P-1 I-10                 ATN10610
00067  4120 00 0 02053  10011   TMI   ATN1+43       XFER IF W BETWN 2**-10+2**10 ATN10620
00070  0500 00 0 02147  10011   CLA   ATN1+103      1                            ATN10630

BINARY CARD (NOT PUNCHED)
00071  0601 00 0 03013  10011   STO   COMMON-8      RESET SW1 TO 1               ATN10640
00072  0500 00 0 03017  10011   CLA   COMMON-4      Q-P-1                        ATN10650
00073  0120 00 0 02064  10011   TPL   ATN1+52       XFER IF W GTR THAN 2**10     ATN10660
00074  0020 00 0 02056  10011   TRA   ATN1+46       XFER IF W LS THAN 2**-10     ATN10670
00075  0500 00 0 03021  10011   CLA   COMMON-2      W                            ATN10680
00076  0302 00 0 02150  10011   FSB   ATN1+104      W-1                          ATN10690
00077  0120 00 0 02064  10011   TPL   ATN1+52       XFER IF W GTR THAN 1         ATN10700
00100  0500 00 0 02146  10011   CLA   ATN1+102      PI/2                         ATN10710
00101  0601 00 0 03014  10011   STO   COMMON-7      SET SW2 TO PI/2              ATN10720
00102  0500 00 0 02150  10011   CLA   ATN1+104      1.   IF W LS THAN 1, USE 1/W, ATN10730
00103  0241 00 0 03021  10011   FDP   COMMON-2      1/W  + SUBT VALUE FROM PI/2  ATN10740
00104  4600 00 0 03021  10011   STQ   COMMON-2                                   ATN10750
00105  0020 00 0 02066  10011   TRA   ATN1+54       XFER TO POLYNOM              ATN10760
00106  4754 00 0 00000  10000   PXD   0,0           ZERO ACC                     ATN10770
00107  0601 00 0 03014  10011   STO   COMMON-7      SET SW2 TO ZERO              ATN10780
00110  0500 00 0 03013  10011   CLA   COMMON-8      SW1                          ATN10790
00111  4100 00 0 02117  10011   TNZ   ATN1+79       IF NOT ZERO, BYPASS POLYNOM. ATN10800
00112  0500 00 0 03021  10011   CLA   COMMON-2      W                            ATN10810
00113  0300 00 0 02150  10011   FAD   ATN1+104      W+1                          ATN10820

BINARY CARD (NOT PUNCHED)
00114  0601 00 0 03017  10011   STO   COMMON-4                                   ATN10830
00115  0302 00 0 02151  10011   FSB   ATN1+105      W-1                          ATN10840
00116  0100 00 0 02115  10011   TZE   ATN1+77       IF ZERO, Z EQUALS PI/4       ATN10850
00117  0241 00 0 03017  10011   FDP   COMMON-4      W-1/W+1 EQUALS R             ATN10860
00120  4600 00 0 03020  10011   STQ   COMMON-3                                   ATN10870
00121  0260 00 0 03020  10011   FMP   COMMON-3      R*R                          ATN10880
```

```
00122  0601 00 0 03017  10011        STO     COMMON-4                                          ATN10890
00123  0761 00 0 00000  10000        NOP                                                       ATN10900
00124  0534 00 2 02152  10011        LXA     ATN1+106,2              SET IRB TO 7              ATN10910
00125  0500 00 0 02153  10011        CLA     ATN1+107                C15                       ATN10920
00126  0601 00 0 03015  10011        STO     COMMON-6                POLYNOM ERASEABLE E       ATN10930
00127  0560 00 0 03015  10011        LDQ     COMMON-6                                          ATN10940
00130  0260 00 0 03017  10011        FMP     COMMON-4                E*R*R                     ATN10950
00131  0300 00 2 02163  10011        FAD     ATN1+115,2              PLUS CI                   ATN10960
00132  2 00001 2 02104  10011        TIX     ATN1+68,2,1             LOOP                      ATN10970
00133  0761 00 0 00000  10000        NOP                                                       ATN10980
00134  0601 00 0 03016  10011        STO     COMMON-5                VALUE OF POLYNOM          ATN10990
00135  0560 00 0 03016  10011        LDQ     COMMON-5                EQUALS T                  ATN11000
00136  0260 00 0 03020  10011        FMP     COMMON-3                T*R                       ATN11010

BINARY CARD (NOT PUNCHED)
00137  0300 00 0 02163  10011        FAD     ATN1+115                T*R+PI/4                  ATN11020
00140  0601 00 0 03021  10011        STO     COMMON-2                EQUALS Z                  ATN11030
00141  0500 00 0 03014  10011        CLA     COMMON-7                TEST SW2   EQU. 0 OR PI/2 ATN11040
00142  0100 00 0 02123  10011        TZE     ATN1+83                 XFER IF ZERO              ATN11050
00143  0302 00 0 03021  10011        FSB     COMMON-2                PI/2-Z                    ATN11060
00144  0601 00 0 03021  10011        STO     COMMON-2                PORPER 1ST QUAD Z NOW IN CMMN+2 ATN11070
00145  0560 00 0 03023  10011        LDQ     COMMON                  Y   INST 83-100 ADJUST Z TO ATN11080
00146  0260 00 0 03022  10011        FMP     COMMON-1                Y*X            PROPER QUAD. ATN11090
00147  0120 00 0 02136  10011        TPL     ATN1+94                 XFER IF SAME SIGN         ATN11100
00150  0500 00 0 03022  10011        CLA     COMMON-1                X                         ATN11110
00151  0120 00 0 02132  10011        TPL     ATN1+90                 XFER IF X+,Y-4TH QUAD.    ATN11120
00152  0500 00 0 02165  10011        CLA     ATN1+117                PI      X-,Y+2ND QUAD.    ATN11130
00153  0020 00 0 02134  10011        TRA     ATN1+92                                           ATN11140
00154  0500 00 0 02165  10011        CLA     ATN1+117                PI     4TH QUAD           ATN11150
00155  0300 00 0 02165  10011        FAD     ATN1+117                2*PI                      ATN11160
00156  0302 00 0 03021  10011        FSB     COMMON-2                -Z                        ATN11170
                  00157              RETURN  STRT                                              ATN11180
00160  0500 00 0 03022  10011        CLA     COMMON-1                X    SAME SIGN            ATN11190
00161  0120 00 0 02143  10011        TPL     ATN1+99                 XFER IF X+,Y+,1ST QUAD    ATN11200

BINARY CARD (NOT PUNCHED)
00162  0500 00 0 02165  10011        CLA     ATN1+117                PI      X-,Y-,3RD QUAD    ATN11210
00163  0300 00 0 03021  10011        FAD     COMMON-2                +Z                        ATN11220
                  00164              RETURN  STRT                                              ATN11230
00165  0500 00 0 03021  10011        CLA     COMMON-2                Z                         ATN11240
                  00166              RETURN  STRT                                              ATN11250
                                *                                                               ATN11260
                                * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ATN11270
                                *                                                               ATN11280
00167  0000 00 0 00000  10000        HTR     0                       FIRST PROGRAM CONSTANT    ATN11290
00170  201622077325     10000        DEC     1.57079633,1,1.,2.,7,-.004054058,.0218612288      ATN11300
00171  000000000001     10000
00172  201400000000     10000
00173  202400000000     10000
00174  000000000007     10000
00175  571411537472     10000
00176  173546131217     10000
00177  574712007036     10000        DEC     -.0559098861,.096420044,-.139085335,.199465360    ATN11310
00200  175612737372     10000
```

ASSEMBLED TEXT.

```
00201  576434661427     10000
00202  176630402456     10000
00203  577525230337     10000          DEC     -.333298561,.999999333,.78539816,128,3.14159265,10      ATN11320
00204  200777777646     10000

BINARY CARD (NOT PUNCHED)
00205  200622077325     10000
00206  000000000200     10000
00207  202622077325     10000
00210  000000000012     10000
                                     *                                                                   ATN11330
                                     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ATN11340
                                     *                                                                   ATN11350
                                     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ATN11360
                                     *                                                                   ATN11370
                                     ERASE  CONTRL  POOL                                                 ATN11380
00211  100000000211     00001        USE    POOL                                                         ATN11390
00211  200000000023     00001  TEMP  BSS    19                                                           ATN11400
00234  200000000001     00001  COMMON BSS   1                                                            ATN11410
                                     *                                                                   ATN11420
                                     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ATN11430
                                                                                                         ATN11440
                        00000  01111        END                                                          ATN11450
```

```
FATN1                                                                    07/31/65
CONTROL DICTIONARY

        $CDICT FATN1                                                                FATN0002

BINARY CARD (NOT PUNCHED)
        000235000000                PREFACE     START=0,LENGTH=157,TYPE=7094,CMPLX=5
        000004000005
        262163450160        FATN1   DECK        LOC=0,LENGTH=157
        000235000000
        216345016060        ATN1    REAL        LOC=22,LENGTH=0
        000000000022
        255121622560        ERASE   REAL        LOC=211,LENGTH=20
        000024000211
        627062434623        SYSLOC  VIRTUAL     SECT. 4
        200000000000

        $DKEND FATN1                                                                FATN0003

NO MESSAGES FOR THIS ASSEMBLY
```

```
                FDATE                                                                07/31/65
              ASSEMBLED TEXT.

                    $TEXT  FDATE                                                                FDAT0001

                                        *     DATE AS MON YY, 19ZZ / DATE                                          DATE0010
                                        *DATE    709/90 FORTRAN LIBRARY / DATE AS MON YY, 19ZZ                     DATE0020
                                        * H. EUGENE WRIGHT  METHODS - TRANSPORT                                    DATE0030
                                        * CONVERTED TO IBMAP AUG 1963 BY D TYLER                                   DATE0040
                                        *                                                                          DATE0050
                                        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * DATE0060
                                        *                                                                          DATE0070
                                        *     CALL DATE (A,B)                                                      DATE0080
                                        *                                                                          DATE0090
                                        * WHERE A AND B ARE THE LOCATIONS WHERE THE DATE WILL BE STORED.           DATE0100
                                        *       THE DATE FORMAT IS                                                 DATE0110
                                        *              LOCATION A    MON YY                                        DATE0120
                                        *              LOCATION B    , 19ZZ                                        DATE0130
                                        *       WHERE MON REPRESENTS A  THREE CHARACTER ABBREVIATION FOR THE       DATE0140
                                        *       MONTH, AND YY AND ZZ REPRESENT THE DAY AND YEAR, RESPECTIVELY.     DATE0150
                                        *                                                                          DATE0160
                                        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * DATE0170
                                                                                                                   DATE0180
                                                ENTRY   DATE                                                       DATE0190
                                                                                                                   DATE0200
                          00101         DATEBX EQU     65            IBSYS DATE                                    DATE0210
                                                                                                                   DATE0220
                                        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * DATE0230

BINARY CARD (NOT PUNCHED)
        00000  000000000000      10000          LDIR                                                                DATE0240
        00001  262421632560      10000
        00002  1 00000 0 00005   10001   DATE   SAVEN   4                                                           DATE0250
        00003  0774 00 4 00000   10000
        00004  0020 00 4 00001   10000
        00005  0634 00 4 03000   10011
        00006  0634 00 4 00003   10001
        00007  0560 00 0 00101   10000          LDQ     DATEBX                                                      DATE0260
        00010  4754 00 0 00000   10000          PXD     0,0                                                         DATE0270
        00011  0601 00 0 00033   10001          STO     MONTH         MAKE ZERO                                     DATE0280
        00012  4763 00 0 00006   10000          LGL     6                                                           DATE0290
        00013  0601 00 0 00033   10001          STO     MONTH                                                       DATE0300
        00014  4754 00 0 00000   10000          PXD     0,0                                                         DATE0310
        00015  4763 00 0 00006   10000          LGL     6                                                           DATE0320
        00016  0737 00 4 00000   10000          PAC     0,4                                                         DATE0330
        00017  0520 00 0 00033   10001          ZET     MONTH         TEST IF OCT-DEC                               DATE0340
        00020  1 77766 4 01001   10011          TXI     *+1,4,-10                                                   DATE0350
        00021  4500 00 4 00033   10001          CAL     MONTH,4       PICK UP MONTH                                 DATE0360
        00022  4763 00 0 00014   10000          LGL     12            MOVE IN DAY                                   DATE0370

BINARY CARD (NOT PUNCHED)
        00023  0534 00 4 00003   10001   IR4    LXA     ..0001,4                                                    DATE0380
        00024  0602 60 4 00003   10000          SLW*    3,4                                                         DATE0390
        00025  4500 00 0 00050   10001          CAL     =H00, 19                                                    DATE0400
        00026  4763 00 0 00014   10000          LGL     12            MOVE IN YEAR                                  DATE0410
        00027  0602 60 4 00004   10000          SLW*    4,4                                                         DATE0420
        00030  4500 00 0 00101   10000          CAL     DATEBX                                                      DATE0430
```

```
00031  0140 00 0 02001  10011        TOV     DATE+1        TURN OFF OVERFLOW LIGHT          DATE0440
00032  0020 00 0 02001  10011        TRA     DATE+1                                         DATE0450
                                  *                                                         DATE0460
                                  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * DATE0470
                                  *                                                         DATE0480
00033  0 00000 0 00000  10000  MONTH PZE     0                                              DATE0490
00034  000041214560     10000        BCI     4,00JAN 00FEB 00MAR 00APR                      DATE0500
00035  000026252260     10000
00036  000044215160     10000
00037  000021475160     10000
00040  000044217060     10000        BCI     4,00MAY 00JUN 00JUL 00AUG                      DATE0510
00041  000041644560     10000
00042  000041644360     10000
00043  000021642760     10000
00044  000062254760     10000        BCI     4,00SEP 00OCT 00NOV 00DEC                      DATE0520
00045  000046236360     10000

BINARY CARD (NOT PUNCHED)
00046  000045466560     10000
00047  000024252360     10000
                                  *                                                         DATE0530
                                  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * DATE0540
00050  000073600111     10000        *LORG                                                  DATE0550
                 00000  01111        END                                                    DATE0560
```

```
FDATE                                                                    07/31/65
CONTROL DICTIONARY

        $CDICT FDATE                                                                FDAT0002

BINARY CARD (NOT PUNCHED)
        000051000000              PREFACE     START=0,LENGTH=41,TYPE=7094,CMPLX=5
        000004000005
        262421632560      FDATE   DECK        LOC=0,LENGTH=41
        000051000000
        242163256060      DATE    REAL        LOC=2,LENGTH=0
        000000000002
        627062434623      SYSLOC  VIRTUAL     SECT. 3
        200000000000

        $DKEND FDATE                                                                FDAT0003

NO MESSAGES FOR THIS ASSEMBLY
```

```
$TEXT  FFSF                                                                    FFSF0001
                                  *FSF     709/90 FORTRAN LIBRARY / FORWARD SPACE FILE                    FFSF0020
                                  * DOT TYLER    10/63                                                   FFSF0030
                                  * METHODS MEMO 7090-25                                                 FFSF0040
                                  *                                                                      FFSF0050
                                  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FFSF0060
                                  *                                                                      FFSF0070
                                  *     CALL FSF (NFILE,LTAPE,LERROR)                                    FFSF0080
                                  *                                                                      FFSF0090
                                  *          NFILE  - NUMBER OF END-OF-FILE MARKS TO SPACE PASS.         FFSF0100
                                  *                   ZERO IS EQUIVALENT TO ONE.                         FFSF0110
                                  *          LTAPE  - LOGICAL TAPE NUMBER.                               FFSF0120
                                  *          LERROR - LOCATION IN WHICH THE ERROR CODE IS STORED.        FFSF0130
                                  *                   0  INDICATES SUCCESS.                              FFSF0140
                                  *                   N  INDICATES NUMBER OF UNSPACED FILES WHEN         FFSF0150
                                  *                      END-OF-TAPE SIGNAL WAS REACHED (SEE MM-23).     FFSF0160
                                  *                                                                      FFSF0170
                                  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FFSF0180
                                                                                                         FFSF0190
BINARY CARD (NOT PUNCHED)
      00000  000000000000    10000        LDIR                                                           FFSF0200
      00001  262662266060    10000
                                                                                                         FFSF0210
                                                                                                         FFSF0220
                                  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FFSF0230
      00002  1 00000 0 00011  10001  FSF   SAVE    (4,2,1)I                                              FFSF0240
      00003  0774 00 2 00000  10000
      00004  0774 00 1 00000  10000
      00005  0774 00 4 00000  10000
      00006  0441 00 0 00010  10001
      00007  0020 00 4 00001  10000
      00010  0 00000 0 00000  10000
      00011  0604 00 0 00010  10001
      00012  0634 00 4 05000  10011
      00013  0634 00 4 00000  10001
      00014  0634 00 4 00005  10001
      00015  0634 00 1 00004  10001
      00016  0634 00 2 00003  10001
      00017  0500 60 4 00004  10000        CLA*    4,4                                                    FFSF0250
      00020  0601 00 0 03000  10011        STO     TEMP          LOGICAL UNIT                             FFSF0260
      00021  0500 60 4 00003  10000        CLA*    3,4                                                    FFSF0270
      00022  0734 00 1 00000  10000        PAX     0,1           CTR TO XR1                               FFSF0280
BINARY CARD (NOT PUNCHED)
      00023  030000300000     00010        CALL    .FVIO.(TEMP,TEMP+1)                                    FFSF0290
      00023  0674 00 4 04000  10011
      00024  1 00002 0 01004  10011
      00025  0 00000 0 00034  10100
      00026  0 00000 0 03000  10011
      00027  0 00000 0 03001  10011
      00030  0535 00 2 03001  10011        LAC     TEMP+1,2                                               FFSF0300
      00031  0441 00 2 00001  10000        LDI     1,2                                                    FFSF0310
```

ASSEMBLED TEXT.

```
00032  0500 00 0 03001  10011          CLA     TEMP+1                                  FFSF0320
00033  0621 00 0 00042  10001          STA     SETA                                    FFSF0330
00034  0621 00 0 00040  10001          STA     SETX                                    FFSF0340
00035  4054 00  040000  10000          LFT     040000                                  FFSF0350
00036  0020 00 0 00041  10001          TRA     LOOP                                    FFSF0360
00037  0074 00 4 06000  10011          TSX     .OPEN,4                                 FFSF0370
00040  5 00000 0 00000  10000  SETX    MON     **                                      FFSF0380
00041  0074 00 4 07000  10011  LOOP    TSX     .READ,4        READ TO EOF              FFSF0390
00042  0 00000 0 00000  10000  SETA    PZE     **,,0                                   FFSF0400
00043  0 00046 0 00053  10101          PZE     EOF,,ERR                                FFSF0410
00044  4 00002 0 03002  10011          IOCP    SKIM,0,2                                FFSF0420

BINARY CARD (NOT PUNCHED)
00045  3 77777 2 00000  10000          IORTN   **,0,-1                                 FFSF0430
00046  4500 00 0 03003  10011  ERR     CAL     SKIM+1                                  FFSF0440
00047  4340 00 0 00063  10001          LAS     CODE           IF TAPEND                FFSF0450
00050  0020 00 0 00041  10001          TRA     LOOP                                    FFSF0460
00051  0020 00 0 00057  10001          TRA     ERRA                                    FFSF0470
00052  0020 00 0 00041  10001          TRA     LOOP                                    FFSF0480
00053  2 00001 1 00041  10001  EOF     TIX     LOOP,1,1                                FFSF0490
00054  0534 00 4 00005  10001          LXA     ..0001,4                                FFSF0500
00055  0600 60 4 00005  10000          STZ*    5,4            SET N=0                  FFSF0510
                       00056          RETURN  FSF                                     FFSF0520
00057  0754 00 1 00000  10000  ERRA    PXA     0,1            SET N TO CTR OF          FFSF0530
00060  0534 00 4 00005  10001          LXA     ..0001,4         UNSKIPPED FILES        FFSF0540
00061  0601 60 4 00005  10000          STO*    5,4                                     FFSF0550
00062  0020 00 0 00056  10001          TRA     ERRA-1                                  FFSF0560
00063  632147254524     10000  CODE    BCI     1,TAPEND                                FFSF0570
                       00064          LORG                                            FFSF0580
                                ERASE  CONTRL  ER                                      FFSF0590
00064  100000000064     00001          USE     ER                                      FFSF0600
00064  200000000002     00001  TEMP    BSS     2                                       FFSF0610
00066  200000000002     00001  SKIM    BSS     2                                       FFSF0620
00070  200000000020     00001          BSS     16                                      FFSF0630

BINARY CARD (NOT PUNCHED)
                       00000  01111          END                                     FFSF0640
```

```
$CDICT FFSF                                                          FFSF0002

BINARY CARD (NOT PUNCHED)
          000110000000          PREFACE   START=0,LENGTH=72,TYPE=7094,CMPLX=5
          000004000005
          262662266060   FFSF   DECK      LOC=0,LENGTH=72
          000110000000
          265226606060   FSF    REAL      LOC=2,LENGTH=0
          000000000002
          255121622560   ERASE  REAL      LOC=64,LENGTH=20
          000024000064
          332665314633   .FVIO. VIRTUAL   SECT. 4,CALL
          200000100000
          627062434623   SYSLOC VIRTUAL   SECT. 5
          200000000000
          334647254560   .OPEN  VIRTUAL   SECT. 6
          200000000000
          335125212460   .READ  VIRTUAL   SECT. 7
          200000000000

$DKEND FFSF                                                          FFSF0003

NO MESSAGES FOR THIS ASSEMBLY
```

FESF 07/31/65

SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

CLASS	SYMBOL	VALUE	REFERENCES
	CODE	00063	47
	EOF	00053	43
	ERRA	00057	51,62
	ERR	00046	43
LCTR	ER		
	FSF	00002	56
	..0001	00005	14,15,16,54,60
	..0002	00007	6,11
	..0003	00011	2
	LOOP	00041	36,50,52,53
LCTR	BLCTR		
QUAL	UNQS		
LCTR	//		
	SETA	00042	33
	SETX	00040	34
	SKIM	00066	44,46
	TEMP	00064	20,26,27,30,32

REFERENCES TO VIRTUAL SYMBOLS.

.EVID.	4	23
.OPEN	6	37
.READ	7	41
SYSLOC	5	

```
$TEXT  FFSR                                                                  FFSR0001

                                    *     FORWARD SPACE RECORD / FSR                                          FSR00020
                                    * JIM STOKER   TRANSPORT - 4/61                                           FSR00030
                                    * CONVERTED TO IBMAP BY DOT TYLER - 10/63                                 FSR00040
                                    * METHODS MEMO 7090-24                                                    FSR00050
                                    *                                                                         FSR00060
                                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FSR00070
                                    *                                                                         FSR00080
                                    *     CALL FSR (NREC,LTAPE,LERROR)                                        FSR00090
                                    *                                                                         FSR00100
                                    *          NREC - NUMBER OF FORTRAN LOGICAL RECORDS TO SPACE FORWARD.     FSR00110
                                    *         LTAPE - LOGICAL TAPE NUMBER.                                    FSR00120
                                    *        LERROR - LOCATION IN WHICH THE ERROR CODE IS STORED.             FSR00130
                                    *                 0  INDICATES SUCCESS.                                   FSR00140
                                    *                 N  INDICATES NUMBER OF UNSPACED RECORDS WHEN AN         FSR00150
                                    *                    END-OF-FILE MARK WAS FOUND.  TAPE LEFT SPACED IN     FSR00160
                                    *                    FRONT OF EOF.                                        FSR00170
                                    *                                                                         FSR00180
                                    *     CALL     FSPR(NPREC,LTAPE,LERROR)                                   FSR00190
                                    *                                                                         FSR00200
                                    *         NPREC - NUMBER PHYSICAL RECORDS TO SPACE FORWARD.               FSR00210
                                    *                                                                         FSR00220
                                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FSR00230
                                                                                                              FSR00240
                                          ENTRY   FSR                                                         FSR00250
                                          ENTRY   FSPR                                                     M3 FSR00260
                                                                                                              FSR00270
                                                                                                              FSR00280
                                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FSR00290

BINARY CARD (NOT PUNCHED)
      00000  000000000000      10000          LDIR                                                            FSR00300
      00001  262662516060      10000
      00002  0600 00 0 04004   10011    FSPR  STZ     CODE                                                    FSR00310
      00003  0020 00 0 01002   10011          TRA     *+2                                                     FSR00320
      00004  4625 00 0 04004   10011    FSR   STL     CODE                                                    FSR00330
      00005  1 00000 0 00014   10001    TURN  SAVEN   4,2,1,1                                                 FSR00340
      00006  0774 00 2 00000   10000
      00007  0774 00 1 00000   10000
      00010  0774 00 4 00000   10000
      00011  0441 00 0 00013   10001
      00012  0020 00 4 00001   10000
      00013  0 00000 0 00000   10000
      00014  0604 00 0 00013   10001
      00015  0634 00 4 06000   10011
      00016  0634 00 4 00010   10001
      00017  0634 00 1 00007   10001
      00020  0634 00 2 00006   10001
      00021  0500 60 4 00004   10000          CLA*    4,4                                                     FSR00350
      00022  0601 00 0 04000   10011          STO     TEMP                                                    FSR00360

BINARY CARD (NOT PUNCHED)
      00023  0500 60 4 00003   10000          CLA*    3,4                                                     FSR00370
```

```
00024  0734 00 1 00000  10000          PAX     0,1                                  FSR00380
00025  0100 00 0 00063  10001          TZE     WIND                                 FSR00390
00026  000000000000     00010          CALL    .FVIO.(TEMP,TEMP+1)                  FSR00400
00026  0074 00 4 05000  10011
00027  1 00002 0 01004  10011
00030  0 00000 0 00047  10100
00031  0 00000 0 04000  10011
00032  0 00000 0 04001  10011
00033  0535 00 2 04001  10011          LAC     TEMP+1,2                             FSR00410
00034  0636 00 2 00044  10001          SCA     SETA,2                               FSR00420
00035  0636 00 2 00042  10001          SCA     SETB,2                               FSR00430
00036  0441 00 2 00001  10000          LDI     1,2                                  FSR00440
00037  4054 00  040000  10000          LFT     040000                               FSR00450
00040  0020 00 0 00043  10001          TRA     READL                                FSR00460
00041  0074 00 4 07000  10011          TSX     .OPEN,4                              FSR00470
00042  5 00000 0 00000  10000  SETB    MON     **                                   FSR00480
00043  0074 00 4 10000  10011  READL   TSX     .READ,4                              FSR00490
00044  0 00000 0 00000  10000  SETA    PZE     **,,0                                FSR00500

BINARY CARD (NOT PUNCHED)
00045  0 00050 0 00060  10101          PZE     EOF,,ERR                             FSR00510
00046  4 00002 0 04002  10011          IOCP    SKIM,0,2                             FSR00520
00047  3 77777 2 00000  10000          IORTN   **,0,-1                              FSR00530
00050  0534 00 4 04002  10011  ERR     LXA     SKIM,4                               FSR00540
00051  0520 00 0 04004  10011          ZET     CODE                                 FSR00550
00052  7 00000 4 00043  10001          TXL     READL,4,0                            FSR00560
00053  2 00001 1 00043  10001          TIX     READL,1,1                            FSR00570
00054  4754 00 0 00000  10000          PXD     0,0                                  FSR00580
00055  0534 00 4 00010  10001  DONE    LXA     ..0001,4                             FSR00590
00056  0601 60 4 00005  10000          STD*    5,4                                  FSR00600
                       00057          RETURN  TURN                                 FSR00610
00060  0074 00 4 11000  10011  EOF     TSX     .NDSEL,4                             FSR00620
00061  0 00005 2 00000  10000          PZE     0,2,5                                FSR00630
00062  0761 00 0 00000  10000          NOP                                          FSR00640
00063  0754 00 1 00000  10000  WIND    PXA     0,1                                  FSR00650
00064  0020 00 0 00055  10001          TRA     DONE                                 FSR00660
                                     *                                              FSR00670
                                     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FSR00680
                                     *                                              FSR00690
                                      ERASE  CONTRL  ER                             FSR00700
00065  100000000065     00001          USE     ER                                   FSR00710
00065  200000000002     00001  TEMP    BSS     2                                    FSR00720
00067  200000000002     00001  SKIM    BSS     2                                    FSR00730

BINARY CARD (NOT PUNCHED)
00071  200000000001     00001  CODE    BSS     1                                    FSR00740
00072  200000000017     00001          BSS     15                                   FSR00750
                                     *                                              FSR00760
                                     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FSR00770
                                                                                    FSR00780
                       00000  01111          END                                     FSR00790
```

```
$CDICT FFSR                                                          FFSR0002

BINARY CARD (NOT PUNCHED)
          000111000000          PREFACE   START=0,LENGTH=73,TYPE=7094,CMPLX=5
          000004000005
          262662516060   FFSR   DECK      LOC=0,LENGTH=73
          000111000000
          266247516060   FSPR   REAL      LOC=2,LENGTH=0
          000000000002
          266251606060   FSR    REAL      LOC=4,LENGTH=0
          000000000004
          255121622560   ERASE  REAL      LOC=65,LENGTH=20
          000024000065
          332665314633   .FVIO. VIRTUAL   SECT. 5,CALL
          200000100000
          627062434623   SYSLOC VIRTUAL   SECT. 6
          200000000000
          334647254560   .OPEN  VIRTUAL   SECT. 7
          200000000000
          335125212460   .READ  VIRTUAL   SECT. 8
          200000000000
          334524622543   .NDSEL VIRTUAL   SECT. 9
          200000000000

$DKEND FFSR                                                          FFSR0003

NO MESSAGES FOR THIS ASSEMBLY
```

```
      SUBROUTINE FORCES

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE
      DIMENSION AREA(200),ALPHA(100),ALPHAB(100),ALPHAL(100),ALPHAS(200)
     1,ALPHAU(100),ALPHAW(100),CL(200),CLBB(100),CLWW(100),CPL(100)
     2,CPU(100),CPB(100),THETA(200),TITLE(12),U(100),          UWBT(100)
     3,UBWT(100),UWWT(100),V(100),          VWBT(100),VBWT(100),VWWT(100
     4),W(100),          WWBT(100),WBWT(100),WWWT(100),XBAR(200),YBAR(20
     50),ZBAR(200),ALPHAT(100),A(200),B(200),C(200)
      DIMENSION UBB(100),VBB(100),WBB(100),UWW(100),VWW(100)
     1,WWW(100),ALPHAX(100),UWB(100),VWB(100),WWB(100)
     2,UBW(100),VBW(100),WBW(100)
      DIMENSION ABX(100),AWX(100)
      DIMENSION WSCLL(20),WSCDL(20),WSCLU(20),WSCDU(20),WSCL(20),WSCD(20
     1)
      DIMENSION CHORD(200),NROW(2)
      DIMENSION NFMT(8)
      DIMENSION XC(200),YC(200),ZC(200)
      DIMENSION XB(51),R(51),THETAB(10),ZDELTA(51),CPBB(51,10)


      KONFIG=KACE                                                       ,1
      NPANEL=NWING+NBODY                                                ,2
      NS=NBODY+1                                                        ,3
      BETA=SQRT(XMACH*XMACH-1.)                                         ,4
      BETA2=BETA*BETA                                                   ,5
      RATIOD=.75                                                        ,6
      CDR=57.2957795                                                    ,7
      IF (KONFIG-2) 5,10,5                                              ,8
 5    CONTINUE                                                          ,9
      READ  (NTAPEC) (I,XBAR(I),YBAR(I),ZBAR(I),XC(I),YC(I),ZC(I),AREA(I
     1),THETA(I),ALPHAS(I),CHORD(I),I=1,NPANEL),NRG,(NROW(I),I=1,NRG),RA
     2TIOX                                                              ,10   ,11   ,12   ,13   ,14   ,15
                                                                        ,16   ,17   ,18   ,19   ,20
      REWIND NTAPEC
                                                                        ,21
 10   CONTINUE
                                                                        ,22
      READ (NTAPEI,5000) TITLE


C     CASE = 1.      CALCULATE ALPHA, GIVEN CL
C     CASE = 2.      CALCULATE CL, GIVEN ALPHA
C     CASE = 3.      OPTIMIZE WING

C     CPCALC =0.     LINEAR CP
C     CPCALC =1.     NON-LINEAR CP

C     POLAR = 0.     POLARS NOT REQUESTED,OMIT DELTA ALPHA CARDS
C     POLARS = 1.    POLARS REQUESTED, INSERT DELTA ALPHA CARDS

C     THICK =0.      WING THICKNESS PRESSURES NOT TO BE ADDED
```

```
C     THICK =1.       WING THICKNESS PRESSURES TO BE ADDED

C     VOUT = 0.       VELOCITY COMPONENTS NOT TO BE PRINTED
C     VOUT = 1.       VELOCITY COMPONENTS TO BE PRINTED

C     RFAREA =0.      WING REFERENCE AREA TO BE CALCULATED
C     RFAREA =X.      WING REFERENCE AREA TO BE SPECIFIED

C     XP,ZP           X AND Z COORDINATES OF POINT ABOUT WHICH MOMENTS
C                     ARE TO BE COMPUTED

                                                                        ,23   ,24   ,25
      READ (NTAPEI,5010) CASE,CPCALC,POLAR,THICK,VOUT                   ,26   ,27   ,28
      IF (CASE) 20,999,20                                               ,29
 20   CONTINUE                                                          ,30
      READ (NTAPEI,5010) RFAREA,XP,ZP
                                                                        ,31   ,32   ,33
      IF (THICK) 50,60,50                                               ,34
 50   CONTINUE                                                          ,35
      CALL FSF (1,NTAPEC,IRR)                                           ,36
      READ (NTAPEC) (ALPHAT(I),I=1,NWING)                               ,37   ,38   ,39   ,40   ,41
      REWIND NTAPEC                                                     ,42
 60   CONTINUE
                                                                        ,43
      IF (VOUT) 75,85,75                                                ,44
 75   CONTINUE                                                          ,45
      DO 80 J=1,7                                                       ,46
 80   NFMT(J)=J                                                         ,47   ,48
      GO TO 100                                                         ,49
   85 CONTINUE                                                          ,50
      DO 90 J=1,7                                                       ,51
   90 NFMT(J)=0                                                         ,52   ,53
  100 CONTINUE                                                          ,54
      KASE=CASE
                                                                        ,55
      IF (RFAREA) 120,120,140                                           ,56
  120 IF (KONFIG-2) 130,125,130                                         ,57
  125 RFAREA=1.0                                                        ,58
      GO TO 140                                                         ,59
  130 RFAREA=0.0                                                        ,60
      DO 135 J=NS,NPANEL                                                ,61
  135 RFAREA=RFAREA+AREA(J)                                             ,62   ,63
  140 CONTINUE
                                                                        ,64
      WRITE (NTAPEO,6002) TITLE                                         ,65   ,66   ,67
      CALL INOUT (NTAPEO,KASE,CPCALC,POLAR,THICK,VOUT,RFAREA,XP,ZP,XMACH
     1,SYM)
                                                                        ,68
      GO TO (300,200,400),KONFIG

C     ******************************************************************

C     BODY ONLY
                                                                        ,69
```

```
  200 CONTINUE                                                          ,70
      NROWB=NROW(1)

C     CALCULATE CL, GIVEN ALPHA
                                                                        ,71
      READ (NTAPEC) ALPHAA                                              ,72   ,73   ,74
      READ (NTAPEC) NBODYS,NXLE,ZA,(XB(I),R(I),I=1,NBODYS)              ,75   ,76   ,77   ,78   ,79   ,80
                                                                        ,81   ,82
      READ (NTAPEC) NTHETA,(THETAB(I),I=1,NTHETA)                       ,83   ,84   ,85   ,86   ,87   ,88

      CALL READ (NTAPEI,NTAPEO,NTAPEC,NBODYS,ZDELTA)                    ,89
      REWIND NTAPEC                                                     ,90
      ARA=-ALPHAA                                                       ,91
      ARADEG=ARA*CDR                                                    ,92
      ALPHAD=ALPHAA*CDR                                                 ,93
      WRITE (NTAPEO,6090) ALPHAD                                        ,94   ,95   ,96
      WRITE (NTAPEO,6092) ARADEG                                        ,97   ,98   ,99
      WRITE (NTAPEO,6094) ZA

C     RETURN POINT FOR POLARS                                           ,100  ,101  ,102
  230 CONTINUE                                                          ,103
      CALL KARMOR (NBODYS,NTHETA,CPCALC,VOUT,ZA,ARA,XP,ZP,XB,R,THETAB
     1,ZDELTA,XC,YC,ZC,THETA,U,V,W,A,CPBB,BCL,BCD,BCM)                  ,104
      BCL=BCL/RFAREA                                                    ,105
      BCD=BCD/RFAREA                                                    ,106
      BCM=BCM/RFAREA                                                    ,107
      GO TO 900

C     END OF BODY CASE

C     ****************************************************************

C     WING ONLY
                                                                        ,108
  300 CONTINUE                                                          ,109
      NROWW=NROW(1)                                                     ,110
      NCOLW=NWING/NROWW                                                 ,111
      ALPHAA=0.                                                         ,112
      ARA=0.                                                            ,113
      ALPHAD=ALPHAA*CDR                                                 ,114
      ARADEG=ARA*CDR                                                    ,115
      WRITE (NTAPEO,6090) ALPHAD                                        ,116  ,117  ,118
      WRITE (NTAPEO,6092) ARADEG                                        ,119  ,120  ,121
      GO TO (305,335,350),KASE

C     CALCULATE ALPHA, GIVEN CL
                                                                        ,122
  305 CONTINUE                                                          ,123
      CALL READ (NTAPEI,NTAPEO,NTAPEC,NWING,CL)                         ,124
      DO 310 J=1,NWING                                                  ,125
  310 CL(J)=-CL(J)                                                      ,126  ,127
      CALL CAMBW (NWING,NTAPEE,A,CL,ALPHAW)                             ,128
      REWIND NTAPEE                                                     ,129
      GO TO 360
```

```
C     CALCULATE CL, GIVEN ALPHA
                                                                          ,130
  335 CONTINUE                                                            ,131
      CALL FSF (2,NTAPEC,IRR)                                             ,132
      CALL READ (NTAPEI,NTAPEO,NTAPEC,NWING,ALPHAW)                       ,133
      REWIND NTAPEC                                                       ,134
  340 CONTINUE                                                            ,135
      CALL FSF (1,NTAPEE,IRR)                                             ,136
      CALL DCPD (NWING,NTAPEE,A,ALPHAW,CL)                                ,137
      REWIND NTAPEE                                                       ,138
      GO TO 360

C     OPTIMIZE CAMBER, GIVEN CL OR CL AND CM
                                                                          ,139
  350 CONTINUE                                                            ,140
      READ (NTAPEI,5010) CONSNT,CLBAR,XCPBAR                              ,141  ,142  ,143
      IF (CONSNT) 354,352,354                                             ,144
  352 CALL FSF (1,NTAPEA,IRR)                                             ,145
      WRITE (NTAPEO,6082) CLBAR                                           ,146  ,147  ,148
      GO TO 356                                                           ,149
  354 CALL FSF (2,NTAPEA,IRR)                                             ,150
      WRITE (NTAPEO,6082) CLBAR                                           ,151  ,152  ,153
      WRITE (NTAPEO,6084) XCPBAR                                          ,154  ,155  ,156
  356 CONTINUE                                                            ,157
      CALL OPTMW (NWING,NTAPEA,A,B,CONSNT,CLBAR,XCPBAR,RFAREA,AREA,CL)    ,158
      REWIND NTAPEA                                                       ,159
      CALL CAMBW (NWING,NTAPEE,A,CL,ALPHAW)                               ,160
      REWIND NTAPEE

                                                                          ,161
  360 CONTINUE                                                            ,162
      CALL CVEL (NWING,NWING,NTAPEB,A,B,C,CL,UWW,VWW,WWW)                 ,163
      REWIND NTAPEB                                                       ,164
      CALL RITE (NFMT(4),NTAPEO,NWING,NROWW,NCOLW,DUMMY,UWW,VWW,WWW)

C     COMPUTE COEFFICIENT OF PRESSURE ON WING WITH OR WITHOUT WING
C     THICKNESS                                                           ,165
      IF (THICK) 365,380,365                                              ,166
  365 CONTINUE                                                            ,167
      CALL FSF (3,NTAPEC,IRR)                                             ,168
      READ (NTAPEC) (UWWT(I),VWWT(I),WWWT(I),I=1,NWING)                   ,169  ,170  ,171  ,172  ,173
      REWIND NTAPEC                                                       ,174
      CALL RITE (NFMT(5),NTAPEO,NWING,NROWW,NCOLW,DUMMY,UWWT,VWWT,WWWT)   ,175
      DO 370 J=1,NWING                                                    ,176
      U(J)=UWW(J)+UWWT(J)                                                 ,177
      V(J)=VWW(J)+VWWT(J)                                                 ,178
  370 W(J)=WWW(J)+WWWT(J)                                                 ,179  ,180
      CALL CP (NWING,BETA2,CPCALC,U,V,W,CPU)                              ,181
      DO 375 J=1,NWING                                                    ,182
      U(J)=UWWT(J)-UWW(J)                                                 ,183
  375 W(J)=WWW(J)-WWWT(J)                                                 ,184  ,185
      CALL CP (NWING,BETA2,CPCALC,U,V,W,CPL)                              ,186
```

```
      DO 378 J=1,NWING                                                  ,187
      ALPHAU(J)=ALPHAW(J)+ALPHAT(J)                                     ,188
  378 ALPHAL(J)=ALPHAW(J)-ALPHAT(J)                                     ,189  ,190
      GO TO 390
                                                                        ,191
  380 CONTINUE                                                          ,192
      DO 382 J=1,NWING                                                  ,193
      U(J)=UWW(J)                                                       ,194
      V(J)=VWW(J)                                                       ,195
 382  W(J)=WWW(J)                                                       ,196  ,197
      CALL CP (NWING,BETA2,CPCALC,U,V,W,CPU)                            ,198
      DO 384 J=1,NWING                                                  ,199
 384  U(J)=-UWW(J)                                                      ,200  ,201
      CALL CP (NWING,BETA2,CPCALC,U,V,W,CPL)                            ,202
      DO 385 J=1,NWING                                                  ,203
      ALPHAU(J)=ALPHAW(J)                                               ,204
  385 ALPHAL(J)=ALPHAW(J)

C     COMPUTE COEFFICIENTS OF LIFT, DRAG, MOMENT ON WING                ,205  ,206
  390 CONTINUE                                                          ,207
      CALL INTPOL (NWING,NROWW,RATIOX,RATIOD,CHORD,ALPHAU,ALPHA)        ,208
      CALL WLDM (NWING,NROWW,XP,ZP,RFAREA,AREA,XBAR,ZBAR,ALPHA,THETA,
     1CPU,WSCLU,WSCDU,WCLU,WCDU,WCMU)                                   ,209
      CALL INTPOL (NWING,NROWW,RATIOX,RATIOD,CHORD,ALPHAL,ALPHA)        ,210
      CALL WLDM (NWING,NROWW,XP,ZP,RFAREA,AREA,XBAR,ZBAR,ALPHA,THETA,
     1CPL,WSCLL,WSCDL,WCLL,WCDL,WCML)                                   ,211
      DO 398 J=1,NCOLW                                                  ,212
      WSCL(J)=WSCLU(J)-WSCLL(J)                                         ,213
  398 WSCD(J)=WSCDU(J)-WSCDL(J)                                         ,214  ,215
      WCL=WCLU-WCLL                                                     ,216
      WCD=WCDU-WCDL                                                     ,217
      WCM=WCMU-WCML                                                     ,218
      GO TO 920

C     END OF WING ONLY CASE

C     ****************************************************************

C     WING-BODY
                                                                        ,219
 400  CONTINUE                                                          ,220
      NROWB=NROW(1)                                                     ,221
      NROWW=NROW(2)                                                     ,222
      NCOLW=NWING/NROWW                                                 ,223
      CALL FSF (3,NTAPEC,IRR)                                           ,224
      READ (NTAPEC) ALPHAA                                              ,225  ,226  ,227
      READ (NTAPEC) NBODYS,NXLE,ZA,(XB(I),R(I),I=1,NBODYS)              ,228  ,229  ,230  ,231  ,232  ,233
                                                                        ,234  ,235
      READ (NTAPEC) NTHETA,(THETAB(I),I=1,NTHETA)                       ,236  ,237  ,238  ,239  ,240  ,241

      CALL READ (NTAPEI,NTAPEO,NTAPEC,NBODYS,ZDELTA)                    ,242
      ARA=-ALPHAA                                                       ,243
      ALPHAD=ALPHAA*CDR                                                 ,244
      ARADEG=ARA*CDR                                                    ,245
```

```
      WRITE (NTAPEO,6090) ALPHAO                                        ,246  ,247  ,248
      WRITE (NTAPEO,6092) ARADEG                                        ,249  ,250  ,251
      WRITE (NTAPEO,6094) ZA                                            ,252  ,253  ,254
      IF (THICK) 415,405,415                                            ,255
 405  CONTINUE                                                          ,256
      DO 410 J=1,NBODY                                                  ,257
 410  ALPHA(J)=0.0                                                      ,258  ,259
      GO TO 420                                                         ,260
 415  CONTINUE                                                          ,261
      CALL FSF (3,NTAPEC,IRR)                                           ,262
      READ (NTAPEC)(ALPHA(I),I=1,NBODY)                                 ,263  ,264  ,265  ,266  ,267
 420  CONTINUE                                                          ,268
      REWIND NTAPEC                                                     ,269
      DO 425 J=1,NBODY                                                  ,270
      ALPHAB(J)=ALPHAS(J)-ARA*COS(THETA(J))                             ,271
 425  ABX(J)=-ALPHA(J)                                                  ,272  ,273
      GO TO (450,510,610),KASE                                          ,274
 450  CONTINUE

C     CALCULATE WING ALPHA, GIVEN WING CL
                                                                        ,275
      CALL READ (NTAPEI,NTAPEO,NTAPEC,NWING,CL(NS))                     ,276
      DO 460 J=NS,NPANEL                                                ,277
 460  CL(J)=-CL(J)
                                                                        ,278  ,279
      CALL KARMOR (NBODYS,NTHETA,CPCALC,VOUT,ZA,ARA,XP,ZP,XB,R,THETAB
     1,ZDELTA,XC,YC,ZC,THETA,UWBT,VWBT,WWBT,ALPHAX,CPBB,BBCL,BBCD,BBCM)  ,280
      CALL FSF (1,NTAPEF,IRR)                                           ,281
      CALL DCPI (NBODY,NWING,NTAPEF,A,ABX,CL(NS),CL)                    ,282
      REWIND NTAPEF                                                     ,283
      CALL CAMBWB (NBODY,NWING,NTAPEE,NTAPEF,A,CL(NS),ABX,ALPHAW)       ,284
      REWIND NTAPEE                                                     ,285
      REWIND NTAPEF                                                     ,286
      GO TO 700

C     CALCULATE WING CL, GIVEN WING ALPHA
                                                                        ,287
 510  CONTINUE                                                          ,288
      CALL FSF (2,NTAPEC,IRR)                                           ,289
      CALL READ (NTAPEI,NTAPEO,NTAPEC,NWING,ALPHA)                      ,290
      REWIND NTAPEC                                                     ,291
      DO 515 J=1,NWING                                                  ,292
 515  ALPHAW(J)=ALPHA(J)-ARA

C     RETURN POINT FOR CALCULATION OF POLARS                            ,293  ,294
 525  CONTINUE                                                          ,295
      CALL KARMOR (NBODYS,NTHETA,CPCALC,VOUT,ZA,ARA,XP,ZP,XB,R,THETAB
     1,ZDELTA,XC,YC,ZC,THETA,UWBT,VWBT,WWBT,ALPHAX,CPBB,BBCL,BBCD,BBCM)  ,296
      DO 535 J=1,NWING                                                  ,297
 535  AWX(J)=ALPHAW(J)-ALPHAX(J)                                        ,298  ,299
      DO 540 J=1,NBODY                                                  ,300
      READ (NTAPEF) (A(I),I=1,NWING)                                    ,301  ,302  ,303  ,304  ,305
      DO 540 I=1,NWING                                                  ,306
 540  AWX(I)=AWX(I)-A(I)*ABX(J)                                         ,307  ,308  ,309
```

```
      CALL FSF (1,NTAPEE,IRR)                                           ,310
      CALL DCPD (NWING,NTAPEE,A,AWX,CL(NS))                             ,311
      REWIND NTAPEE                                                     ,312
      CALL FSF (1,NTAPEF,IRR)                                           ,313
      CALL DCPI (NBODY,NWING,NTAPEF,A,ABX,CL(NS),CL)                    ,314
      REWIND NTAPEF                                                     ,315
      GO TO 700

C     OPTIMIZE CAMBER FOR GIVEN CL, OR CL AND CM
                                                                        ,316
  610 CONTINUE                                                          ,317
      READ (NTAPEI,5010) CONSNT,CLBAR,XCPBAR                            ,318  ,319  ,320
      IF (CONSNT) 630,625,630                                           ,321
  625 CALL FSF (1,NTAPEA,IRR)                                           ,322
      WRITE (NTAPEO,6082) CLBAR                                         ,323  ,324  ,325
      GO TO 635                                                         ,326
  630 CALL FSF (2,NTAPEA,IRR)                                           ,327
      WRITE (NTAPEO,6082) CLBAR                                         ,328  ,329  ,330
      WRITE (NTAPEO,6084) XCPBAR                                        ,331  ,332  ,333
  635 CONTINUE                                                          ,334
      CALL KARMOR (NBODYS,NTHETA,CPCALC,VOUT,ZA,ARA,XP,ZP,XB,R,THETAB
     1,ZDELTA,XC,YC,ZC,THETA,UWBT,VWBT,WWBT,ALPHAX,CPBB,BBCL,BBCD,BBCM)  ,335
      IF (THICK) 645,640,645                                            ,336
  640 NTAPEZ=0                                                          ,337
      GO TO 650                                                         ,338
  645 CALL FSF (6,NTAPEC,IRR)                                           ,339
      NTAPEZ=NTAPEC                                                     ,340
  650 CONTINUE                                                          ,341
      CALL OPTMWB (NWING,NBODY,NTAPEA,NTAPEF,NTAPEZ,A,B,ALPHA,ALPHAX
     1,AREA(NS),CONSNT,CLBAR,XCPBAR,RFAREA,CL(NS))                      ,342
      REWIND NTAPEA                                                     ,343
      REWIND NTAPEC                                                     ,344
      CALL FSF (1,NTAPEF,IRR)                                           ,345
      CALL DCPI (NBODY,NWING,NTAPEF,A,ABX,CL(NS),CL)                    ,346
      REWIND NTAPEF                                                     ,347
      CALL CAMBWB (NBODY,NWING,NTAPEE,NTAPEF,A,CL(NS),ABX,ALPHAW)       ,348
      REWIND NTAPEE                                                     ,349
      REWIND NTAPEF
                                                                        ,350
  700 CONTINUE
                                                                        ,351
      CALL FSF (1,NTAPED,IRR)                                           ,352
      CALL CVEL (NWING,NBODY,NTAPED,A,B,C,CL,UWB,VWB,WWB)               ,353
      CALL FSF (2,NTAPED,IRR)                                           ,354
      CALL CVEL (NWING,NWING,NTAPED,A,B,C,CL(NS),UWW,VWW,WWW)           ,355
      REWIND NTAPED
                                                                        ,356
      CALL RITE (NFMT(6),NTAPED,NWING,NROWW,NCOLW,DUMMY,UWB,VWB,WWB)    ,357
      CALL RITE (NFMT(4),NTAPED,NWING,NROWW,NCOLW,DUMMY,UWW,VWW,WWW)    ,358
      CALL RITE (NFMT(7),NTAPED,NWING,NROWW,NCOLW,DUMMY,UWBT,VWBT,WWBT)

C     COMPUTE COEFFICIENT OF PRESSURE ON WING WITH OR WITHOUT WING
```

```
C     THICKNESS                                                          ,359
      IF (THICK) 705,720,705                                             ,360
  705 CONTINUE                                                           ,361
      CALL FSF (5,NTAPEC,IRR)                                            ,362
      READ (NTAPEC) (UWWT(I),VWWT(I),WWWT(I),I=1,NWING)                  ,363  ,364  ,365  ,366  ,367
      REWIND NTAPEC                                                      ,368
      CALL RITE (NFMT(5),NTAPED,NWING,NROWW,NCOLW,DUMMY,UWWT,VWWT,WWWT)  ,369
      DO 710 J=1,NWING                                                   ,370
      U(J)=UWB(J)+UWW(J)+UWBT(J)+UWWT(J)                                 ,371
      V(J)=VWB(J)+VWW(J)+VWBT(J)+VWWT(J)                                 ,372
  710 W(J)=WWB(J)+WWW(J)+WWBT(J)+WWWT(J)                                 ,373  ,374
      CALL CP (NWING,BETA2,CPCALC,U,V,W,CPU)                             ,375
      DO 715 J=1,NWING                                                   ,376
      U(J)=UWB(J)-UWW(J)+UWBT(J)+UWWT(J)                                 ,377
  715 W(J)=WWB(J)+WWW(J)+WWBT(J)-WWWT(J)                                 ,378  ,379
      CALL CP (NWING,BETA2,CPCALC,U,V,W,CPL)                             ,380
      DO 718 J=1,NWING                                                   ,381
      ALPHAU(J)=ALPHAW(J)+ALPHAT(J)                                      ,382
  718 ALPHAL(J)=ALPHAW(J)-ALPHAT(J)                                      ,383  ,384
      GO TO 730
                                                                         ,385
  720 CONTINUE                                                           ,386
      DO 722 J=1,NWING                                                   ,387
      U(J)=UWB(J)+UWW(J)+UWBT(J)                                         ,388
      V(J)=VWB(J)+VWW(J)+VWBT(J)                                         ,389
 722  W(J)=WWB(J)+WWW(J)+WWBT(J)                                         ,390  ,391
      CALL CP (NWING,BETA2,CPCALC,U,V,W,CPU)                             ,392
      DO 724 J=1,NWING                                                   ,393
 724  U(J)=UWB(J)-UWW(J)+UWBT(J)                                         ,394  ,395
      CALL CP (NWING,BETA2,CPCALC,U,V,W,CPL)                             ,396
      DO 725 J=1,NWING                                                   ,397
      ALPHAU(J)=ALPHAW(J)                                                ,398
  725 ALPHAL(J)=ALPHAW(J)                                                ,399  ,400
  730 CONTINUE

C     COMPUTE COEFFICIENTS OF LIFT, DRAG, MOMENT ON WING                 ,401
      CALL INTPOL (NWING,NROWW,RATIOX,RATIOD,CHORD(NS),ALPHAU,ALPHA)     ,402
      CALL WLDM (NWING,NROWW,XP,ZP,REAREA,AREA(NS),XBAR(NS),ZBAR(NS)
     1,ALPHA,THETA(NS),CPU,WSCLU,WSCDU,WCLU,WCDU,WCMU)                   ,403
      CALL INTPOL (NWING,NROWW,RATIOX,RATIOD,CHORD(NS),ALPHAL,ALPHA)     ,404
      CALL WLDM (NWING,NROWW,XP,ZP,REAREA,AREA(NS),XBAR(NS),ZBAR(NS)
     1,ALPHA,THETA(NS),CPL,WSCLL,WSCDL,WCLL,WCDL,WCML)                   ,405
      DO 738 J=1,NCOLW                                                   ,406
      WSCL(J)=WSCLU(J)-WSCLL(J)                                          ,407
  738 WSCD(J)=WSCDU(J)-WSCDL(J)                                          ,408  ,409
      WCL=WCLU-WCLL                                                      ,410
      WCD=WCDU-WCDL                                                      ,411
      WCM=WCMU-WCML

                                                                         ,412
      CALL CVEL (NBODY,NBODY,NTAPED,A,B,C,CL,UBB,VBB,WBB)                ,413
      CALL FSF (2,NTAPED,IRR)                                            ,414
      CALL CVEL (NBODY,NWING,NTAPED,A,B,C,CL(NS),UBW,VBW,WBW)            ,415
      REWIND NTAPED
```

```
                                                                          ,416
      CALL RITE (NFMT(1),NTAPEO,NBODY,NROWB,NTHETA,THETAB,UBB,VBB,WBB)    ,417
      CALL RITE (NFMT(2),NTAPEO,NBODY,NROWB,NTHETA,THETAB,UBW,VBW,WBW)

C     COMPUTE COEFFICIENT OF PRESSURE ON BODY WITH OR WITHOUT THE
C     EFFECT OF WING THICKNESS                                            ,418
      IF (THICK) 745,760,745                                              ,419
  745 CONTINUE                                                            ,420
      CALL FSF (4,NTAPEC,IRR)                                             ,421
      READ (NTAPEC) (UBWT(I),VBWT(I),WBWT(I),I=1,NBODY)                   ,422  ,423  ,424  ,425  ,426
      REWIND NTAPEC                                                       ,427
      CALL RITE (NFMT(3),NTAPEO,NBODY,NROWB,NTHETA,THETAB,UBWT,VBWT,WBWT
     1)                                                                   ,428
      DO 750 J=1,NBODY                                                    ,429
      U(J)=UBB(J)+UBW(J)+UBWT(J)                                          ,430
      V(J)=VBB(J)+VBW(J)+VBWT(J)                                          ,431
  750 W(J)=WBB(J)+WBW(J)+WBWT(J)                                          ,432  ,433
      GO TO 780

                                                                          ,434
  760 CONTINUE                                                            ,435
      DO 770 J=1,NBODY                                                    ,436
      U(J)=UBB(J)+UBW(J)                                                  ,437
      V(J)=VBB(J)+VBW(J)                                                  ,438
  770 W(J)=WBB(J)+WBW(J)                                                  ,439  ,440
  780 CONTINUE                                                            ,441
      CALL CP (NBODY,BETA2,CPCALC,U,V,W,CPB)

C     COMPUTE COEFFICIENTS OF LIFT, DRAG, MOMENT ON BODY PANELS           ,442
      CALL INTPOL (NBODY,NROWB,RATIOX,RATIOD,CHORD,ALPHAB,ALPHA)          ,443
      CALL BLDM (NBODY,XP,ZP,RFAREA,AREA,XBAR,ZBAR,ALPHA,THETA,CPB
     1,BWCL,BWCD,BWCM)
                                                                          ,444
      BBCL=BBCL/RFAREA                                                    ,445
      BBCD=BBCD/RFAREA                                                    ,446
      BBCM=BBCM/RFAREA                                                    ,447
      BCL=BWCL+BBCL                                                       ,448
      BCD=BWCD+BBCD                                                       ,449
      BCM=BWCM+BBCM                                                       ,450
      WBCL=BCL+WCL                                                        ,451
      WBCD=BCD+WCD                                                        ,452
      WBCM=BCM+WCM                                                        ,453
      GO TO 940

C     END OF WING-BODY CASE

C     ****************************************************************

C     PRINT OUT OF BODY ONLY CASE
                                                                          ,454
  900 CONTINUE                                                            ,455
      WRITE (NTAPEO,6020)                                                 ,456  ,457
      WRITE (NTAPEO,6080) BCD,BCL,BCM                                     ,458  ,459  ,460
```

```
      WRITE (NTAPEO,6030)                                          ,461  ,462
      WRITE (NTAPEO,6036) (THETAB(I),I=1,NTHETA)                   ,463  ,464  ,465  ,466  ,467
      WRITE (NTAPEO,6038)                                          ,468  ,469
      DO 903 I=1,NBODYS                                            ,470
  903 WRITE (NTAPEO,6040) XB(I),(CPBB(I,J),J=1,NTHETA)
                                                                   ,471  ,472  ,473  ,474  ,475  ,476
                                                                   ,477
      IF (POLAR) 905,10,905                                        ,478
  905 CONTINUE                                                     ,479
      READ (NTAPEI,5010) DADEG                                     ,480  ,481  ,482
      IF (DADEG) 910,10,910                                        ,483
  910 CONTINUE                                                     ,484
      DARAD=DADEG/CDR                                              ,485
      ARA=ARA+DARAD                                                ,486
      ARADEG=ARA*CDR                                               ,487
      WRITE (NTAPEO,6002) TITLE                                    ,488  ,489  ,490
      WRITE (NTAPEO,6008) DADEG                                    ,491  ,492  ,493
      WRITE (NTAPEO,6092) ARADEG                                   ,494  ,495  ,496
      GO TO 230

C     PRINT OUT OF WING ONLY CASE
                                                                   ,497
  920 CONTINUE
                                                                   ,498
      DO 922 J=1,NWING                                             ,499
  922 CL(J)=-CL(J)                                                 ,500  ,501
      WRITE (NTAPEO,6018)                                          ,502  ,503
      WRITE (NTAPEO,6080) WCD,WCL,WCM                              ,504  ,505  ,506
      WRITE (NTAPEO,6060)                                          ,507  ,508
      WRITE (NTAPEO,6064) (I,I=1,NCOLW)                            ,509  ,510  ,511  ,512  ,513
      WRITE (NTAPEO,6000) (WSCD(I),I=1,NCOLW)                      ,514  ,515  ,516  ,517  ,518
      WRITE (NTAPEO,6062)                                          ,519  ,520
      WRITE (NTAPEO,6064) (I,I=1,NCOLW)                            ,521  ,522  ,523  ,524  ,525
      WRITE (NTAPEO,6000) (WSCL(I),I=1,NCOLW)                      ,526  ,527  ,528  ,529  ,530
      WRITE (NTAPEO,6050)                                          ,531  ,532
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,CPU)                     ,533
      WRITE (NTAPEO,6052)                                          ,534  ,535
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,CPL)                     ,536
      WRITE (NTAPEO,6054)                                          ,537  ,538
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,CL)                      ,539
      WRITE (NTAPEO,6056)                                          ,540  ,541
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,ALPHAU)                  ,542
      WRITE (NTAPEO,6058)                                          ,543  ,544
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,ALPHAL)
                                                                   ,545
      IF (POLAR) 925,10,925                                        ,546
  925 CONTINUE                                                     ,547
      READ (NTAPEI,5010) DADEG                                     ,548  ,549  ,550
      IF (DADEG) 930,10,930                                        ,551
  930 CONTINUE                                                     ,552
      DARAD=DADEG/CDR                                              ,553
      DO 935 I=1,NWING                                             ,554
```

```
  935 ALPHAW(I)=ALPHAW(I)-DARAD                                          ,555  ,556
      ARA=ARA+DARAD                                                      ,557
      ARADEG=ARA*CDR                                                     ,558
      WRITE (NTAPEO,6002) TITLE                                          ,559  ,560  ,561
      WRITE (NTAPEO,6008) DADEG                                          ,562  ,563  ,564
      WRITE (NTAPEO,6092) ARADEG                                         ,565  ,566  ,567
      GO TO 340

C     PRINT OUT OF WING-BODY CASE
                                                                         ,568
  940 CONTINUE
                                                                         ,569
      DO 942 J=NS,NPANEL                                                 ,570
  942 CL(J)=-CL(J)                                                       ,571  ,572
      WRITE (NTAPEO,6010)                                                ,573  ,574
      WRITE (NTAPEO,6080) BBCD,BBCL,BBCM                                 ,575  ,576  ,577
      WRITE (NTAPEO,6030)                                                ,578  ,579
      WRITE (NTAPEO,6036) (THETAB(I),I=1,NTHETA)                         ,580  ,581  ,582  ,583  ,584
      WRITE (NTAPEO,6038)                                                ,585  ,586
      DO 943 I=1,NBODYS                                                  ,587
  943 WRITE (NTAPEO,6040) XB(I),(CPBB(I,J),J=1,NTHETA)                   ,588  ,589  ,590  ,591  ,592  ,593
                                                                         ,594
      WRITE (NTAPEO,6014)                                                ,595  ,596
      WRITE (NTAPEO,6080) BWCD,BWCL,BWCM                                 ,597  ,598  ,599
      WRITE (NTAPEO,6032)                                                ,600  ,601
      CALL OUTB (NTAPEO,NBODY,NTHETA,NROWB,THETAB,CPB)                   ,602
      WRITE (NTAPEO,6034)                                                ,603  ,604
      CALL OUTB (NTAPEO,NBODY,NTHETA,NROWB,THETAB,ALPHAB)                ,605
      WRITE (NTAPEO,6012)                                                ,606  ,607
      WRITE (NTAPEO,6080) WCD,WCL,WCM                                    ,608  ,609  ,610
      WRITE (NTAPEO,6060)                                                ,611  ,612
      WRITE (NTAPEO,6064) (I,I=1,NCOLW)                                  ,613  ,614  ,615  ,616  ,617
      WRITE (NTAPEO,6000) (WSCD(I),I=1,NCOLW)                            ,618  ,619  ,620  ,621  ,622
      WRITE (NTAPEO,6062)                                                ,623  ,624
      WRITE (NTAPEO,6064) (I,I=1,NCOLW)                                  ,625  ,626  ,627  ,628  ,629
      WRITE (NTAPEO,6000) (WSCL(I),I=1,NCOLW)                            ,630  ,631  ,632  ,633  ,634
      WRITE (NTAPEO,6050)                                                ,635  ,636
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,CPU)                           ,637
      WRITE (NTAPEO,6052)                                                ,638  ,639
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,CPL)                           ,640
      WRITE (NTAPEO,6054)                                                ,641  ,642
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,CL(NS))                        ,643
      WRITE (NTAPEO,6056)                                                ,644  ,645
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,ALPHAU)                        ,646
      WRITE (NTAPEO,6058)                                                ,647  ,648
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,ALPHAL)                        ,649
      WRITE (NTAPEO,6016)                                                ,650  ,651
      WRITE (NTAPEO,6080) WBCD,WBCL,WBCM
                                                                         ,652  ,653  ,654
      IF (POLAR) 948,10,948                                              ,655
  948 CONTINUE                                                           ,656
      READ (NTAPEI,5010) DADEG                                           ,657  ,658  ,659
```

```
      IF (DADEG) 950,10,950                                          ,660
  950 CONTINUE                                                       ,661
      DARAD=DADEG/CDR                                                ,662
      DO 955 I=1,NBODY                                               ,663
  955 ALPHAB(I)=ALPHAB(I)-DARAD*COS(THETA(I))                        ,664  ,665
      DO 960 I=1,NWING                                               ,666
  960 ALPHAW(I)=ALPHAW(I)-DARAD                                      ,667  ,668
      ARA=ARA+DARAD                                                  ,669
      ARADEG=ARA*CDR                                                 ,670
      WRITE (NTAPEO,6002) TITLE                                      ,671  ,672  ,673
      WRITE (NTAPEO,6008) DADEG                                      ,674  ,675  ,676
      WRITE (NTAPEO,6092) ARADEG                                     ,677  ,678  ,679
      GO TO 525
                                                                     ,680
 5000 FORMAT(12A6)
 5010 FORMAT(7F10.0)


 6000 FORMAT(1H ,F29.5,9F10.5)
 6002 FORMAT(1H1,12A6)
 6008 FORMAT(1H0,23HDELTA ANGLE OF ATTACK =,F10.4,2X,4HDEG.)
 6010 FORMAT(1H0//48H PRESSURES, FORCES, AND MOMENTS ON ISOLATED BODY)
 6012 FORMAT(1H0//66H PRESSURES, FORCES, AND MOMENTS ON WING PANELS IN P
     1RESENCE OF BODY)
 6014 FORMAT(1H0//70H INCREMENTAL PRESSURES, FORCES, AND MOMENTS ON BODY
     1 PANELS DUE TO WING)
 6016 FORMAT(1H0//44H FORCES AND MOMENTS ON WING-BODY COMBINATION)
 6018 FORMAT(1H0//38H PRESSURES, FORCES AND MOMENTS ON WING)
 6020 FORMAT(1H0//38H PRESSURES, FORCES AND MOMENTS ON BODY)
 6030 FORMAT(1H0/32H  BODY PRESSURE COEFFICIENTS(CP))
 6032 FORMAT(1H0/37H  BODY PANEL PRESSURE COEFFICIENT(CP))
 6034 FORMAT(1H0/18H  BODY PANEL SLOPE)
 6036 FORMAT(1H0,4X,11HTHETA(DEG.),F14.4,9F10.4,(F29.4,9F10.4))
 6038 FORMAT(1H ,9X,1HX)
 6040 FORMAT(1H ,F14.4,F15.5,9F10.5,(F29.5,9F10.5))
 6050 FORMAT(1H0/52H  UPPER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)
     1)
 6052 FORMAT(1H0/52H  LOWER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)
     1)
 6054 FORMAT(1H0/36H  WING PANEL PRESSURE DIFFERENCE(CL))
 6056 FORMAT(1H0/32H  UPPER SURFACE WING PANEL SLOPE)
 6058 FORMAT(1H0/32H  LOWER SURFACE WING PANEL SLOPE)
 6060 FORMAT(1H0/26H  SPANWISE CD DISTRIBUTION)
 6062 FORMAT(1H0/26H  SPANWISE CL DISTRIBUTION)
 6064 FORMAT(1H0,2X,16HSPANWISE STATION,I8,9I10,(18X,I8,9I10))
 6080 FORMAT(1H0/6H  CD =,F10.5,10X,4HCL =,F10.5,10X,4HCM =,F10.5)
 6082 FORMAT(1H0,27HWING OPTIMIZED FOR CL BAR =,F10.4)
 6084 FORMAT(1H , 18X,9HXCP BAR =,F10.4)
 6090 FORMAT(1H0,56HINCLINATION OF BODY AXIS WITH RESPECT TO DEFINING AX
     1IS =,F10.4,2X,4HDEG.)
 6092 FORMAT(1H0,43HANGLE OF ATTACK WITH RESPECT TO BODY AXIS =,F10.4,
     1,2X,4HDEG.)
 6094 FORMAT(1H0,38HHEIGHT OF WING PLANE ABOVE BODY AXIS =,F10.4)
```

```
999  CONTINUE                                                          ,681
     RETURN                                                            ,682
     END                                                               ,683
```

```
      SUBROUTINE FROOTA(R,J)
C
      DIMENSION R(10),F(10)
C
C     FROOTA IS A ROOT FINDER, DESIGNED TO BE CALLED REPEATEDLY
C     UNTIL CODE J INDICATES THAT A ROOT HAS BEEN FOUND.
C
C     ASSUME A SINGLE-VALUED FUNCTION F(X) WHICH THE CALLING PROGRAM
C     CAN EVALUATE FOR ANY X.  FURTHER, THE PROGRAM KNOWS THAT THE
C     ROOT LIES BETWEEN XA AND XB.
C
C     ON THE FIRST CALL FOR A CASE, THE PROGRAM STORES
C     XA,F(XA),XB,F(XB) IN R(1),R(2),R(3),R(4), 0. IN R(9),
C     AND A TOLERANCE IN R(10).
C     FROOTA THEN COMPUTES TWO VALUES OF X, SAY XC AND XD,WHICH IT
C     STORES IN R(1) AND R(3), AND SETS J=1 WHICH INDICATES THAT
C     IT WANTS TO BE CALLED AGAIN.  THE CALLING PROGRAM
C     RESPONDS BY STORING F(XC) IN R(2) AND F(XD) IN R(4), AND
C     THEN CALLING FROOTA.THIS PROCESS CONTINUES UNTIL FROOTA
C     SIGNALS THAT THE ROOT HAS BEEN FOUND BY SETTING J=0.
C
C     FROOTA KEEPS THE POINTS WHICH BRACKET THE ROOT IN R(5)-R(8).
C     THE ROOT IS CONSIDERED TO BE FOUND WHEN
C     (1) AN EXACT ROOT IS FOUND,
C  OR (2) THE DIFFERENCE IN X-VALUES OF THE BRACKETING POINTS IS
C         SMALLER THAN THE GIVEN TOLERANCE IN R(10)
C  OR (3) THE ROOT HAS BEEN LOCATED TO THE NEAREST 1-BIT CHANGE IN X.
C
C     FROOTA ESTIMATES ONE VALUE OF X AS HALF-WAY BETWEEN THE X,S OF
C     THE BRACKETING POINTS, AND THE OTHER BY ASSUMING THE
C     FUNCTION IS LINEAR BETWEEN THE BRACKETING POINTS.
C
C     J IS SET TO -2 OR -3 IF IT DETECTS AN ERROR.
C     J = -2 IF R(9) IS .LT. 0.
C     J = -3 IF (ON THE 1ST CALL ONLY) THE POINTS IN R(1)-R(4)
C     DO NOT BRACKET THE ROOT (I.E., IF F(A)*F(B) .GT. 0.)
C
C     R(9) IS A COUNTER (INCREASED BY 1. AT EACH CALL).

      J=1                                                               ,1
      DO 100 I=1,10                                                     ,2
 100  F(I)=R(I)                                                         ,3    ,4
      F(9)=F(9)+1.
C                                                                       ,5
      IF(F(9)-1.)200,300,1000                                           ,6
 200  J=-2                                                              ,7
      GO TO 8000
C                                                                       ,8
 300  IF(F(2)*F(4))500,1300,400                                         ,9
 400  J=-3                                                              ,10
      GO TO 8000
C     THE BRACKETING POINT WITH THE SMALLER X IS STORED IN R(5),R(6)    ,11
 500  IF(F(1) .LT. F(3)) GO TO 1800                                     ,12   ,13   ,14
```

```
                                                                          ,15
 600  F(5)=F(3)                                                           ,16
      F(6)=F(4)
      GO TO 1700                                                          ,17
C
 1000 IF(F(1) .LE. F(3)) GO TO 1200                                       ,18   ,19   ,20
C     REVERSE THE 2 NEW POINTS SO ONE WITH SMALLER X WILL BE FIRST        ,21
 1100 F(1)=F(3)                                                           ,22
      F(2)=F(4)                                                           ,23
      F(3)=R(1)
      F(4)=R(2)                                                           ,24
C                                                                         ,25
 1200 CONTINUE
      IF(F(2)*F(4)) 1800,1300,1500                                        ,26
C                                                                         ,27
 1300 CONTINUE                                                            ,28
      IF(F(2))1400,3000,1400                                              ,29
 1400 F(1)=F(3)
      GO TO 3000                                                          ,30
C                                                                         ,31
 1500 IF(F(2)*F(6)) 1700,1700,1600                                        ,32
 1600 F(5)=F(3)                                                           ,33
      F(6)=F(4)
      GO TO 2000                                                          ,34
C                                                                         ,35
 1700 CONTINUE                                                            ,36
      F(7)=F(1)                                                           ,37
      F(8)=F(2)
      GO TO 2000                                                          ,38
C                                                                         ,39
 1800 CONTINUE                                                            ,40
      F(5)=F(1)                                                           ,41
      F(6)=F(2)                                                           ,42
      F(7)=F(3)                                                           ,43
      F(7)=F(3)                                                           ,44
      F(8)=F(4)
      IF(F(1) .EQ. F(7)) GO TO 3000                                       ,45   ,46   ,47
C
 2000 CONTINUE                                                            ,48
C     FIND NEW TRIAL VALUES OF X                                          ,49
      F(3)=.5*(F(5)+F(7))                                                 ,50
      U=F(7)-F(5)
      V=F(8)-F(6)                                                         ,51
C     WE COMPUTE F(1) IN TWO WAYS TO IMPROVE ACCURACY                     ,52
      IF(ABS(F(6))-ABS(F(8))) 2100,2100,2300                              ,53
 2100 F(1)=F(5)-(F(6)/V)*U                                                ,54   ,55   ,56
      IF(F(1) .EQ. F(5)) GO TO 3000                                       ,57
      GO TO 2500
 2300 F(1)=F(7)-(F(8)/V)*U                                                ,58
C                                                                         ,59
 2500 CONTINUE
      IF(U .GT. ABS(F(10))) GO TO 8000                                    ,60   ,61   ,62
C                                                                         ,63
 3000 CONTINUE
      J=0
```

```
EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

C                                                                    ,64
 8000 CONTINUE                                                       ,65
      DO 8100 I=1,10                                                 ,66
 8100 R(I)=F(I)                                                      ,67    ,68
 9000 RETURN                                                         ,69
      END                                                            ,70
```

```
ETAN                                                                    07/31/65
IBJOB LIBRARY / TANGENT FUNCTION

                                    *     REWRITTEN FOR IBMAP / 3-3-63 / E. WRIGHT - TRANSPORT -BOEING      FTAN0020
                                    *     METHODS MEMO 7090-43                                              FTAN0030
                                    *                                                                       FTAN0040
                                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN0050
                                    *                                                                       FTAN0060
                                    *     A=TAN (B)                                                         FTAN0070
                                    *                                                                       FTAN0080
                                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN0090
                                                                                                            FTAN0100
                                    *      ENTRY   TAN                                                      FTAN0110
                                                                                                            FTAN0120
                                                                                                            FTAN0130
                                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN0140

BINARY CARD (NOT PUNCHED)
      00000  1 00000 0 00004  10001  TAN    SAVE    1                                                      FTAN0150
      00001  0774 00 1 00000  10000
      00002  0774 00 4 00000  10000
      00003  0020 00 4 00001  10000
      00004  0634 00 4 05000  10011
      00005  0634 00 4 00162  10001
      00006  0634 00 4 00002  10001
      00007  0634 00 1 00001  10001
      00010  0500 60 4 00003  10000         CLA*    3,4                                                    FTAN0160
      00011  0601 00 0 03000  10011         STQ     COMMON        X                                        FTAN0170
      00012  0760 00 0 00003  10000         SSP                   ABS X                                    FTAN0180
      00013  0560 00 0 00164  10001         LDQ    =210828713.    TEST IF X GREATER THAN PI*(2**26)        FTAN0190
      00014  0040 00 0 00116  10001         TLQ     ERROR1        YES, ERROR                               FTAN0200
      00015  0560 00 0 00165  10001         LDQ    =3.14159265    NO,  TEST IF X GREATER THAN PI           FTAN0210
      00016  0040 00 0 00023  10001         TLQ     TAN1          YES, NEEDS REDUCING                      FTAN0220
      00017  0560 00 0 00166  10001         LDQ    =.0001         NO,  TEST IF X GREATER THAN .0001        FTAN0230
      00020  0040 00 0 00034  10001         TLQ     TAN2          YES, X IN PROPER RANGE                   FTAN0240
      00021  0500 00 0 03000  10011         CLA     COMMON        NO                                       FTAN0250
                     00022                  RETURN  TAN           RETURN X = TAN X                         FTAN0260
                                    *                                                                       FTAN0270
                                    * REDUCE X TO MODULO PI                                                 FTAN0280
                                    *                                                                       FTAN0290

BINARY CARD (NOT PUNCHED)
      00023  4500 00 0 03000  10011  TAN1   CAL     COMMON        X GREATER THAN PI (N.M*PI)               FTAN0300
      00024  0630 00 0 00167  10001         STP    =67108864.     MAKE SIGN SAME AS X                      FTAN0310
      00025  4130 00 0 00000  10000         XCL                                                             FTAN0320
      00026  0260 00 0 00170  10001         FMP    =.318309886    1/PI (LEAVES N.M)                        FTAN0330
      00027  0300 00 0 00167  10001         FAD    =67108864.     2**26 (SEPARATES N AND M)                FTAN0340
      00030  0131 00 0 00000  10000         XCA                   KEEP FRACTIONAL PART (.M)                FTAN0350
      00031  0300 00 0 00171  10001         FAD    =17179869184   NORMALIZE                                FTAN0360
      00032  0241 00 0 00170  10001         FDP    =.318309886    1/PI (LEAVES .M*PI)                      FTAN0370
      00033  4600 00 0 03000  10011         STQ     COMMON        X REPLACED BY X MODULO PI                FTAN0380
                                    *                                                                       FTAN0390
                                    * TEST IF X IS A MULTIPLE  OF PI/2                                      FTAN0400
                                    *                                                                       FTAN0410
      00034  0500 00 0 00172  10001  TAN2   CLA    =1.57079633    PI/2                                     FTAN0420
      00035  4400 00 0 03000  10011         SBM     COMMON                                                  FTAN0430
```

FTAN
IBJOB LIBRARY / TANGENT FUNCTION 07/31/65

```
00036  0771 00 0 00001  10000          ARS     1              ALLOW 1 BIT RANGE                    FTAN0440
00037  0100 00 0 00124  10001          TZE     ERROR2         ERROR, X=PI/2                        FTAN0450
                                *                                                                   FTAN0460
                                * IF X GREATER THAN PI/2, PLACE IN PROPER QUADRANT                  FTAN0470
                                *                                                                   FTAN0480
00040  0500 00 0 03000  10011          CLA     COMMON         X                                    FTAN0490
00041  4130 00 0 00000  10000          XCL                    ABS X                                FTAN0500
00042  0500 00 0 00172  10001          CLA     =1.57079633    TEST IF X GREATER THAN PI/2          FTAN0510
00043  0040 00 0 00053  10001          TLQ     TAN4           NO,  X OK                            FTAN0520
00044  0500 00 0 03000  10011          CLA     COMMON         YES, PLACE IN PROPER QUADRANT        FTAN0530
00045  4120 00 0 00051  10001          TMI     TAN3                                                FTAN0540

BINARY CARD (NOT PUNCHED)
00046  0302 00 0 00165  10001          FSB     =3.14159265    QUADRANT 2                           FTAN0550
00047  0601 00 0 03000  10011          STO     COMMON         X REPLACED BY X-PI                   FTAN0560
00050  0020 00 0 00054  10001          TRA     TAN5                                                FTAN0570
00051  0300 00 0 00165  10001   TAN3   FAD     =3.14159265    QUADRANT 3                           FTAN0580
00052  0601 00 0 03000  10011          STO     COMMON         X REPLACED BY -X+PI                  FTAN0590
                                *                                                                   FTAN0600
                                *                                                                   FTAN0610
00053  0500 00 0 03000  10011   TAN4   CLA     COMMON         X                                    FTAN0620
00054  0760 00 0 00003  10000   TAN5   SSP                    ABS X                                FTAN0630
00055  0560 00 0 00173  10001          LDQ     =0.78539816    TEST IF X GREATER THAN PI/4          FTAN0640
00056  0040 00 0 00061  10001          TLQ     TAN6           YES                                  FTAN0650
                                *                                                                   FTAN0660
                                * TAN X = CONTINUED FRACTION                                       FTAN0670
                                *                                                                   FTAN0680
00057  0074 00 4 00076  10001          TSX     CONTFR,4       NO                                   FTAN0690
00060  0020 00 0 00074  10001          TRA     TAN9                                                FTAN0700
                                *                                                                   FTAN0710
                                * TAN X = 1/TAN(PI/2-ABS X)                                        FTAN0720
                                *                                                                   FTAN0730
00061  0500 00 0 03000  10011   TAN6   CLA     COMMON         IS X NEGATIVE                        FTAN0740
00062  4120 00 0 00066  10001          TMI     TAN7           YES                                  FTAN0750
00063  0302 00 0 00172  10001          FSB     =1.57079633    NO                                   FTAN0760
00064  0760 00 0 00002  10000          CHS                    REPLACE X BY PI/2-X                  FTAN0770
00065  0020 00 0 00070  10001          TRA     TAN8                                                FTAN0780
00066  0300 00 0 00172  10001   TAN7   FAD     =1.57079633                                         FTAN0790
00067  0760 00 0 00002  10000          CHS                    REPLACE X BY -(PI/2-ABS X)           FTAN0800
00070  0601 00 0 03000  10011   TAN8   STO     COMMON                                              FTAN0810

BINARY CARD (NOT PUNCHED)
00071  0074 00 4 00076  10001          TSX     CONTFR,4       TAN(PI/2-ABS X)                      FTAN0820
00072  0500 00 0 00161  10001          CLA     A1             1.0                                  FTAN0830
00073  0241 00 0 03002  10011          FDP     COMMON+2       TAN X = 1/TAN(PI/2-ABS X)            FTAN0840
                                *                                                                   FTAN0850
00074  0131 00 0 00000  10000   TAN9   XCA                    ANSWER IN ACCUMULATOR                FTAN0860
                  00075                RETURN  TAN            RETURN                               FTAN0870
                                *                                                                   FTAN0880
                                * TAN X = X/(1-X**2/(3-X**2/(5-X**2/(7-X**2/(9-X**2/11)))))        FTAN0890
                                *                                                                   FTAN0900
00076  0774 00 1 00005  10000   CONTFR AXT     5,1                                                 FTAN0910
00077  0560 00 0 03000  10011          LDQ     COMMON         X                                    FTAN0920
00100  0260 00 0 03000  10011          FMP     COMMON                                              FTAN0930
```

```
                FTAN                                                                07/31/65
                IBJOB LIBRARY / TANGENT FUNCTION

        00101   0601 00 0 03001   10011          STO     COMMON+1      X**2                                  FTAN0940
        00102   0500 00 0 00154   10001          CLA     A6            A(6)=11.                              FTAN0950
        00103   0601 00 0 03002   10011          STO     COMMON+2                                            FTAN0960
        00104   0502 00 0 03001   10011   TAN12 CLS     COMMON+1      -X**2                                 FTAN0970
        00105   0241 00 0 03002   10011          FDP     COMMON+2                                            FTAN0980
        00106   0131 00 0 00000   10000          XCA                                                         FTAN0990
        00107   0300 00 1 00162   10001          FAD     A1+1,1        A(I)                                  FTAN1000
        00110   0601 00 0 03002   10011          STO     COMMON+2                                            FTAN1010
        00111   2 00001 1 00104   10001          TIX     TAN12,1,1     5 TIMES                               FTAN1020
        00112   0500 00 0 03000   10011          CLA     COMMON        X                                     FTAN1030
        00113   0241 00 0 03002   10011          FDP     COMMON+2                                            FTAN1040

BINARY CARD (NOT PUNCHED)
        00114   4600 00 0 03002   10011          STO     COMMON+2                                            FTAN1050
        00115   0020 00 4 00001   10000          TRA     1,4                                                 FTAN1060
                                          *                                                                   FTAN1070
                                          * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN1080
                                          *                                                                   FTAN1090
        00116   000000000000      00010   ERROR1 CALL    .EXEM.(BAIL)   ABS X TOO LARGE                       FTAN1100
        00116   0074 00 4 04000   10011
        00117   1 00001 0 01003   10011
        00120   0 00162 0 00202   10100
        00121   0 00000 0 00134   10001
        00122   4754 00 0 00000   10000          ZAC                    SET RESULT TO ZERO                   FTAN1110
                          00123                  RETURN  TAN                                                 FTAN1120
        00124   000000000000      00010   ERROR2 CALL    .EXEM.(GIVE)   X ODD MULTIPLE                        FTAN1130
        00124   0074 00 4 04000   10011
        00125   1 00001 0 01003   10011
        00126   0 00162 0 00217   10100
        00127   0 00000 0 00144   10001
        00130   0500 00 0 03000   10011          CLA     COMMON        SET RESULT TO LARGEST NUMBER          FTAN1140
        00131   0760 00 0 00000   10000          CLM                     WITH CORRECT SIGN.                  FTAN1150
        00132   0760 00 0 00006   10000          COM                                                         FTAN1160
                          00133                  RETURN  TAN                                                 FTAN1170
        00134   000000000103      10000   BAIL   DEC     67                                                  FTAN1171

BINARY CARD (NOT PUNCHED)
        00135   0 00003 0 00137   10001          PZE     MESA,,3                                             FTAN1172
        00136   0 00002 0 00142   10001          PZE     MESB,,2                                             FTAN1173
        00137   002122626067      10000   MESA   BCI     3,0ABS X TOO LARGE                                  FTAN1174
        00140   606346466043      10000
        00141   215127256060      10000
        00142   002145626013      10000   MESB   BCI     2,0ANS = 0                                          FTAN1175
        00143   600060606060      10000
        00144   000000000102      10000   GIVE   DEC     66                                                  FTAN1176
        00145   0 00003 0 00147   10001          PZE     MESC,,3                                             FTAN1177
        00146   0 00002 0 00152   10001          PZE     MESD,,2                                             FTAN1178
        00147   006760462424      10000   MESC   BCI     3,0X ODD MULTIPLE                                   FTAN1179
        00150   604464436331      10000
        00151   474325606060      10000
        00152   002145621333      10000   MESD   BCI     2,0ANS=.17E39                                       FTAN117A
        00153   010725031160      10000
                                          *                                                                   FTAN1180
                                          * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN1190
```

```
          FTAN                                                                                  07/31/65
          IBJOB LIBRARY / TANGENT FUNCTION

                                          *
     00154  204540000000      10000       A6     DEC      11.,9.,7.,5.,3.                                  FTAN120
     00155  204440000000      10000                                                                        FTAN121
     00156  203700000000      10000
     00157  203500000000      10000

BINARY CARD (NOT PUNCHED)
     00160  202600000000      10000
     00161  201400000000      10000       A1     DEC      1.                                               FTAN122
                                          *                                                                 FTAN123
                                          * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN124
                                          *                                                                 FTAN125
     00162  000000000000      10000       LK.DP  LDIR                                                      FTAN126
     00163  266321456060      10000
     00164  234622077325      10000              LORG                                                      FTAN127
     00165  202622077325      10000
     00166  163643334273      10000
     00167  233400000000      10000
     00170  177505746033      10000
     00171  200000000000      10000
     00172  201622077325      10000
     00173  200622077325      10000
                                          *                                                                 FTAN128
                                          * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN129
                                          *                                                                 FTAN130
                                           FRASE CONTRL  ER                                                 FTAN131
     00174  100000000174      00001              USE    ER                                                  FTAN132
     00174  200000000003      00001       COMMON BSS    3                                                   FTAN133
     00177  200000000021      00001              BSS    17                                                  FTAN134
                                          *                                                                 FTAN135
                                          * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN136
                                                                                                            FTAN137
                        00000  01111              END                                                       FTAN138
```

```
                                                                                                3FA00030
                                                CALLING SEQUENCE IS CALL .FVIO.(LN,ERAS) WHERE LN IS    3FA00040
                                                LOCATION OF VARIABLE LOGICAL UNIT AND ERAS WILL CONTAIN 3FA00050
                                                CONTENTS OF APPROPRIATE .UNXX.  (.UNXX. CONTAINS        3FA00060
                                                PZE UNITXX WHERE XX CORRESPONDS TO LOGICAL UNIT N).     3FA00070
                                                FVIO IS CALLED FOR ANY I/O STATEMENT SPECIFYING         3FA00080
                                                A VARIABLE LOGICAL UNIT.                                3FA00090
                                                                                                        3FA00100

BINARY CARD (NOT PUNCHED)
      00000  1 00000 0 00004  10001  .FVIO. SAVE    (2)                                          (1)3FA00110
      00001  0774 00 2 00000  10000
      00002  0774 00 4 00000  10000
      00003  0020 00 4 00001  10000
      00004  0634 00 4 04000  10011
      00005  0634 00 4 00102  10001
      00006  0634 00 4 00002  10001
      00007  0634 00 2 00001  10001
      00010  0500 60 4 00003  10000         CLA*    3,4                 PICK UP LOGICAL UNIT NUMBER  (1)3FA00120
      00011  0737 00 2 00000  10000         PAC     ,2                                           (1)3FA00130
      00012  7 00000 2 00016  11101         TXL     ERROR,2,-NUNITS-1   IS UNIT ZERO, OR TOO LARGE  (1)3FA00140
             1 00070 0 00102  10101
             1 00001 0 00000  10011
             1 00000 0 00000  11100
             1 00001 7 00000  10011
      00013  0500 60 2 00070  10001         CLA*    TABLE,2             SAVE ADDRESS OF              (1)3FA00150
      00014  0601 60 4 00004  10000         STO*    4,4                 FILE CONTROL BLOCK           (1)3FA00160
                     00015                  RETURN  .FVIO.                                           (1)3FA00170
      00016  0500 60 4 00003  10000 ERROR   CLA*    3,4                 LOGICAL UNIT IN ERROR        (1)3FA00250

BINARY CARD (NOT PUNCHED)
      00017  4320 00 0 00064  10001         ANA     ADMSK               DEFINED FOR THIS UNIT VALUE.    3FA00260
      00020  0131 00 C 00000  10000         XCA                         CONVERT THIS ILLEGAL VALUE      3FA00270
      00021  0774 00 4 00000  10000         AXT     0,4                 TO DECIMAL FOR ERROR MESSAGE.   3FA00280
      00022  0600 00 0 00065  10001         STZ     TEMP                                            3FA00290
      00023  0754 00 0 00000  10000 CNVT    PXA     0,0                                             3FA00300
      00024  0221 00 0 00066  10001         DVP     L(10)                                           3FA00310
      00025  0767 00 4 00000  10000         ALS     0,4                                             3FA00320
      00026  4602 00 0 00065  10001         ORS     TEMP                                            3FA00330
      00027  0500 00 0 00104  10001         CLA     =1                                              3FA00340
      00030  0040 00 0 01002  10011         TLQ     *+2                                             3FA00350
      00031  1 77772 4 00023  10001         TXI     CNVT,4,-6                                       3FA00360
      00032  4500 00 C 00067  10001         CAL     BLANKS                                          3FA00370
      00033  0767 00 4 00006  10000         ALS     6,4                                             3FA00380
      00034  4501 00 0 00065  10001         ORA     TEMP                                            3FA00390
      00035  0602 00 0 00054  10001         SLW     E47MES+6                                        3FA00400
      00036  000000000000     00010 UNERR   CALL    .FXEM.(CODE)        EXIT FOR EXECUTION ERROR.       3FA00410
      00036  0074 00 4 03000  10011
      00037  1 00001 0 01003  10011
      00040  0 00102 0 00053  10100

BINARY CARD (NOT PUNCHED)
      00041  0 00000 0 00043  10001
```

```
00042  0020 00 0 05000  10011          TRA     .LXERR                NO OPTIONAL EXIT.                  3FA00420
00043  0 00000 0 00057  10000  CODE    PZE     47                                                       3FA00430
00044  0 00007 0 00046  10001  ER47MS  PZE     E47MES,,7                                                3FA00440
00045  0 00007 0 00055  10001  ER47OP  PZE     NOOPXT,,7                                                3FA00450
00046  004346273123     10000  E47MES  BCI     7,0LOGICAL UNIT NOT DEFINED FOR VALUE                    3FA00460
00047  214360644531     10000
00050  636045466360     10000
00051  242526314525     10000
00052  246026465160     10000
00053  652143642560     10000
00054  606060606060     10000
00055  004546604647     10000  NOOPXT  BCI     7,0NO OPTIONAL EXIT - EXECUTION TERMINATED               3FA00470
00056  633146452143     10000
00057  602567316360     10000
00060  406025672523     10000
00061  646331464560     10000
00062  632551443145     10000
00063  216325246060     10000

BINARY CARD (NOT PUNCHED)
00064  000000077777     10000  ADMSK   OCT     000000077777                                             3FA00480
00065  0 00000 0 00000  10000  TEMP    PZE     **                                                       3FA00490
00066  000000000012     10000  L(10)   DEC     10                                                       3FA00500
00067  606060606060     10000  BLANKS  BCI     1,                                                       3FA00510
00070  0 00000 0 00000  10011  TABLE   PZE     NUNITS                                                   3FA00550
       1 00070 0 00102  10101
       1 00001 7 00000  10011
00071  0 00000 0 06000  10011          PZE     .UN01.                                                   3FA00560
00072  0 00000 0 07000  10011          PZE     .UN02.                                                   3FA00570
00073  0 00000 0 10000  10011          PZE     .UN03.                                                   3FA00580
00074  0 00000 0 11000  10011          PZE     .UN04.                                                   3FA00590
00075  0 00000 0 12000  10011          PZE     .UN05.                                                   3FA00600
00076  0 00000 0 13000  10011          PZE     .UN06.                                                   3FA00610
00077  0 00000 0 14000  10011          PZE     .UN07.                                                   3FA00620
00100  0 00000 0 15000  10011          PZE     .UN08.                                                   3FA00630
                                *       ADDITIONAL UNITS MAY BE INSERTED HERE. FOR EACH UNIT             3FA00640
                                *       INSERTED, A CORRESPONDING ROUTINE MUST BE INSERTED TO            3FA00650
                                *       PRODUCE A $FILE CARD FOR THE ADDITIONAL UNIT.                    3FA00660
00101  0 00000 0 16000  10011          PZE     .UN09.
                  00011         NUNITS  EQU     *-TABLE-1                                                3FA00670
00102  000000000000     10000          *LDIR
00103  266531466060     10000
00104  000000000001     10000          *LORG

BINARY CARD (NOT PUNCHED)
                  00000  01111          END                                                              3FA00680
```

```
      SUBROUTINE GEOMD(DAT,LI,LC,LERR,LTAPE)
C     GEOMETRIC DEFINITION LINK
      DIMENSION BUF(10000),DICT(9),NU(3),LOK(3,9),DAT(2)
     1 ,BAXIS(2),BTITLE(12),WTITLE(12)
      COMMON  /LGEOMD/LGDEF(3,6)
      LOGICAL LGDEF,LGTAPE,LTDUMP
      EQUIVALENCE  (LOK(1,1),LUWPL),(LOK(2,1),MUWPL),(LOK(3,1),NUWPL),
     X(LOK(1,2),LLWPL),(LOK(2,2),MLWPL),(LOK(3,2),NLWPL),
     X(LOK(1,3),LUWPLX),(LOK(2,3),MUWPLX),(LOK(3,3),NUWPLX),
     X(LOK(1,4),LLWPLX),(LOK(2,4),MLWPLX),(LOK(3,4),NLWPLX),
     X(LOK(1,5),LBPL),(LOK(2,5),MBPL),(LOK(3,5),NBPL),
     X(LOK(1,6),LSTA),(LOK(3,6),NSTA),
     X(LOK(1,7),LWPF),(LOK(2,7),MWPF),(LOK(3,7),NWPF),
     X(LOK(1,8),LUWB),(LOK(3,8),NUWB),
     X(LOK(1,9),LLWB),(LOK(3,9),NLWB)
      DATA  DICT/4HBODY,4HBODY,4HWING,4HWING,6HDEFEND,3HWBX,
     1      6HNOTAPE,6HNOTAPE,5HTDUMP/,LGTAPE/T/,
     1      NBUF,NDICT,NU(3),LOK/10000, 9,50,27*0/
C
C     INTWB IS USED BY DEFEND
      INTWB=0                                                           ,1
      LTDUMP=.FALSE.
C
C     THE LGDEF ARRAY (LOGICAL) TELLS THE STATUS OF DEFINITION.
C     LGDEF(I,J) HAS FOLLOWING FORMAT.
C     J=1, UPPER WING.  J=2, LOWER WING.  J=3, BODY.  J=4, WING
C     PLANFORM.  J=5, UPPER WING INTERSECTION.  J=6, LOWER WING.
C     I=1, REQUESTED.  I=2, SUCCESSFULLY DEFINED.
C     I=3, DEFINITION ON TAPE (LTAPE).
C                                                                       ,2
      DO 100 J=1,6                                                      ,3
      DO 100 I=1,3                                                      ,4
  100 LGDEF(I,J)=.FALSE.
C
C     CHECK LTAPE                                                       ,5    ,6    ,7
      IF (LTAPE*(LTAPE-40))430,420,420                                  ,8
  420 LGTAPE=.FALSE.
C                                                                       ,9
  430 NU(2)=LERR
C     SET UP STORAGE BUFFER                                             ,10
      CALL INIBFR(BUF,NBUF)
C
C     READ DATA CARD, SEE IF VALID COMMAND                              ,11
  500 WRITE(LO,501)                                                     ,12   ,13
  501 FORMAT(1H1)
  502 IGO=INTURP(DICT,NDICT,LI,LO)
C
C                                                                       ,14
      IF (IGO)510,510,530
C     NOT VALID                                                         ,15
  510 WRITE (LO,520)                                                    ,16   ,17
  520 FORMAT(1H+  90X,7HCOMMENT)
      GO TO 502
```

```
C
  500 GO TO (800,800,1000,1000,1600,2000,1700,1700,                    ,18
     X 2100),IGO
C
C     BODY
  800 CALL BODY1(DAT,LI,LO,BUF,LSTA,NSTA,LBPL,MBPL,NBPL,LTHETA         ,19
     1 ,BAXIS,BTITLE,LGDEF,NU)
  810 GO TO 500                                                        ,20
C
C     WING
 1000 CALL WINGA(BUF,LI,LO,DAT,LGDEF,LOK,LWPF,WTITLE,NU)               ,21
 1010 CONTINUE                                                         ,22
      GO TO 500                                                        ,23
C
C     DEFEND
 1600 GO TO 2050                                                       ,24
C
C     NOTAPE
 1700 LGTAPE=.FALSE.                                                   ,25
      GO TO 500                                                        ,26
C
C     WBX
 2000 INTWB=-1                                                         ,27
C
 2050 CALL DEFEND(BUF,LOK,LO,LTAPE,LGDEF,LGTAPE,INTWB)                 ,28
      IF(INTWB)2060,2080,2080                                          ,29
C
 2060 INTWB=1                                                          ,30
      CALL WBXUL(BUF,LI,LO,DAT,LGDEF,LOK,BTITLE,WTITLE,BAXIS,NU)       ,31
 2070 GO TO 500                                                        ,32
C
 2080 IF(.NOT. LTDUMP) GO TO 9000                                      ,33
      CALL TDUMP(LO,LTAPE,LGDEF)                                       ,34   ,35   ,36
      GO TO 9000                                                       ,37
C
C     TDUMP
 2100 LTDUMP=.TRUE.                                                    ,38
      GO TO 500                                                        ,39
C
 9000 CALL CLEAR(BUF)                                                  ,40
      RETURN                                                           ,41
      END                                                              ,42
                                                                       ,43
```

```
CIADARY     MICHAEL SYNGE          6-9247       COMPILED  21 AUG 64
      FUNCTION  IADARY(ARRAY,IN,BFR,NARAY)                          IADA0030
      DIMENSION ARRAY(1),BFR(1)                                     IADA0040
C                                                                   IADA0060
C  TO ADD AN ARRAY TO A BUFFER                                      IADA0070
C                                                                   IADA0090
C              I      - THE LENGTH OF THE CURRENT ARRAY             IADA0100
C              M      - FIRST WORD OF CURRENT ARRAY                 IADA0110
C              MM     - THE TOTAL LENGTH OF THE BUFFER              IADA0120
C              N      - THE LENGTH OF THE ARRAY TO BE ADDED         IADA0130
C                                                                   IADA0140
C              SEARCH FOR A FREE ARRAY WHICH IS BIG ENOUGH          IADA0150
C                                                                   IADA0160
      N=IABS(IN)                                                    IADA0170    ,1
      K=0                                                           IADA0180    ,2
      IF(ICODE(BFR)-4)95,1,95                                       IADA0190    ,3
    1 MM=-IBFR(BFR(3))                                              IADA0200    ,4
    2 M=4                                                           IADA0210    ,5
    3 I=LENG(BFR(M-1))                                              IADA0220    ,6
      IF( ICODE(BFR(M-2)))5,4,5                                     IADA0230    ,7
    4 IF(I-N)5,10,10                                                IADA0240    ,8
    5 M=M+I+2                                                       IADA0250    ,9
      IF(M-MM)3,6,6                                                 IADA0260
C                                                                   IADA0270
C              PACK THE BUFFER IF NECESSARY                         IADA0280
C                                                                   IADA0290    ,10
    6 IF(K)90,7,90                                                  IADA0300    ,11
    7 K=1                                                           IADA0310    ,12
      IF(IPACK(BFR))85,85,2                                         IADA0320
C                                                                   IADA0330
C              FREE ARRAY FOUND                                     IADA0340
C              SET UP CODE WORDS                                    IADA0350
C                                                                   IADA0360    ,13
   10 IADARY=M                                                      IADA0370    ,14
      IF(IN)25,25,15                                                IADA0380    ,15
   15 DO 20 K=1,N                                                   IADA0390    ,16
      BFR(M)=ARRAY(K)                                               IADA0400    ,17
 20   M=M+1                                                         IADA0410    ,18    ,19
   25 BFR(IADARY-1)=ARSEC(I,NARAY)                                  IADA0420    ,20
      BFR(IADARY-2)=ARFUL(IADARY)                                   IADA0430    ,21
   30 NARAY=IADARY                                                  IADA0440    ,22
      CALL IRLEAS(BFR(IADARY),N)                                    IADA0450    ,23
 80   RETURN                                                        IADA0460
C                                                                   IADA0470
C              ERRORS                                               IADA0480
C                                                                   IADA0490    ,24
   85 IADARY=-1                                                     IADA0500    ,25
      GO TO 80                                                      IADA0510    ,26
   90 IADARY=-2                                                     IADA0520    ,27
      GO TO 80                                                      IADA0530    ,28
   95 IADARY=-3                                                     IADA0540    ,29
      GO TO 80                                                      IADA0550    ,30
      END                                                           IADA0560    ,31
```

```
IBFR                                                                    07/31/65
7094 RELMOD ASSEMBLY.

        $IBLDR IBFR                                                                    IBFR0000
ASSEMBLED TEXT.

        $TEXT  IBFR                                                                    IBFR0001

                                *IBFR       MICHAEL SYNGE          6-9247      ASSEMBLED  21 AUG 64
                                *
                                *  TO LOCATE THE FIRST CELL OF A BUFFER
                                *  RELATIVE TO THE FIRST CELL OF AN ARRAY
                                *
                                *              CALLING SEQUENCE
                                *
                                *                      M=IBFR(ARRAY)
                                *
                                *              NOTE - M IS ALWAYS NEGATIVE
                                *              UNLESS THE GIVEN ARGUMENT IS
                                *              NOT THE FIRST WORD OF AN ARRAY.
                                *
                                       ENTRY   IBFR
                                *

BINARY CARD (NOT PUNCHED)
      00000  0500 00 4 00003  10000   IBFR  CLA     3,4
      00001  0402 00 0 00017  10001         SUB     =2
      00002  0621 00 0 00003  10001         STA     A
      00003  4500 00 0 00000  10000   A     CAL     **
      00004  0621 00 0 00016  10001         STA     B
      00005  4320 00 0 00020  10001         ANA     =077777700000
      00006  4340 00 0 00021  10001         LAS     =0700000
      00007  0020 00 0 00011  10001         TRA     ERROR
      00010  0020 00 0 00013  10001         TRA     SUCCES
      00011  0500 00 0 00022  10001   ERROR CLA     =1
      00012  0020 00 0 00015  10001         TRA     EXIT
      00013  0500 00 0 00016  10001  SUCCES CLA     B
      00014  4760 00 0 00003  10000         SSM
      00015  0020 00 4 00001  10000   EXIT  TRA     1,4
                                *
      00016  0 00000 0 00000  10000   B
      00017  000000000002     10000         *LORG
      00020  077777700000     10000
      00021  000000700000     10000
      00022  000000000001     10000

BINARY CARD (NOT PUNCHED)
                       00000  01111         END
```

```
          IBFR                                                                  07/31/65
        CONTROL DICTIONARY

                $CDICT IBFR                                                                  IBFR0002

BINARY CARD (NOT PUNCHED)
          000023000000                PREFACE     START=0,LENGTH=19,TYPE=7094,CMPLX=5
          000004000005
          312226516060        IBFR    DECK        LOC=0,LENGTH=19
          000023000000
          312226516060        IBFR    REAL        LOC=0,LENGTH=0
          000000000000

                $DKEND IBFR                                                                  IBFR0003


NO MESSAGES FOR THIS ASSEMBLY
        SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

  CLASS  SYMBOL   VALUE   REFERENCES

         A        00003   2
         B        00016   4,13
         ERROR    00011   7
         EXIT     00015   12
         IBFR     00000
  LCTR    BLCTR
  QUAL    UNQS
  LCTR   //
         SUCCES   00013   10
```

```
ICODE                                                                  07/31/65
7094 RELMOD ASSEMBLY.

        $IBLDR ICODE                                                                  ICOD0000
ASSEMBLED TEXT.

        $TEXT  ICODE                                                                  ICOD0001

                                *ICODE      MICHAEL SYNGE          6-9247      ASSEMBLED  21 AUG 64
                                *              TO VERIFY THAT THE GIVEN WORD
                                *              IS INDEED THE FIRST CODE OF AN
                                *              ARRAY AND TO CONVERT ITS PREFIX
                                *              INTO AN ADDRESS  INTEGER. THE
                                *              RESULT IS  -1  IF THE WORD IS
                                *              NOT THE EXPECTED CODE.
                                *
                                *                      CALLING SEQUENCE
                                *
                                *                            I=ICODE(A)
                                *
                                        ENTRY   ICODE
                                *

BINARY CARD (NOT PUNCHED)
      00000  0560 60 4 00003  10000   ICODE LDQ*    3,4
      00001  0754 00 0 00000  10000         PXA     0,0
      00002  4763 00 0 00003  10000         LGL     3
      00003  0600 00 0 00016  10001         STZ     TEMP
      00004  0621 00 0 00016  10001         STA     TEMP
      00005  4130 00 0 00000  10000         XCL
      00006  4320 00 0 00017  10001         ANA     =0777777000000
      00007  4340 00 0 00020  10001         LAS     =07000000
      00010  0020 00 0 01002  10011         TRA     *+2
      00011  0020 00 0 00014  10001         TRA     OK
      00012  0500 00 0 00021  10001         CLA     =-1
      00013  0020 00 0 00015  10001         TRA     EXIT
      00014  0500 00 0 00016  10001   OK    CLA     TEMP
      00015  0020 00 4 00001  10000   EXIT  TRA     1,4
                                *
      00016  0 00000 0 00000  10000   TEMP
      00017  777777000000     10000         *LORG
      00020  000007000000     10000
      00021  400000000001     10000
                     00000  01111         END
```

```
          ICODE                                                                07/31/65
        CONTROL DICTIONARY

                $CDICT ICODE                                                                  ICOD0002

BINARY CARD (NOT PUNCHED)
            000022000000              PREFACE     START=0,LENGTH=18,TYPE=7094,CMPLX=5
            000004000005
            312346242560      ICODE   DECK        LOC=0,LENGTH=18
            000022000000
            312346242560      ICODE   REAL        LOC=0,LENGTH=0
            000000000000

                $DKEND ICODE                                                                  ICOD0003

NO MESSAGES FOR THIS ASSEMBLY
          SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

  CLASS  SYMBOL   VALUE   REFERENCES

         EXIT     00015   13
         ICODE    00000
         OK       00014   11
  LCTR    BLCTR
  QUAL    UNQS
  LCTR   //
         TEMP     00016   3,4,14
```

```
CIDLETE     MICHAEL SYNGE          6-9247       COMPILED  21 AUG 64
C  TO DELETE AN ARRAY FROM A BUFFER                                    IDLE0030
C                                                                      IDLE0050
      FUNCTION IDLETE(ARRAY)                                           IDLE0060
C                                                                      IDLE0070
      DIMENSION ARRAY(1)                                               IDLE0080
C                                                                      IDLE0100
      M=-IBFR(ARRAY)                                                   IDLE0110    ,1
      IF (M)90,90,10                                                   IDLE0120    ,2
   10 I=32767                                                          IDLE0130    ,3
      ARRAY(I)=ARFRE(M)                                                IDLE0140    ,4
      IDLETE=0                                                         IDLE0150    ,5
      GO TO 100                                                        IDLE0160    ,6
   90 IDLETE=1                                                         IDLE0170    ,7
  100 RETURN                                                           IDLE0180
C                                                                      IDLE0190
C                                                                      IDLE0200    ,8
      END                                                              IDLE0210    ,9
```

```
          INIBFR                                                                07/31/65
        7C94 RELMOD ASSEMBLY.

                $IBLDR INIBFR                                                                  INIB0000
        ASSEMBLED TEXT.

                $TEXT  INIBFR                                                                  INIB0001
                                        *INIBFR     MICHAEL SYNGE          6-9247      ASSEMBLED  21 SEP 64
                                        *
                                        *
                                        *  TO INITIALIZE ANY NUMBER OF STORAGE BUFFERS
                                        *
                                        *     CALLING SEQUENCE -
                                        *
                                        *              CALL INIBFR(BUFA,NBUFA,BUFB,NBUFB,.......)
                                        *
                                              ENTRY   INIBFR
                                        *

BINARY CARD (NOT PUNCHED)
      00000  0634 00 4 00004  10001  INIBFR SXA     EXIT,4
      00001  0500 00 4 00003  10000         CLA     3,4
      00002  4320 00 0 00026  10001         ANA     =0777777700000
      00003  0100 00 0 00006  10001         TZE     NEXT
      00004  0774 00 4 00000  10000   EXIT  AXT     0,4
      00005  0020 00 4 00001  10000         TRA     1,4
                                        *
                                        *
      00006  0500 00 4 00003  10000   NEXT  CLA     3,4
      00007  0400 00 0 00027  10001         ADD     =1
      00010  0621 00 0 00017  10001         STA     ARR1
      00011  0400 00 0 00027  10001   ODD   ADD     =1
      00012  0621 00 0 00024  10001         STA     ARR2
      00013  0500 60 4 00004  10000         CLA*    4,4             BUFFER LENGTH
      00014  4501 00 0 00030  10001         ORA     =0400000700000
      00015  0602 60 4 00003  10000   LENG  SLW*    3,4             FIRST CODE WORD OF BUFFER
      00016  0500 00 0 00031  10001         CLA     =0700004
      00017  0601 00 0 00000  10000   ARR1  STO     **
      00020  0500 60 4 00004  10000         CLA*    4,4
      00021  0402 00 0 00032  10001         SUB     =3
      00022  0767 00 0 00022  10000         ALS     18

BINARY CARD (NOT PUNCHED)
      00023  0600 60 0 00024  10001         STZ*    ARR2
      00024  0622 00 0 00000  10000   ARR2  STO     **              SECOND CODE WORD OF FIRST ARRAY
      00025  1 77776 4 02001  10011         TXI     INIBFR+1,4,-2
                                        *
      00026  777777700000     10000         *LORG
      00027  000000000001     10000
      00030  400000700000     10000
      00031  000000700004     10000
      00032  000000000003     10000
                       00000  01111         END
```

```
          INIBFR                                                                07/31/65
        CONTROL DICTIONARY

                $CDICT INIBFR                                                                  INIB0002

BINARY CARD (NOT PUNCHED)
            000033000000              PREFACE     START=0,LENGTH=27.TYPE=7094,CMPLX=5
            000004000005
            314531222651       INIBFR DECK        LOC=0,LENGTH=27
            000033000000
            314531222651       INIBFR REAL        LOC=0,LENGTH=0
            000000000000

                $DKEND INIBFR                                                                  INIB0003

NO MESSAGES FOR THIS ASSEMBLY
          SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

  CLASS  SYMBOL   VALUE   REFERENCES

         ARR1     00017   10
         ARR2     00024   12,23
         EXIT     00004   0
         INIBFR   00000   25
         LENG     00015
         NEXT     00006   3
         ODD      00011
  LCTR    BLCTR
  QUAL    UNQS
  LCTR   //
```

```
      SUBROUTINE INCUT (NTAPEO,KASE,CPCALC,POLAR,THICK,VOUT,REFAREA,XP,ZP
     1,XMACH,SYM)
C     PRINTS OUT ALL INPUT AERODYNAMIC DATA

      WRITE (NTAPEO,6010)                                                ,1    ,2
      IF (SYM) 110,100,110                                               ,3
  100 WRITE (NTAPEO,6015)                                                ,4    ,5
      GO TO 120                                                          ,6
  110 WRITE (NTAPEO,6020)                                                ,7    ,8
  120 CONTINUE
                                                                         ,9
      GO TO (130,140,150),KASE                                            10
  130 WRITE (NTAPEO,6025)                                                ,11   ,12
      GO TO 160                                                          ,13
  140 WRITE (NTAPEO,6030)                                                ,14   ,15
      GO TO 160                                                          ,16
  150 WRITE (NTAPEO,6035)                                                ,17   ,18
  160 CONTINUE
                                                                         ,19
      IF (CPCALC) 180,170,180                                            ,20
  170 WRITE (NTAPEO,6040)                                                ,21   ,22
      GO TO 190                                                          ,23
  180 WRITE (NTAPEO,6045)                                                ,24   ,25
  190 CONTINUE
                                                                         ,26
      IF (POLAR) 210,200,210                                             ,27
  200 WRITE (NTAPEO,6050)                                                ,28   ,29
      GO TO 220                                                          ,30
  210 WRITE (NTAPEO,6055)                                                ,31   ,32
  220 CONTINUE
                                                                         ,33
      IF (THICK) 240,230,240                                             ,34
  230 WRITE (NTAPEO,6060)                                                ,35   ,36
      GO TO 250                                                          ,37
  240 WRITE (NTAPEO,6065)                                                ,38   ,39
  250 CONTINUE
                                                                         ,40
      IF (VOUT) 270,260,270                                              ,41
  260 WRITE (NTAPEO,6070)                                                ,42   ,43
      GO TO 280                                                          ,44
  270 WRITE (NTAPEO,6075)                                                ,45   ,46
  280 CONTINUE
                                                                         ,47
      WRITE (NTAPEO,6085) REFAREA                                        ,48   ,49   ,50
      WRITE (NTAPEO,6090) XP,ZP                                          ,51   ,52   ,53
      WRITE (NTAPEO,6095) XMACH
                                                                         ,54   ,55   ,56
 6010 FORMAT(1H0//30H DESCRIPTION OF CASE REQUESTED)
 6015 FORMAT(1H0/35H UNSYMMETRICAL CONFIGURATION - PANELS LOCATED ONLY O
     1N ONE SIDE OF X-Z PLANE(SYM = 0.))
```

```
6020 FORMAT(1H0/80H SYMMETRICAL CONFIGURATION - PANELS LOCATED ON BOTH
    1SIDES OF X-Z PLANE(SYM = 1.))
6025 FORMAT(1H0,9HCASE = 1.,10X,35HCALCULATE WING SHAPE, GIVEN WING CL)
6030 FORMAT(1H0,9HCASE = 2.,10X,25HCALCULATE CL, GIVEN SHAPE)
6035 FORMAT(1H0,9HCASE = 3.,10X,19HOPTIMIZE WING SHAPE)
6040 FORMAT(1H0,11HCPCALC = 0.,8X,9HLINEAR CP)
6045 FORMAT(1H0,11HCPCALC = 1.,8X,13HNON-LINEAR CP)
6050 FORMAT(1H0,10HPOLAR = 0. ,9X,20HPOLARS NOT REQUESTED)
6055 FORMAT(1H0,10HPOLAR = 1. ,9X,16HPOLARS REQUESTED)
6060 FORMAT(1H0,10HTHICK = 0.,9X,40HWING THICKNESS PRESSURES NOT TO BE
    1ADDED)
6065 FORMAT(1H0,10HTHICK = 1.,9X,36HWING THICKNESS PRESSURES TO BE ADDE
    1D)
6070 FORMAT(1H0,9HVOUT = 0.,10X,37HVELOCITY COMPONENTS NOT TO BE PRINTE
    1D)
6075 FORMAT(1H0,9HVOUT = 1.,10X,33HVELOCITY COMPONENTS TO BE PRINTED)
6085 FORMAT(1H0,21HWING REFERENCE AREA =,F10.4)
6090 FORMAT(1H0,48HPOINT ABOUT WHICH THE MOMENTS ARE TO BE COMPUTED/16H
    1  X-COORDINATE =,F10.4/16H  Z-COORDINATE =,F10.4)
6095 FORMAT(1H0,13HMACH NUMBER =,F10.4)

     RETURN                                                          ,57
     END                                                             ,58
```

```
      SUBROUTINE INPUTB
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KOCEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/ NPLNB,NPLNW,JLEAD,JTRAIL,IMID,NPTS(16),X(16,90),Y(16
     1,90),Z(16,90),XCEPT(21),XCEPTB(21),XCEPTW(16),YCEPTW(16),ZCEPTW(16
     2),CODEBW(16),KPANEL(15,20),XCOR(16,21),YCOR(16,21),ZCOR(16,21),XIN
     3T(15,20,2),YINT(15,20,2),ZINT(15,20,2),XCEN(15,20),YCEN(15,20),ZCE
     4N(15,20),XCON(15,20),YCON(15,20),ZCON(15,20),AREA(15,20),ARAT(15,2
     50),THETA(15,20),ALPHA(15,20),CHORD(15,20)
      DIMENSION REC(5000),LREC(10)

C     SUBROUTINE TO INPUT DATA FOR BODY PANELING
C     SUBROUTINES
      ND4=ND2(1)

C     READ CARD INPUT - PLANB,PLANW ARE CODES FOR PLANES
C     THAT INTERSECT BODY      PLANB IS NUMBER OF CUTTING
C     PLANES TO INTERSECT FORE AND AFT OF BODY-WING
C     INTERSECTION REGION      PLANW IS NUMBER OF CUTTING
C     PLANES TO INTERSECT IN THE BODY-WING INTERSECTION
C     REGION      PLANE IS TOTAL NUMBER OF PLANES
C     TO INTERSECT BODY AND DEFINE PANEL LEADING AND
C     TRAILING EDGES      TOLB IS TOLERANCE USED
C     IN PROGRAM CHECK FOR SMALL SUBPANELS      IF SLOPE
C     OF SUBPANEL EDGE FROM STREAMWISE IS LESS THAN
C     TOLERANCE, SUBPANEL IS OMITTED FROM OUTPUT                          ,1
   10 READ (NREAD,900) PLANB,PLANW,TOLB                                   ,2    ,3    ,4
      WRITE (NWRITE,950)

C     DEFINE CONSTANTS                                                    ,5    ,6
      NPLNB=PLANB                                                         ,7
      IF (NPLNB .LT. 0 .OR. NPLNB .GT. 21) GO TO 298                      ,8    ,9    ,10
      NPLNW=PLANW                                                         ,11
      IF (NPLNW .LT. 0 .OR. NPLNW .GT. 16) GO TO 298                      ,12   ,13   ,14
      NPLANE=NPLNB+NPLNW                                                  ,15
      IF (NPLANE .LT. 2 .OR. NPLANE .GT. 21) GO TO 298                    ,16   ,17   ,18
      NPLN1=NPLANE-1                                                      ,19
      KOPTB=0                                                             ,20
      SYM=TOLB                                                            ,21
      IF (KOPTB) 300,15,200                                               ,22
   15 IF (NPLANE) 300,300,20                                              ,23
   20 IF (NPLNB) 300,35,25

C     READ CARD INPUT - X-AXIS INTERCEPT VALUES OF
C     BODY CUTTING PLANES TO INTERSECT FORE AND AFT OF
C     BODY-WING INTERSECTION REGION                                       ,24
   25 READ (NREAD,900) (XCEPTB(I),I=1,NPLNB)                              ,25   ,26   ,27   ,28   ,29
      READ (ND1) (REC(I),I=1,6)                                           ,30   ,31   ,32   ,33   ,34
      A=REC(5)                                                            ,35
      B=REC(6)
```

```
C     CHANGE X-AXIS INTERCEPT VALUES TO ALLOW FOR
C     BODY CAMBER                                                    ,36
      DO 30 I=1,NPLNB                                                ,37
 30   XCEPTB(I)=A*XCEPTB(I)+B                                        ,38   ,39
      CALL CHECK (XCEPTB,NPLNB,2,L,NWRITE)                           ,40
      IF (L) 298,35,298

C     READ TAPE INPUT - BODY-WING INTERSECTION                       ,41
 35   NREC=4                                                         ,42
      KCHK=NREC+1                                                    ,43
      CALL FSR (NREC,ND4,LERROR)                                     ,44
      READ (ND4) (LREC(J),J=1,10)                                    ,45   ,46   ,47   ,48   ,49
      KREC=LREC(1)                                                   ,50
      IF (KREC-KCHK) 293,40,293                                      ,51
 40   KCODE=LREC(3)

C     DEFINE KINT      IF KINT = 0, BODY-WING INTER-
C     SECTION HAS NOT BE REQUESTED OR CALCULATED
C     IF KINT = 1, OTHERWISE                                         ,52
      IF (KCODE) 293,44,43                                           ,53
 43   KINT=0                                                         ,54
      GO TO 115                                                      ,55
 44   KINT=1                                                         ,56
      NWORD=LREC(2)                                                  ,57
      IF (NWORD-5000) 45,45,294                                      ,58
 45   NPER=LREC(4)                                                   ,59
      IF (NPER-16) 46,46,294

C     READ X-, Y- AND Z-COORDINATES OF
C     BODY-WING INTERSECTION                                         ,60
 46   READ (ND4) (REC(J),J=1,NWORD)                                  ,61   ,62   ,63   ,64   ,65
      DO 47 J=1,NPER                                                 ,66
      XI(J)=REC(3*J-2)                                               ,67
      YI(J)=REC(3*J-1)                                               ,68
 47   ZI(J)=REC(3*J)                                                 ,69   ,70
 50   IF (NPLNW) 300,115,53

C     READ CARD INPUT - CODES INDICATING WING PERCENT
C     LINES THAT ARE TO DEFINE BODY-CUTTING PLANES TO
C     INTERSECT IN BODY-WING INTERSECTION REGION

C     DEFINE CONSTANT NLOOP TO BE NUMBER OF CARDS
C     TO BE READ                                                     ,71
   53 NLOOP=((NPLNW-1)/6)+1                                          ,72
      DO 57 J=1,NLOOP                                                ,73
      JSTART=1+(J-1)*6                                               ,74
      JEND=6+(J-1)*6                                                 ,75
      READ (NREAD,900) (CODEBW(K),K=JSTART,JEND)                     ,76   ,77   ,78   ,79   ,80
      IF (J-1) 300,55,57

C     CHECK FOR CONSTANT OF -1 IN FIRST FIELD OF
C     FIRST CARD                                                     ,81
 55   IF (CODEBW(1)) 65,300,57                                       ,82
```

```
 57   CONTINUE

C     DEFINE X-AXIS INTERCEPTS OF BODY CUTTING PLANES
C     THAT INTERSECT IN BODY-WING INTERSECTION REGION                    ,83   ,84
 60   DO 63 J=1,NPLNW                                                    ,85
      K=CODEBW(J)+.001                                                   ,86
      XCEPTW(J)=XI(K)                                                    ,87
      YCEPTW(J)=YI(K)                                                    ,88
 63   ZCEPTW(J)=ZI(K)                                                    ,89   ,90
      GO TO 70                                                           ,91
 65   DO 67 J=1,NPLNW                                                    ,92
      XCEPTW(J)=XI(J)                                                    ,93
      YCEPTW(J)=YI(J)                                                    ,94
 67   ZCEPTW(J)=ZI(J)

C     REDEFINE BODY-CUTTING PLANES                                       ,95   ,96
 70   IF (NPLNB) 300,105,75                                              ,97
 75   IF (NPLNW) 300,115,80                                              ,98
 80   JLEAD=0                                                            ,99
      DO 100 J=1,NPLANE                                                  ,100
      IF (XI(1)-XCEPTB(J)) 90,297,85                                     ,101
 85   XCEPT(J)=XCEPTB(J)                                                 ,102
      JLEAD=J                                                            ,103
      GO TO 100                                                          ,104
 90   DO 93 K=1,NPLNW                                                    ,105
      L=JLEAD+K                                                          ,106
 93   XCEPT(L)=XCEPTW(K)                                                 ,107  ,108
      IF (XI(NPER)-XCEPTB(JLEAD+1)) 95,297,297                           ,109
 95   JTRAIL=JLEAD+NPLNW                                                 ,110
      JLEFT=NPLANE-JTRAIL                                                ,111
      DO 97 K=1,JLEFT                                                    ,112
      L=JTRAIL+K                                                         ,113
      M=JLEAD+K                                                          ,114
 97   XCEPT(L)=XCEPTB(M)                                                 ,115  ,116
      GO TO 130                                                          ,117
 100  CONTINUE                                                           ,118  ,119
      GO TO 130                                                          ,120
 105  IF (NPLANE-NPLNW) 300,107,300                                      ,121
 107  DO 110 J=1,NPLNW                                                   ,122
 110  XCEPT(J)=XCEPTW(J)                                                 ,123  ,124
      GO TO 130                                                          ,125
 115  IF (NPLANE-NPLNB) 300,117,300                                      ,126
 117  DO 120 J=1,NPLNB                                                   ,127
 120  XCEPT(J)=XCEPTB(J)                                                 ,128  ,129
 130  REWIND ND4

C     WRITE TAPE OUTPUT - X-PRIME INTERCEPTS OF
C     BODY CUTTING PLANES                                                ,130
      WRITE (NWRITE,960) NPLANE,(XCEPT(J),J=1,NPLANE)

C     READ TAPE INPUT - NPER IS NUMBER
C     OF DEFINING BODY PERCENT LINES

C     READ X-, Y- AND Z-COORDINATES OF POINTS ON
```

```
C     BODY MERIDIAN LINES                                                ,131  ,132  ,133  ,134  ,135  ,136

 160  READ (ND4) (LREC(I),I=1,10)                                        ,137  ,138  ,139  ,140  ,141
      KREC=LREC(1)                                                       ,142
      IF (KREC-1) 290,163,290                                            ,143
 163  KCODE=LREC(3)                                                      ,144
      IF (KCODE) 290,164,290                                             ,145
 164  NWORD=LREC(2)                                                      ,146
      IF (NWORD-5000) 167,167,294                                        ,147
 167  NPER=LREC(4)                                                       ,148
      IF (NPER-16) 168,168,294                                           ,149
 168  NPER1=NPER-1

C     READ TAPE INPUT - NPTS IS NUMBER OF POINTS
C     ON BODY PERCENT LINES                                              ,150
      READ (ND4) (REC(I),I=1,NWORD)                                      ,151  ,152  ,153  ,154  ,155
      K=-2                                                               ,156
      DO 170 I=1,NPER                                                    ,157
      K=K+3                                                              ,158
 170  NPTS(I)=REC(K+1)                                                   ,159  ,160
      DO 175 I=1,NPER                                                    ,161
      IF (NPTS(I)-90) 175,175,294                                        ,162
 175  CONTINUE                                                           ,163  ,164
      DO 180 I=1,NPER                                                    ,165
      NSTOP=NPTS(I)                                                      ,166
      DO 180 J=1,NSTOP                                                   ,167
      K=K+3                                                              ,168
      X(I,J)=REC(K)                                                      ,169
      Y(I,J)=REC(K+1)                                                    ,170
 180  Z(I,J)=REC(K+2)                                                    ,171  ,172  ,173
      REWIND ND4                                                         ,174
      GO TO 300

C     READ CARD INPUT - X-, Y- AND Z-COORDINATES OF
C     BODY PANEL CORNER POINTS                                           ,175
 200  NPER=COLB+1.                                                       ,176
      NPER1=NPER-1                                                       ,177
      DO 210 I=1,NPER1                                                   ,178
      DO 210 J=1,NPLN1                                                   ,179
 210  READ (NREAD,900) XCOR(I,J),XCOR(I+1,J),XCOR(I,J+1),
     1XCOR(I+1,J+1),YCOR(I,J),YCOR(I+1,J),YCOR(I,J+1),YCOR(I+1,J+1),
     2ZCOR(I,J),ZCOR(I+1,J),ZCOR(I+1,J),ZCOR(I+1,J+1)                    ,180  ,181  ,182  ,183  ,184
      WRITE (NWRITE,940)                                                 ,185  ,186
      GO TO 300

C     ERROR MESSAGES                                                     ,187
 290  KODEB=1                                                            ,188
      GO TO 300                                                          ,189
 293  KODEI=1                                                            ,190
      GO TO 300                                                          ,191
  294 WRITE (NWRITE,970)                                                 ,192  ,193
      GO TO 290                                                          ,194
 297  WRITE (NWRITE,990)                                                 ,195  ,196
      GO TO 290                                                          ,197
```

```
298  WRITE (NWRITE,980)                                                    ,198  ,199
     GO TO 290                                                             ,200
300  RETURN                                                                ,201
900  FORMAT (6F10.0)
940  FORMAT (1H1,9X,43HBODY PANELS HAVE BEEN DEFINED ON CARD INPUT)
950  FORMAT (1H1,9X,18HBODY PANEL ROUTINE)
960  FORMAT (1H0,9X,10HTHERE ARE ,I3,51H TRANSVERSE VERTICAL PLANES THA
    1T INTERSECT THE BODY/10X,42HTO DEFINE PANEL LEADING AND TRAILING E
    2DGES//10X,12HX-INTERCEPTS//(F20.5))
965  FORMAT (1H0,9X,43HBODY PANELS HAVE BEEN DEFINED ON CARD INPUT)
970  FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE INPUTB/10X,
    151HIMPROPER BODY DEFINITION OR INTERSECTION DEFINITION/10X,
    250HDIMENSION EXCEEDED FOR NPTS, X, Y, Z OR REC ARRAYS)
980  FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE INPUTB/10X,
    121HINPUT CARD DATA ERROR)
990  FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE INPUTB/10X
    121HINPUT CARD DATA ERROR/10X
    1104HCUTTING PLANE THAT IS DEFINED BY X-AXIS INTERCEPT VALUE INTERS
    2ECTS BODY IN BODY-WING INTERSECTION REGION)
      END                                                                  ,202
```

```
      SUBROUTINE INPUTW
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/IJ,NPTS(16),X(90,16),Y(90,16),Z(90,16),KPNT,VALUE(5),
     1YCEPT(16),SLOPE,KPANEL(2,15),XCOR(16,16),YCOR(16,16),ZCOR(16,16),X
     2INT(2,15),YINT(2,15),ZINT(2,15),XCEN(15,15),YCEN(15,15),ZCEN(15,15
     3),XCON(15,15),YCON(15,15),ZCON(15,15),AREA(15,15),ARAT(15,15),THET
     4A(15,15),ALPHA(15,15),CHORD(15,15),XFOIL(16,25,2),ZFOIL(16,25,2),X
     5NUM(16,2),XTAB1(25),ZTAB1(25),XTAB2(25),ZTAB2(25)
      DIMENSION REC(5000),LREC(10)

C     SUBROUTINE TO INPUT DATA FOR WING PANELING
C     SUBROUTINES

 10   ND4=ND2(1)                                                          ,1
      REWIND ND4                                                          ,2
      IF (IJ-1) 20,190,300

C     READ CARD INPUT - PLANE IS NUMBER OF PLANES
C     TO INTERSECT WING AND DEFINE WING PANEL
C     INBOARD AND OUTBOARD EDGES      OPTF IS CODE FOR CARD
C     INPUT OPTION      IF OPTF = 1, UPPER AND LOWER
C     SURFACE AIRFOIL COORDINATES ARE READ FROM
C     ADDITIONAL INPUT CARDS      IF OPTF = 0, OTHERWISE
C     SNUM IS NUMBER OF AIRFOIL SECTION COORDINATE
C     TABLES TO BE READ      TOLW IS TOLERANCE USED IN
C     PROGRAM CHECK FOR SMALL SUBPANELS      IF SLOPE
C     OF SUBPANEL EDGE FROM STREAMWISE IS LESS THAN
C     TOLERANCE, SUBPANEL IS OMITTED FROM OUTPUT
C     NOTE THAT ALL PANELS OF MAIN WING REGION
C     ARE ONE PART PANELS AND NO CHECK IS NECESSARY
C     HOWEVER, FOR PANELS ON INBOARD EDGE OF WING
C     IN BODY-WING CASE WITH NON-STREAMWISE INTER-
C     SECTION LINE AND FOR PANELS ON NON-STREAM-
C     WISE OUTBOARD EDGE SUBPANEL IS CHECKED AGAINST
C     TOLERANCE                                                           ,3
   20 READ (NREAD,900) PLANE,OPTF,SNUM,TOLW

C     DEFINE CONSTANTS                                                    ,4    ,5    ,6
      NPLANE=PLANE                                                        ,7
      IF (NPLANE .LT. 1 .OR. NPLANE .GT. 16) GO TO 295                    ,8    ,9    ,10
      NPLN1=NPLANE-1                                                      ,11
      KOPTW=0                                                             ,12
      KOPTF=OPTF                                                          ,13
      NUMS=SNUM                                                           ,14
      KACE=0                                                              ,15
      SYM=TOLW                                                            ,16
      IF (KOPTW) 300,25,250                                               ,17
 25   IF (NPLANE) 300,300,30                                              ,18
   30 NREC=2
```

```
C     READ CARD INPUT - Y-AXIS INTERCEPT VALUES FOR
C     PLANES THAT INTERSECT WING     NOTE THAT ALL BUT
C     LAST (OR OUTBOARD) CUTTING PLANE MUST BE
C     STREAMWISE AND Y-AXIS INTERCEPTS ARE SAME
C     AS Y-COORDINATES OF CUTTING PLANE AND LEADING
C     EDGE INTERSECTION     LAST CUTTING PLANE HAS, IN
C     GENERAL, AN ARBITRARY SLOPE AND IS DEFINED
C     BY CODE AND ARRAY OF VALUES     CPNT IS CODE
C     IF CPNT = 0, X- AND Y-COORDINATES OF LEADING
C     AND TRAILING EDGE INTERSECTIONS WITH CUTTING
C     PLANE ARE GIVEN     IF CPNT = 1, LEADING EDGE
C     INTERCEPTS AND SLOPE ARE GIVEN     IF CPNT =
C     2, TRAILING INTERCEPTS AND SLOPE ARE GIVEN                    ,19
 70   READ (NREAD,900) (YCEPT(I),I=1,NPLN1)                         ,20  ,21  ,22  ,23  ,24
      READ (NREAD,900) CPNT,(VALUE(I),I=1,5)                        ,25  ,26  ,27  ,28  ,29  ,30

      CALL CHECK (YCEPT,NPLN1,2,L,NWRITE)                           ,31
      IF (L) 295,75,295                                             ,32
   75 KPNT=CPNT

C     CALCULATE OUTBOARD CUTTING PLANE SLOPE IF
C     NOT GIVEN AS INPUT                                            ,33
      IF (KPNT-1) 80,90,90                                          ,34
 80   IF (ABS(VALUE(4)-VALUE(2))-.001) 83,83,87                     ,35
 83   SLOPE=0.                                                      ,36
      GO TO 100                                                     ,37
 87   SLOPE=(VALUE(3)-VALUE(1))/(VALUE(4)-VALUE(2))                 ,38
      GO TO 100                                                     ,39
 90   SLOPE=VALUE(5)

C     WRITE TAPE OUTPUT - INPUT CARD DATA                           ,40
 100  WRITE (NWRITE,910)                                            ,41  ,42
      WRITE (NWRITE,920) NPLANE                                     ,43  ,44  ,45
      WRITE (NWRITE,930) (YCEPT(I),I=1,NPLN1)                       ,46  ,47  ,48  ,49  ,50
      IF (KPNT-1) 115,115,120                                       ,51
 115  WRITE (NWRITE,935) VALUE(1),VALUE(2),SLOPE                    ,52  ,53  ,54
      GO TO 125                                                     ,55
 120  WRITE (NWRITE,940) VALUE(3),VALUE(4),SLOPE                    ,56  ,57  ,58
      GO TO 125                                                     ,59
 125  WRITE (NWRITE,950)

C     READ TAPE INPUT - NPTS IN NUMBER OF POINTS
C     ON ARBITRARY PERCENT LINE                                     ,60  ,61
 150  CALL FSR (NREC,ND4,LERROR)                                    ,62
      KCHECK=NREC+1                                                 ,63
      READ (ND4) (LREC(I),I=1,10)                                   ,64  ,65  ,66  ,67  ,68
      KREC=LREC(1)                                                  ,69
      IF (KREC-KCHECK) 280,160,280                                  ,70
 160  KCODE=LREC(3)                                                 ,71
      IF (KCODE) 280,163,280                                        ,72
 163  NWORD=LREC(2)                                                 ,73
      IF (NWORD-5000) 165,165,287                                   ,74
 165  NPER=LREC(4)                                                  ,75
```

```
      IF (NPER-16) 167,167,287                                  ,76
  167 NPER1=NPER-1

C     READ X-, Y- AND Z-COORDINATES OF POINTS ON
C     WING PERCENT CHORD LINES                                  ,77
      READ (ND4) (REC(I),I=1,NWORD)                             ,78   ,79   ,80   ,81   ,82
      K=-2                                                      ,83
  175 DO 180 J=1,NPER                                           ,84
      K=K+3                                                     ,85
  180 NPTS(J)=REC(K+1)                                          ,86   ,87
      DO 183 J=1,NPER                                           ,88
      IF (NPTS(J)-90) 183,183,287                               ,89
  183 CONTINUE                                                  ,90   ,91
      DO 185 J=1,NPER                                           ,92
      NSTOP=NPTS(J)                                             ,93
      DO 185 I=1,NSTOP                                          ,94
      K=K+3                                                     ,95
      X(I,J)=REC(K)                                             ,96
      Y(I,J)=REC(K+1)                                           ,97
  185 Z(I,J)=REC(K+2)                                           ,98   ,99   ,100
      REWIND ND4                                                ,101
      GO TO 300

C     READ CARD INPUT - UPPER AND LOWER SURFACE
C     AIRFOIL COORDINATES                                       ,102
  190 IF (KOPTF) 300,290,195                                    ,103
  195 KACE=1                                                    ,104
      IF (NUMS-1) 300,200,205                                   ,105
  200 ISTOP=1                                                   ,106
      GO TO 210                                                 ,107
  205 ISTOP=NPLN1+KSTART-1                                      ,108
  210 DO 220 I=1,ISTOP                                          ,109
      DO 220 N=1,2                                              ,110
      READ (NREAD,900) XNUM(I,N)                                ,111  ,112  ,113
      NUM=XNUM(I,N)                                             ,114
  220 READ (NREAD,900) (XFOIL(I,J,N),ZFOIL(I,J,N),J=1,NUM)

C     CONVERT AIRFOIL(S) TO UNIT AIRFOIL(S)                     ,115  ,116  ,117  ,118  ,119  ,120
  225 DO 240 I=1,ISTOP                                          ,121
      DO 240 N=1,2                                              ,122
      NUM=XNUM(I,N)                                             ,123
      IF (XFOIL(I,NUM,N)-1.) 230,240,230                        ,124
  230 DO 235 J=1,NUM                                            ,125
      ZFOIL(I,J,N)=ZFOIL(I,J,N)/XFOIL(I,NUM,N)                  ,126
  235 XFOIL(I,J,N)=XFOIL(I,J,N)/XFOIL(I,NUM,N)                  ,127
  240 CONTINUE                                                  ,128  ,129
      GO TO 300                                                 ,130  ,131  ,132

C     READ CARD INPUT - X-, Y- AND Z-COORDINATES OF
C     WING PANEL CORNER POINTS                                  ,133
  250 NPER=ROW+1.                                               ,134
      NPER1=NPER-1                                              ,135
      NPLANE=NPLANE+1                                           ,136
```

```
      NPLN1=NPLANE-1                                                    ,137
      DO 260 I=1,NPLN1                                                  ,138
      DO 260 J=1,NPER1                                                  ,139
 260  READ (NREAD,900) XCOR(I,J),XCOR(I+1,J),XCOR(I,J+1),
     1XCOR(I+1,J+1),YCOR(I,J),YCOR(I+1,J),YCOR(I,J+1),YCOR(I+1,J+1),
     2ZCOR(I,J),ZCOR(I+1,J),ZCOR(I,J+1),ZCOR(I+1,J+1)                   ,140  ,141  ,142  ,143  ,144
      GO TO 300

C     ERROR MESSAGES                                                    ,145
 280  IJ1=IJ+1                                                          ,146
      GO TO (283,285),IJ1                                               ,147
 283  KODEW=1                                                           ,148
      GO TO 300                                                         ,149
 285  KODEWU=1                                                          ,150
      GO TO 300                                                         ,151
 287  WRITE (NWRITE,980)                                                ,152  ,153
      GO TO 283                                                         ,154
 290  WRITE (NWRITE,970)                                                ,155  ,156
      GO TO 283                                                         ,157
 295  WRITE (NWRITE,990)                                                ,158  ,159
      GO TO 283                                                         ,160
 300  RETURN                                                            ,161
 900  FORMAT (6F10.0)
 910  FORMAT (1H1,15X,18HWING PANEL ROUTINE)
 920  FORMAT (1H0,9X,10HTHERE ARE ,I3,67H VERTICAL PLANES THAT INTERSECT
     1 THE WING TO DEFINE PANEL SIDE EDGES)
 930  FORMAT (1H0,9X,12HLEADING-EDGE/10X,
     111HY-INTERCEPT//(F20.5))
 935  FORMAT (1H0,19X,22HOUTBOARD CUTTING PLANE/17X,
     112HLEADING-EDGE/10X,11HX-INTERCEPT,5X,11HY-INTERCEPT,8X,
     25HSLOPE/(F19.5,F16.5,F18.5))
 940  FORMAT (1H0,19X,22HOUTBOARD CUTTING PLANE/17X,
     113HTRAILING-EDGE/10X,11HX-INTERCEPT,5X,11HY-INTERCEPT,8X,
     25HSLOPE/(F19.5,F16.5,F18.5))
 950  FORMAT (1H0,9X,45HZERO-SLOPE INDICATES STREAMWISE CUTTING PLANE)
 960  FORMAT (1H0,9X,43HWING PANELS HAVE BEEN DEFINED ON CARD INPUT)
 970  FORMAT (1H1,9X,63HUPPER AND LOWER SURFACE AIRFOIL COORDINATES NOT
     1READ FROM CARDS)
 980  FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE INPUTW/10X,
     151HIMPROPER WING DEFINITION OR INTERSECTION DEFINITION/10X,
     250HDIMENSION EXCEEDED FOR NPTS, X, Y, Z OR REC ARRAYS)
 990  FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE INPUTW/10X,
     121HINPUT CARD DATA ERROR)
      END                                                               ,162
```

```
      SUBROUTINE INTAPE (THKW)

C     READS IN GEOMETRICAL DATA FROM PANELING AND DEFINITION LINKS

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE

      COMMON /BLOCK/X(200,3,4),Y(200,3,4),Z(200,3,4),XBAR(200)
     1,YBAR(200),ZBAR(200),XC(200),YC(200),ZC(200),THETA(200),ALPHAS(200
     2),AREA(200),CHORD(200),NPART(200),NPANEL,ISYM,A(200),U(200),V(200)
     3,VPM(200),VV(200),W(200),WPM(200),WW(200),VPMM(200),WPMM(200),ALPH
     1AC(100),ALPHAT(100)

      DIMENSION JSAVE(2),NSAVE(2),NROW(2),XB(51),R(51),ZDELTA(51)
      DIMENSION ARRAY(3,10),THETAB(10)


      REWIND NTAPEA                                                     ,1
      REWIND NTAPEB                                                     ,2
      REWIND NTAPEC                                                     ,3
      REWIND NTAPED                                                     ,4
      REWIND NTAPEE                                                     ,5
      REWIND NTAPEF

C     READ IN MACH NUMBER AND CONFIGURATION SYMMETRY  CONDITION         ,6
      READ (NTAPEI,5000) XMACH,SYM                                      ,7    ,8    ,9
      IF (SYM) 50,45,50                                                 ,10
 45   ISYM=1                                                            ,11
      GO TO 55                                                          ,12
 50   ISYM=2                                                            ,13
 55   CONTINUE

C     CODE TO INDICATE IF THE CASE INCLUDES A THICK WING                ,14
      THKW=KACE

C     TEST FOR TYPE OF CONFIGURATION                                    ,15
      IF (NBODY) 120,110,140

C     WING ONLY CASE                                                    ,16
 110  KACE=1                                                            ,17
      JSAVE(1)=0                                                        ,18
      NSAVE(1)=NWING                                                    ,19
      NPANEL=NWING                                                      ,20
      NRG=1                                                             ,21
      GO TO 160

C     BODY ONLY CASE                                                    ,22
 120  KACE=2                                                            ,23
      NWING=0                                                           ,24
      GO TO 325

C     WING AND BODY CASE                                                ,25
 140  KACE=3                                                            ,26
```

```
      JSAVE(1)=0                                                        ,27
      NSAVE(1)=NBODY                                                    ,28
      JSAVE(2)=NBODY                                                    ,29
      NSAVE(2)=NWING                                                    ,30
      NPANEL=NWING+NBODY                                                ,31
      NRG=2
                                                                        ,32
  160 CONTINUE
                                                                        ,33
      DO 225L=1,NRG                                                     ,34
      J=JSAVE(L)                                                        ,35
      N=NSAVE(L)
                                                                        ,36
      DO 190I=1,N                                                       ,37
      J=J+1
C     READS IN PANEL COORDINATES                                        ,38
      READ (NTAPEC) NPN,NPART(J),(X(J,1,K),Y(J,1,K),Z(J,1,K),K=1,4)     ,39  ,40  ,41  ,42  ,43  ,44
                                                                        ,45
      IF (NPART(J)-1)170,190,170                                        ,46
  170 NPT=NPART(J)                                                      ,47
      DO 180M=2,NPT                                                     ,48
      READ (NTAPEC)(X(J,M,K),Y(J,M,K),Z(J,M,K),K=1,4)                   ,49  ,50  ,51  ,52  ,53
  180 CONTINUE                                                          ,54  ,55
  190 CONTINUE
                                                                        ,56  ,57
      J=JSAVE(L)                                                        ,58
      DO 200I=1,N                                                       ,59
      J=J+1
C     READS IN COORDINATES OF CENTROIDS, CONTROL POINTS, AREAS, THETAS,
C     ALPHAS, AND CHORD LENGTHS                                         ,60
      READ (NTAPEC) NPN,XBAR(J),YBAR(J),ZBAR(J),XC(J),YC(J),ZC(J),AREA(J
     1),THETA(J),ALPHAS(J),CHORD(J)                                     ,61  ,62  ,63  ,64
  200 CONTINUE
C     READS IN THE NUMBER OF PANELS IN A COLUMN AND THE LOCATION OF THE
C     CONTROL POINT                                                     ,65  ,66
      READ (NTAPEC) NROW(L),XCTP                                        ,67  ,68  ,69
  225 CONTINUE                                                          ,70  ,71
      REWIND NTAPEC
                                                                        ,72
      DO 250 J=1,NWING                                                  ,73
      JJ=J+NBODY
C     STORES THICKNESS SLOPES AND CAMBER SLOPES                         ,74
      ALPHAT(J)=THETA(JJ)                                               ,75
  250 ALPHAC(J)=ALPHAS(JJ)                                              ,76  ,77
      NS=NBODY+1                                                        ,78
      DO 300 J=NS,NPANEL                                                ,79
      THETA(J)=0.                                                       ,80
  300 ALPHAS(J)=0.
```

```
C     WRITES COORDINATES OF CENTROIDS, CONTROL POINTS, AREAS, THETAS,
C     ALPHAS, CHORD LENGTHS, NUMBER OF REGIONS, NUMBER OF ROWS PER
C     COLUMN IN A REGION, AND CONTROL POINT LOCATION                    ,81   ,82
      WRITE (NTAPEC) (I,XBAR(I),YBAR(I),ZBAR(I),XC(I),YC(I),ZC(I),AREA(I
     1),THETA(I),ALPHAS(I),CHORD(I),I=1,NPANEL),NRG,(NROW(I),I=1,NRG),XC
     2TP                                                               ,83   ,84   ,85   ,86   ,87   ,88
      END FILE NTAPEC                                                  ,89   ,90   ,91   ,92

C     WRITE THICKNESS SLOPES                                            ,93
      WRITE (NTAPEC) (ALPHAT(I),I=1,NWING)                             ,94   ,95   ,96   ,97   ,98
      END FILE NTAPEC

C     WRITE CAMBER SLOPES                                               ,99
      WRITE (NTAPEC) (ALPHAC(I),I=1,NWING)                             ,100  ,101  ,102  ,103  ,104
      END FILE NTAPEC

      IF (NBODY) 325,999,325                                           ,105
  325 CONTINUE                                                         ,106

C     FOR CONFIGURATIONS WITH A BODY READ UNIT VECTORS OF THE BODY AXIS
C     WITH RESPECT TO DEFINING AXIS, AND THE HEIGHT OF THE WING PLANE
C     ABOVE BODY AXIS
      READ (NTAPEB) XUV,YUV,ZUV,ZA,DUMMY1,DUMMY2                       ,107

C     READ NUMBER OF BODY SOURCE SEGMENTS AND THE INDEX OF THE X -
C     COORDINATE AT LEADING EDGE OF THE WING                           ,108  ,109  ,110
      READ (NTAPEB) NBODYS,NXLE

C     READ BODY X-COORDINATES, BODY RADIUS, AND INCREMENT OF BODY CAMBER ,111  ,112  ,113  ,114
      READ (NTAPEB) (XB(I),R(I),ZDELTA(I),I=1,NBODYS)                  ,115  ,116  ,117  ,118  ,119
      REWIND NTAPEB                                                    ,120
      ALPHAA=ZUV/XUV

C     WRITE SLOPE OF BODY AXIS WITH RESPECT TO THE DEFINING AXIS       ,121
      WRITE (NTAPEC) ALPHAA

C     WRITE NUMBER OF BODY SOURCE SEGMENTS, INDEX OF X- COORDINATE AT
C     LEADING EDGE OF WING, HEIGHT OF WING PLANE ABOVE BODY AXIS, X
C     COORDINATE OF BODY , AND RADIUS OF BODY                          ,122  ,123  ,124
      WRITE (NTAPEC) NBODYS,NXLE,ZA,(XB(I),R(I),I=1,NBODYS)            ,125  ,126  ,127  ,128  ,129  ,130

      IF (NWING) 400,350,400                                           ,131
  350 CONTINUE

C     FOR BODY ONLY CASE, READ NUMBER OF THETAS REPRESENTING BODY      ,132
      READ (NTAPED) NDUMMY,NDUMMY,NDUMMY,NTHETA

C     READ BODY THETAS                                                 ,133  ,134  ,135  ,136  ,137  ,138
      READ (NTAPED) ((ARRAY(I,J),I=1,3),J=1,NTHETA)                    ,139  ,140  ,141  ,142  ,143  ,144
```

```
                                                                    ,145  ,146
      REWIND NTAPED                                                 ,147
      DO 375 J=1,NTHETA                                             ,148
  375 THETAB(J)=ARRAY(1,J)                                          ,149  ,150
      GO TO 450

C     FOR WING-BODY CASE, COMPUTE NUMBER OF THETAS AND THETAS FROM
C     PANEL REPRESENTATION OF BODY                                  ,151
  400 CONTINUE                                                      ,152
      NROWB=NROW(1)                                                 ,153
      NTHETA=NBODY/NROWB                                            ,154
      DO 425 J=1,NTHETA                                             ,155
      JJ=(J-1)*NROWB+1                                              ,156
  425 THETAB(J)=ABS(THETA(JJ))*57.2957795                           ,157  ,158
  450 CONTINUE

C     WRITE NUMBER OF THETAS, AND THETAS                            ,159
      WRITE (NTAPEC) NTHETA,(THETAB(I),I=1,NTHETA)

C     WRITE INCREMENT OF BODY CAMBER                                ,160  ,161  ,162  ,163  ,164  ,165

      WRITE (NTAPEC) (ZDELTA(I),I=1,NBODYS)                         ,166  ,167  ,168  ,169  ,170
      END FILE NTAPEC
                                                                    ,171
  999 CONTINUE                                                      ,172
 5000 FORMAT(7F10.0)
      RETURN                                                        ,173
      END                                                           ,174
```

```
      SUBROUTINE INTPCL (NM,NROW,RATIOX,RATIOC,CHORD,ALPHAM,ALPHAD)

C     INTERPOLATES FOR NORMAL VELOCITY COMPONENT AT A SPECIFIED PERCENT
C     OF PANEL CHORD

      DIMENSION CHORD(1),ALPHAM(1),ALPHAD(1)

      NCOL=NM/NROW
      R=RATIOC-RATIOX                                                   ,1
      RR=-R                                                             ,2
      J=0                                                               ,3

      DO 100 K=1,NCOL                                                   ,4
      J=J+1                                                             ,5
      ALPHAD(J)=ALPHAM(J)+R*(ALPHAM(J+1)-ALPHAM(J))/(1.+RATIOX          ,6
     1*(CHORD(J+1)/CHORD(J)-1.))
      DO 100 I=2,NROW                                                   ,7
      J=J+1                                                             ,8
      CR=CHORD(J)/CHORD(J-1)                                            ,9
      ALPHAD(J)=ALPHAM(J)+CR*R*(ALPHAM(J)-ALPHAM(J-1))/(1.+             ,10
     1RATIOX*(CR-1.))
  100 CONTINUE                                                          ,11

      RETURN                                                            ,12  ,13  ,14
      END                                                               ,15
                                                                        ,16
```

```
      FUNCTION INTURP(IDIC,NDIC,LI,LO)

C     READ AND WRITE A COMMAND CARD,SEE IF COMMAND (IN THE
C     FIRST WORD) IS IN THE DICTIONARY.

      DIMENSION IDIC(1),COM(14)
      EQUIVALENCE (COM,ICOM)

C     IDIC (INSTEAD OF DIC) IS USED AS AN ARGUMENT
C     SO THAT A FIXED-POINT COMPARISON CAN BE MADE

      READ (LI,110)COM                                          ,1    ,2    ,3
 110  FORMAT(13A6,A2)
      IF(LO)140,140,115                                         ,4
 115  WRITE (LO,120)COM                                         ,5    ,6    ,7
 120  FORMAT(/1H014A6)
 140  DO 200 INTERP=1,NDIC                                      ,8
      IF (IDIC(INTERP)-ICOM)200,500,200                         ,9
 200  CONTINUE
                                                                ,10   ,11
      INTERP=0                                                  ,12
 500  INTURP=INTERP                                             ,13
      RETURN                                                    ,14
      END                                                       ,15
```

```
$TEXT  INV4S                                                                    INV40001

                                    *INV4S   SINGLE PRECISION 7094 INVERSION ROUTINE (ADAPTION OF INV4D)
                                    *              CALLING SEQUENCE -
                                    *              CALL INV4S,A,N,ERR1,ERR2,SCALE,DET,DETXP
                                    *              A = STARTING ADDRESS OF MATRIX
                                    *                   N = NUMBER OF ROWS (COLUMNS) OF MATRIX
                                    *                ERR1 = 0 IF INVERSION SUCCESSFUL
                                    *                     = 1 IF OVERFLOW OCCURS
                                    *                     = 2 IF MATRIX IS SINGULAR
                                    *                     = 3 IF SCALED INVERSE CANNOT BE RE-SCALED
                                    *                     = 4 IF ROWS AND COLUMNS CANNOT BE RE-ARRANGED
                                    *                         THIS IS BASICALLY A MACHINE ERROR
                                    *                     = 10 + I IF ERROR CODE 3 OCCURED SUBSEQUENT
                                    *                              TO ERROR CODE I
                                    *                     = 20 + I IF ERROR CODE 4 OCCURED SUBSEQUENT
                                    *                              TO ERROR CODE I
                                    *                ERR2 = CURRENT REDUCTION STAGE IF ERR1=1
                                    *                     = RANK OF MATRIX IF ERR1 =2
                                    *                     = 0 OTHERWISE
                                    *               SCALE = 0 IF ERR1 NOT = 3
                                    *                     = SCALING FACTOR IF ERR1=3
                                    *                 DET= DECIMAL PART OF DETERMINANT IF ERR1.
                                    *                       NOT = 2 - A NUMBER GT OR E TO 1. AND LT 10.
                                    *                     = 0 IF ERR1=2
                                    *               DETXP = EXPONENT PART OF DETERMINANT IF ERR1
                                    *                       NOT = 2 - AN ADDRESS INTEGER GIVING
                                    *                       POWER OF TEN FOR DETERMINANT
                                    *                       DETERMINANT = DET*10.**DETXP
                                          ENTRY   INV4S
                                    *
                                    *              SET UP CONSTANTS AND INITIALIZE CELLS
                                    *

BINARY CARD ID. INV40002
      00000  0760 00 0 00016  10000   INV4S LMTM
      00001  4760 00 0 00002  10000         EFTM
      00002  0634 00 1 00361  10001         SXA     XR1,1
      00003  0634 00 2 00362  10001         SXA     XR2,2
      00004  0634 00 3 00363  10001         SXA     XR3,3
      00005  0634 00 4 00364  10001         SXA     XR4,4
      00006  0634 00 5 00365  10001         SXA     XR5,5
      00007  0634 00 6 00366  10001         SXA     XR6,6
      00010  0634 00 7 00367  10001         SXA     XR7,7
      00011  0600 00 0 00000  10000         STZ     0               CLEAR CELL 0 FOR FPT
      00012  0600 00 0 00466  10001         STZ     CHRCNG          CLEAR CELL FOR CHARACTERISTIC SCALING
      00013  0600 00 0 00467  10001         STZ     ERRCD1          CLEAR ERROR CODE CELLS
      00014  0600 00 0 00470  10001         STZ     ERRCD2
      00015  4500 00 0 00010  10000         CAL     8               SAVE CELL 8 FOR (FPT)
      00016  0602 00 0 00443  10001         SLW     TEMP8
      00017  4500 00 0 00444  10001         CAL     FPTTRA          STORE TRANSFER FOR FPT ANALYSIS.
      00020  0602 00 0 00010  10000         SLW     8
      00021  0600 60 4 00007  10000         STZ*    7,4             ZERO SCALING CELL
      00022  0600 00 0 00451  10001         STZ     DETXP           ZERO DETERMINANT EXPONENT
```

```
BINARY CARD ID. INV40003
      00023  0560 60 4 00004  10000          LDQ*    4,4
      00024  4600 00 0 00452  10001          STQ     N
      00025  0200 00 0 00452  10001          MPY     N
      00026  0131 00 0 00000  10000          XCA
      00027  0601 00 0 00453  10001          STO     NSQ          N**2
      00030  0361 00 4 00003  10000          ACL     3,4
      00031  0621 00 0 00461  10001          STA     APLNSQ        A + N**2
      00032  0361 00 0 00452  10001          ACL     N
      00033  0621 00 0 00462  10001          STA     ADRINT       ADDRESS OF ROW INTEGERS +N
      00034  0361 00 0 00452  10001          ACL     N
      00035  0621 00 0 00463  10001          STA     ADCINT       ADDRESS OF COLUMN INTEGERS +N
      00036  0621 00 0 00447  10001          STA     MULTPL       ADDRESS OF REDUCTION FACTOR
      00037  0400 00 0 00733  10001          ADD     =1
      00040  0621 00 0 00446  10001          STA     DET          ADDRESS OF DETERMINANT
      00041  0400 00 0 00733  10001          ADD     =1
      00042  0621 00 0 00450  10001          STA     TEN          ADDRESS OF FLOATING POINT 10
      00043  0500 00 4 00003  10000          CLA     3,4
      00044  0621 00 0 00454  10001          STA     ADA          ADDRESS OF A
      00045  0621 00 0 00455  10001          STA     ADPVEL       ADDRESS OF CURRENT PIVOT ELEMENT

BINARY CARD ID. INV40004
      00046  0361 00 0 00452  10001          ACL     N
      00047  0621 00 0 00456  10001          STA     ADPVRW       ADDRESS OF CURRENT PIVOT ROW +   N
      00050  0534 00 1 00452  10001          LXA     N,1
      00051  4634 00 1 00652  10001          SXD     REDUC4,1     STUFF LOOP CONTROL FOR REDUCE
      00052  0500 00 0 00734  10001          CLA     =1.
      00053  0601 60 0 00446  10001          STO*    DET          START DET OFF AT 1.
      00054  0500 00 0 00735  10001          CLA     =10.
      00055  0601 60 0 00450  10001          STO*    TEN          LOAD 10 IN DOUBLE PRECISION CELL
      00056  0534 00 7 00452  10001          LXA     N,7
      00057  4634 00 7 00204  10001          SXD     RDLPED,7     TEST AT END OF REDUCTION LOOP
      00060  4634 00 7 00067  10001          SXD     STOINT,7     PUT INTEGERS INTO ROW AND COLUMN
      00061  0754 00 0 00000  10000          PXA     0,0
      00062  0774 00 1 00001  10000          AXT     1,1          INTERCHANGE ARRAYS
      00063  0754 00 1 00000  10000          PXA     0,1
      00064  0601 60 0 00463  10001          STO*    ADCINT       ADCINT HAS A TAG OF 1
      00065  0601 60 0 00462  10001          STO*    ADRINT       ADRINT HAS A TAG OF 1
      00066  1 00001 1 01001  10011          TXI     *+1,1,1      INTEGERS ARE STORED  BACKWARDS
      00067  7 00000 1 41004  10011  STOINT  TXL     *-4,1,**     N IN DECREMENT
                                     *
                                     *              START OF REDUCTION LOOP
                                     *              REDUCTION STAGE IS KEPT IN XR1
                                     *              AT THE END OF THIS LOOP ALL INVERSION
                                     *                 ARITHMETIC IS DONE
                                     *
      00070  0774 00 1 00001  10000          AXT     1,1          INITIALIZE REDUCTION STAGE

BINARY CARD ID. INV40005
      00071  0074 00 4 00546  10001  RDLPST  TSX     FINDLE,4     FIND LARGEST ELEMENT IN N-XR1+1 SUB MATRIX
      00072  0634 00 1 00473  10001          SXA     TEMPXR,1
      00073  0754 00 1 00000  10000          PXA     0,1
      00074  0340 00 0 00464  10001          CAS     LGSTRW       DO WE INTERCHANGE ROWS
      00075  0020 00 0 01002  10011          TRA     *+2          YES
```

```
        00076  0020 00 0 00115  10001          TRA     EXCHCL        NO
        00077  0500 60 0 00446  10001          CLA*    DET           YES,ADJUST DET. SIGN
        00100  0760 00 0 00002  10000          CHS
        00101  0601 60 0 00446  10001          STO*    DET
        00102  0074 00 4 00476  10001          TSX     INRCHR,4      EXCHANGE ROWS TO GET LGST NO ON PIV POS.
105     00103  0074 00 0 00473  10001          TSX     TEMPXR
106     00104  0074 00 0 00464  10001          TSX     LGSTRW
                                         *                            EXCHANGE ROW INTEGERS
        00105  0500 00 0 00462  10001          CLA     ADRINT
        00106  0402 00 0 00464  10001          SUB     LGSTRW
        00107  0621 00 0 01002  10011          STA     *+2
        00110  0621 00 0 01003  10011          STA     *+3
        00111  0500 00 0 00000  10000          CLA     **            LGSTRW INTEGER
        00112  0560 60 0 00462  10001          LDQ*    ADRINT        ADRINT HAS TAG OF 1
        00113  4600 00 0 00000  10000          STQ     **            LGSTRW INTEGER

BINARY CARD ID. INV40006
        00114  0601 60 0 00462  10001          STO*    ADRINT
        00115  0754 00 1 00000  10000  EXCHCL  PXA     0,1
        00116  0340 00 0 00465  10001          CAS     LGSTCL        DO WE INTERCHANGE COLS.
        00117  0020 00 0 01002  10011          TRA     *+2           YES
        00120  0020 00 0 00137  10001          TRA     STRTSC        NO
        00121  0500 60 0 00446  10001          CLA*    DET           YES, ADJUST DET. SIGN
        00122  0760 00 0 00002  10000          CHS
        00123  0601 60 0 00446  10001          STO*    DET
        00124  0074 00 4 00521  10001          TSX     INRCHC,4      INTERCHANGE COLUMNS FOR LGST NO.
124     00125  0074 00 0 00473  10001          TSX     TEMPXR
125     00126  0074 00 0 00465  10001          TSX     LGSTCL
                                         *                            INTERCHANGE COLUMN INTEGERS
        00127  0500 00 0 00463  10001          CLA     ADCINT
        00130  0402 00 0 00465  10001          SUB     LGSTCL
        00131  0621 00 0 01002  10011          STA     *+2
        00132  0621 00 0 01003  10011          STA     *+3
        00133  0500 00 0 00000  10000          CLA     **
        00134  0560 60 0 00463  10001          LDQ*    ADCINT        ADCINT HAS A TAG OF 1
        00135  4600 00 0 00000  10000          STQ     **
        00136  0601 60 0 00463  10001          STO*    ADCINT
                                         *
                                         *                 START OF CHARACTERISTIC SCALING ROUTINE
                                         *                 THE LARGEST MAGNITUDE IN THE MATRIX WILL
                                         *                 BE FORCED TO HAVE A CHARACTERISTIC OF 177
                                         *                 AND ALL OTHERS WILL BE ADJUSTED ACCORDINGLY
                                         *

BINARY CARD ID. INV40007
        00137  3 00001 1 00162  10001  STRTSC  TXH     SCLOOP+1,1,1
        00140  4500 60 0 00455  10001          CAL*    ADPVEL        LARGEST ELEMENT = PIVOT ELEMENT
        00141  4320 00 0 00736  10001          ANA     =0377000000000   MASK OUT ALL BUT CHARACTERISTIC
        00142  0402 00 0 00737  10001          SUB     =0177000000000   FIND DIFFERENCE FROM 177
        00143  0760 00 0 00002  10000          CHS
        00144  0601 00 0 00466  10001          STO     CHRCNG        SAVE CHARACTERISTIC SCALING
        00145  0100 00 0 00162  10001          TZE     SCLOOP+1      BY-PASS SCALING IF NOT NEEDED
        00146  0534 00 7 00453  10001          LXA     NSQ,7
        00147  4520 60 0 00461  10001  FSTSCL  NZT*    APLNSQ        APLNSQ HAS A TAG OF 7
```

```
00150  0020 00 0 00161  10001          TRA     SCLOOP        NUMBER IS ZERO, DO NOT SCALE
00151  0500 00 0 00466  10001          CLA     CHRCNG        SCALE NUMBER
00152  0401 60 0 00461  10001          ADM*    APLNSQ
00153  0120 00 0 01003  10011          TPL     *+3
00154  0600 60 0 00461  10001          STZ*    APLNSQ        IF NEGATIVE, UNDERFLOW OCCURRED
00155  0020 00 0 00161  10001          TRA     SCLOOP
00156  0560 60 0 00461  10001          LDQ*    APLNSQ        SCALING SUCCESSFUL, RESTORE SIGN
00157  0763 00 0 00000  10000          LLS     0
00160  0601 60 0 00461  10001          STO*    APLNSQ        STORE IN ORIGINAL LOCATION
00161  2 00001 7 00147  10001  SCLOOP  TIX     FSTSCL,7,1    END OF SCALING LOOP

BINARY CARD ID. INV40008
00162  0074 00 4 00612  10001          TSX     REDUCE,4      PERFORM CURRENT REDUCTION STAGE
00163  0500 60 0 00455  10001          CLA*    ADPVEL
00164  0074 00 4 00654  10001          TSX     SCALE,4       SCALE PIVOT ELEMENT
00165  0131 00 0 00000  10000          XCA
00166  0260 60 0 00446  10001          FMP*    DET
00167  0074 00 4 00654  10001          TSX     SCALE,4       SCALE PRODUCT
00170  0601 60 0 00446  10001          STO*    DET           RUNNING PRODUCT OF PIVOT ELEMENTS
00171  0534 00 7 00452  10001          LXA     N,7
00172  0500 60 0 00455  10001          CLA*    ADPVEL
00173  0601 60 0 00447  10001          STO*    MULTPL        KEEP PIVOT ELEMENT TEMP. IN MULTPL
00174  0500 60 0 00456  10001  ADJUST  CLA*    ADPVRW        ADPVRW HAS TAG OF 7
00175  0241 60 0 00447  10001          FDP*    MULTPL
00176  4600 60 0 00456  10001          STQ*    ADPVRW        DIVIDE PIVOT ROW BY PIVOT ELEMENT
00177  2 00001 7 00174  10001          TIX     ADJUST,7,1
00200  0500 00 0 00734  10001          CLA     =1.
00201  0241 60 0 00447  10001          FDP*    MULTPL
00202  4600 60 0 00455  10001          STQ*    ADPVEL        PUT 1/(PIV. ELEM) INTO PIVOT POSITION
00203  1 00001 1 01001  10011          TXI     *+1,1,1       INCREMENT TO NEW REDUCTION STAGE
00204  3 00000 1 00215  10001  RDLPED  TXH     REDOVR,1,**   N IN DECREMENT-LOOP EXIT

BINARY CARD ID. INV40009
00205  0500 00 0 00455  10001          CLA     ADPVEL
00206  0400 00 0 00733  10001          ADD     =1
00207  0400 00 0 00452  10001          ADD     N
00210  0621 00 0 00455  10001          STA     ADPVEL        INCREMENT BY   N+1
00211  0500 00 0 00456  10001          CLA     ADPVRW
00212  0400 00 0 00452  10001          ADD     N
00213  0621 00 0 00456  10001          STA     ADPVRW        INCREMENT BY   N
00214  0020 00 0 00071  10001          TRA     RDLPST        ANOTHER REDUCTION STAGE
                                  *
                                  *                ALL ARITHMETIC IS OVER NOW REARANGE ROWS
                                  *                AND COLUMNS
                                  *
                                  *                FIRST, INTERCHANGE COLUMNS ACCORDING TO ROW TABLE
                                  *
00215  0534 00 7 00452  10001  REDOVR LXA     N,7
00216  0600 00 0 00471  10001          STZ     TEMP1         CLEAR FOR XR USE
00217  0754 00 7 00000  10000          PXA     0,7
00220  0734 00 1 00000  10000          PAX     0,1
00221  0754 00 7 00000  10000  RWSRCH PXA     0,7
00222  0402 60 0 00462  10001          SUB*    ADRINT        ADRINT HAS A TAG OF 1
00223  0100 00 0 00226  10001          TZE     FOUNDR
```

```
        00224  2 00001 1 00221  10001          TIX     RWSRCH,1,1                                  202
        00225  0020 00 0 00431  10001          TRA     ERR4          CANNOT FIND ROW               203
        00226  0754 00 7 00000  10000  FOUNDR  PXA     0,7                                         204
        00227  0634 00 1 00471  10001          SXA     TEMP1,1                                     205

BINARY CARD ID. INV40010
        00230  0402 00 0 00471  10001          SUB     TEMP1                                       206
        00231  0100 00 0 00246  10001          TZE     RWLPED        COLUMN IS IN CORRECT PLACE    207
        00232  0634 00 7 00473  10001          SXA     TEMPXR,7                                    208
        00233  0074 00 4 00521  10001          TSX     INRCHC,4                                    209
208     00234  0074 00 0 00473  10001          TSX     TEMPXR        ROW INTEGER FOUND             210
209     00235  0074 00 0 00471  10001          TSX     TEMP1         LOCATION OF ROW INTEGER       211
                                         *               INTERCHANGE ROW INTEGERS.                   212
        00236  0534 00 7 00473  10001          LXA     TEMPXR,7                                    213
        00237  0500 00 0 00462  10001          CLA     ADRINT                                      214
        00240  0402 00 0 00473  10001          SUB     TEMPXR                                      215
        00241  0621 00 0 01001  10011          STA     *+1                                         216
        00242  0500 00 0 00000  10000          CLA     **                                          217
        00243  0601 60 0 00462  10001          STO*    ADRINT        ADRINT HAS A TAG OF 1         218
        00244  0500 00 0 00473  10001          CLA     TEMPXR                                      219
        00245  0601 60 0 41003  10011          STO*    *-3                                         220
        00246  2 00001 7 00217  10001  RWLPED  TIX     RWSRCH-2,7,1  LOOP FOR N ROWS.              221
                                         *                                                         222
                                         *               NOW INTERCHANGE ROWS ACCORDING TO COLUMN TABLE 223
                                         *                                                         224
        00247  0534 00 7 00452  10001          LXA     N,7                                         225
        00250  0754 00 7 00000  10000          PXA     0,7                                         226
        00251  0734 00 1 00000  10000          PAX     0,1                                         227
        00252  0754 00 7 00000  10000  CLSRCH  PXA     0,7                                         228

BINARY CARD ID. INV40011
        00253  0402 60 0 00463  10001          SUB*    ADCINT        ADCINT HAS A TAG OF 1         229
        00254  0100 00 0 00257  10001          TZE     FOUNDC                                      230
        00255  2 00001 1 00252  10001          TIX     CLSRCH,1,1                                  231
        00256  0020 00 0 00431  10001          TRA     ERR4          CANNOT FIND COLUMN            232
        00257  0754 00 7 00000  10000  FOUNDC  PXA     0,7                                         233
        00260  0634 00 1 00471  10001          SXA     TEMP1,1                                     234
        00261  0402 00 0 00471  10001          SUB     TEMP1                                       235
        00262  0100 00 0 00277  10001          TZE     CLLPED        ROW IS IN CORRECT PLACE       236
        00263  0634 00 7 00473  10001          SXA     TEMPXR,7                                    237
        00264  0074 00 4 00476  10001          TSX     INRCHR,4                                    238
237     00265  0074 00 0 00473  10001          TSX     TEMPXR                                      239
238     00266  0074 00 0 00471  10001          TSX     TEMP1                                       240
                                         *               INTERCHANGE COLUMN INTEGERS                 241
        00267  0534 00 7 00473  10001          LXA     TEMPXR,7                                    242
        00270  0500 00 0 00463  10001          CLA     ADCINT                                      243
        00271  0402 00 0 00473  10001          SUB     TEMPXR                                      244
        00272  0621 00 0 01001  10011          STA     *+1                                         245
        00273  0500 00 0 00000  10000          CLA     **                                          246
        00274  0601 60 0 00463  10001          STO*    ADCINT        ADCINT HAS A TAG OF 1         247
        00275  0500 00 0 00473  10001          CLA     TEMPXR                                      248

BINARY CARD ID. INV40012
        00276  0601 60 0 41003  10011          STO*    *-3                                         249
```

```
00277  2 00001 7 00250  10001  CLLPED TIX     CLSRCH-2,7,1  LOOP FOR N COLUMNS
                                *
                                *             INVERSE IS NOW IN THE CORRECT FORM - TIME TO
                                *             RESCALE THE SCALED INVERSE
                                *
00300  0500 00 0 00466  10001         CLA     CHRCNG
00301  0100 00 0 00361  10001         TZE     DSLPED+1      IF CHRCNG=0, RESCALING UNNECESSARY
00302  4120 00 0 00315  10001         TMI     RESCAL        IF NEGATIVE, NO OVERFLOW PROBLEMS
00303  0774 00 1 00001  10000         AXT     1,1           SET UP XR1 FOR FINDLE
00304  0500 00 0 00454  10001         CLA     ADA           SET UP TO PIVOT ROW 1 FOR FINDLE
00305  0400 00 0 00452  10001         ADD     N
00306  0621 00 0 00456  10001         STA     ADPVRW
00307  0074 00 4 00546  10001         TSX     FINDLE,4      FIND LARGEST ELEMENT IN SCALED INVERSE
00310  0500 00 0 00466  10001         CLA     CHRCNG
00311  0401 00 0 00471  10001         ADM     TEMP1         FINDLE STORES LARGEST IN TEMP1
00312  4760 00 0 00001  10000         PBT                   WAS THERE OVERFLOW
00313  0020 00 0 01002  10011         TRA     *+2           NO
00314  0020 00 0 00431  10001         TRA     ERR4          YES
00315  0534 00 7 00453  10001  RESCAL LXA     NSQ,7
00316  0500 00 0 00466  10001         CLA     CHRCNG
00317  0401 60 0 00461  10001         ADM*    APLNSQ        APLNSQ HAS A TAG OF 7
00320  0120 00 0 01003  10011         TPL     *+3           NO UNDERFLOW IF POSITIVE

BINARY CARD ID. INV40013
00321  0600 60 0 00461  10001         STZ*    APLNSQ        UNDERFLOW, STORE ZERO
00322  0020 00 0 00326  10001         TRA     RSCLED
00323  0560 60 0 00461  10001         LDQ*    APLNSQ
00324  0763 00 0 00000  10000         LLS     0
00325  0601 60 0 00461  10001         STQ*    APLNSQ
00326  2 00001 7 00316  10001  RSCLED TIX     RESCAL+1,7,1  LOOP FOR 2*N**2 NUMBERS
                                *
                                *             NOW RESCALE THE DETERMINANT
                                *
00327  4520 60 0 00446  10001         NZT*    DET           IS MATRIX SINGULAR
00330  0020 00 0 00361  10001         TRA     DSLPED+1      IF SINGULAR, RETURN
00331  0500 00 0 00466  10001         CLA     CHRCNG
00332  0760 00 0 00002  10000         CHS
00333  0601 00 0 00466  10001         STQ     CHRCNG
00334  0534 00 7 00452  10001         LXA     N,7
00335  0500 00 0 00466  10001  STDSCL CLA     CHRCNG
00336  0401 60 0 00446  10001         ADM*    DET
00337  4760 00 0 00001  10000         PBT
00340  0020 00 0 01002  10011         TRA     *+2           NO OVERFLOW
00341  0020 00 0 00350  10001         TRA     DIVDET        OVERFLOW, DIVIDE BY 10
00342  4120 00 0 00354  10001         TMI     MLTDET        UNDERFLOW, MULTIPLY BY 10
00343  0560 60 0 00446  10001         LDQ*    DET

BINARY CARD ID. INV40014
00344  0763 00 0 00000  10000         LLS     0
00345  0074 00 4 00654  10001         TSX     SCALE,4
00346  0601 60 0 00446  10001         STQ*    DET
00347  0020 00 0 00360  10001         TRA     DSLPED        GO TO END OF LOOP
00350  0500 60 0 00446  10001  DIVDET CLA*    DET           DIVIDE DET BY TEN
00351  0241 60 0 00450  10001         FDP*    TEN
```

```
00352  4600 60 0 00446  10001          STQ*    DET
00353  0020 00 0 00335  10001          TRA     STDSCL
00354  0560 60 0 00446  10001  MLTDET  LDQ*    DET            MULTIPLY DET BY TEN
00355  0260 60 0 00450  10001          FMP*    TEN
00356  0601 60 0 00446  10001          STQ*    DET
00357  0020 00 0 00335  10001          TRA     STDSCL
00360  2 00001 7 00335  10001  DSLPED  TIX     STDSCL,7,1     LOOP N TIMES FOR COMPLETE SCALING
                                *
                                *              END OF ALL ARITHMETIC NOW TIME TO ARRANGE
                                *              ERROR CODES AND SUCH THEN RETURN
00361  0774 00 1 00000  10000     XR1  AXT     **,1
00362  0774 00 2 00000  10000     XR2  AXT     **,2
00363  0774 00 3 00000  10000     XR3  AXT     **,3
00364  0774 00 4 00000  10000     XR4  AXT     **,4
00365  0774 00 5 00000  10000     XR5  AXT     **,5
00366  0774 00 6 00000  10000     XR6  AXT     **,6

BINARY CARD ID. INV40015
00367  0774 00 7 00000  10000     XR7  AXT     **,7
00370  4500 00 0 00443  10001          CAL     TEMP8          RESTORE CELL 8
00371  0602 00 0 00010  10000          SLW     8
00372  0500 60 0 00446  10001          CLA*    DET
00373  0601 60 4 00010  10000          STQ*    8,4
00374  0500 00 0 00451  10001          CLA     DETXP
00375  0601 60 4 00011  10000          STQ*    9,4
00376  0500 00 0 00467  10001          CLA     ERRCD1
00377  0601 60 4 00005  10000          STQ*    5,4
00400  0500 00 0 00470  10001          CLA     ERRCD2
00401  0601 60 4 00006  10000          STQ*    6,4
00402  0020 00 4 00001  10000          TRA     1,4
                                *
                                *              FOLLOWING ARE ALL THE ERROR STOPS
                                *
00403  0500 00 0 00733  10001    ERR1  CLA     =1             OVERFLOW RETURN
00404  0601 00 0 00467  10001          STQ     ERRCD1
00405  0634 00 1 00470  10001          SXA     ERRCD2,1       SAVE REDUCTION STAGE
00406  0600 60 0 00446  10001          STZ*    DET            ZERO DETERMINANT
00407  0600 00 0 00451  10001          STZ     DETXP
00410  0020 00 0 00215  10001          TRA     REDOVR         TRANSFER BACK TO INTERCHANGE ROWS
                                *                             AND COLUMNS, AND RESCALE
00411  0500 00 0 00740  10001    ERR2  CLA     =2

BINARY CARD ID. INV40016
00412  0601 00 0 00467  10001          STQ     ERRCD1
00413  0754 00 1 00000  10000          SXA     0,1
00414  0402 00 0 00733  10001          SUB     =1
00415  0601 00 0 00470  10001          STQ     ERRCD2         RANK OF MATRIX =XR1-1
00416  0020 00 0 00406  10001          TRA     ERR1+3         TRANSFER BACK TO INTERCHANGE
                                *                             ROWS AND COLUMNS, AND RESCALE
00417  0500 00 0 00467  10001    ERR3  CLA     ERRCD1
00420  4100 00 0 00437  10001          TNZ     ERR3PL
00421  0500 00 0 00741  10001          CLA     =3
00422  0601 00 0 00467  10001          STQ     ERRCD1
00423  0600 00 0 00470  10001          STZ     ERRCD2
```

```
00424  0500 00 0 00734  10001          CLA    =1.          =0201400000000
00425  0400 00 0 00466  10001          ADD    CHRCNG
00426  0534 00 4 00364  10001          LXA    XR4,4
00427  0601 60 4 00007  10000          STO*   7,4          STORE SCALING FACTOR
00430  0020 00 0 00327  10001          TRA    RSCLED+1     GO BACK AND SCALE DETERMINANT
00431  0500 00 0 00467  10001     ERR4 CLA    ERRCD1       CANNOT RE-ARRANGE ROWS,COLUMNS -MACH. ERROR
00432  4100 00 0 00441  10001          TNZ    ERR4PL
00433  0500 00 0 00742  10001          CLA    =4
00434  0601 00 0 00467  10001          STO    ERRCD1       CANNOT RE-ORDER ROWS AND COLUMNS

BINARY CARD ID. INV40017
00435  0600 00 0 00470  10001          STZ    ERRCD2       FPT GIVES STORAGE TRAP
00436  0020 00 0 00361  10001          TRA    DSLPED+1     RETURN IMMEDIATELY
00437  0400 00 0 00743  10001  ERR3PL  ADD    =10
00440  0020 00 0 00422  10001          TRA    ERR3+3
00441  0400 00 0 00744  10001  ERR4PL  ADD    =20
00442  0020 00 0 00434  10001          TRA    ERR4+3
                                *
                                *              STORAGE FOR PROGRAM CONSTANTS
                                *
00443  0 00000 0 00000  10000   TEMP8 PZE    0            STORAGE FOR CORE LOCATION 8 (DEC)
00444  0020 00 0 00707  10001  FPTTRA TRA    FLPSPL
00445  0 00000 0 00000  10000   CELLO PZE    0            STORAGE FOR FLPSPL CODE
00446  0 00000 0 00000  10000     DET PZE    0            ADDRESS OF DETERMINANT
00447  0 00000 0 00000  10000  MULTPL PZE    0            ADDRESS OF REDUCTION FACTOR
00450  0 00000 0 00000  10000     TEN PZE    0            ADDRESS OF FLOATING POINT 10.
00451  0 00000 0 00000  10000   DETXP PZE    0            CELL FOR DETERMINANT EXPONENT
00452  0 00000 0 00000  10000       N PZE    0
00453  0 00000 0 00000  10000     NSQ PZE    0            N**2
00454  0 00000 0 00000  10000     ADA PZE    0            ADDRESS OF MATRIX
00455  0 00000 0 00000  10000  ADPVEL PZE    0            ADDRESS OF CURRENT PIVOT ELEMENT
00456  0 00000 7 00000  10000  ADPVRW PZE    0,7          ADDRESS OF CURRENT PIVOT ROW +2N
00457  0 00000 6 00000  10000  ADOPEL PZE    0,6          ADDRESS OF ELEMENT BEING REDUCED

BINARY CARD ID. INV40018
00460  0 00000 7 00000  10000  ADOPRW PZE    0,7          ROW OF ADOPEL
00461  0 00000 7 00000  10000  APLNSQ PZE    0,7          ADDRESS OF FIRST CELL BEYOND MATRIX
00462  0 00000 1 00000  10000  ADRINT PZE    0,1          ADDRESS OF ROW INTEGERS + N
00463  0 00000 1 00000  10000  ADCINT PZE    0,1          ADDRESS OF COLUMN INTEGERS + N
00464  0 00000 0 00000  10000  LGSTRW PZE    0            ROW INDEX OF LARGEST ELEMENT IN SUB-MATRIX
00465  0 00000 0 00000  10000  LGSTCL PZE    0            COLUMN INDEX OF LARGEST ELEM. IN SUB-MATRIX
00466  0 00000 0 00000  10000  CHRCNG PZE    0            CHARACTERISTIC SCALE FACTOR
00467  0 00000 0 00000  10000  ERRCD1 PZE    0            ERROR CODE 1
00470  0 00000 0 00000  10000  ERRCD2 PZE    0            ERROR CODE 2
00471  0 00000 0 00000  10000   TEMP1 PZE    0
00472  0 00000 0 00000  10000   TEMP2 PZE    0
00473  0 00000 0 00000  10000  TEMPXR PZE    0            TEMPORARY FOR ADDRESS ONLY
00474  0 00000 0 00000  10000  TEMPAC PZE    0            TEMPORARY STORAGE FOR FLPSPL
00475  0 00000 0 00000  10000  TEMPMQ PZE    0            SAME AS ABOVE
                                *              CLOSED SUBROUTINE TO INTERCHANGE ROWS
                                *              TWO ARGUMENTS -I,J=ROWS TO BE MOVED
                                *
                                *
00476  0560 60 4 00001  10000  INPCHR LDQ*   1,4
```

```
00477  0200 00 0 00452  10001          MPY     N
00500  0131 00 0 00000  10000          XCA
00501  0400 00 0 00454  10001          ADD     ADA           ADDRESS OF ROW I +  N
00502  0621 00 0 00513  10001          STA     INR1

BINARY CARD ID. INV40019
00503  0621 00 0 00516  10001          STA     INR1+3
00504  0560 60 4 00002  10000          LDQ*    2,4
00505  0200 00 0 00452  10001          MPY     N
00506  0131 00 0 00000  10000          XCA
00507  0400 00 0 00454  10001          ADD     ADA           ADDRESS OF ROW I +   N
00510  0621 00 0 00514  10001          STA     INR1+1
00511  0621 00 0 00515  10001          STA     INR1+2
00512  0534 00 7 00452  10001          LXA     N,7
00513  0500 00 7 00000  10000     INR1 CLA     **,7          ROW I
00514  0560 00 7 00000  10000          LDQ     **,7          ROW J
00515  0601 00 7 00000  10000          STO     **,7          ROW J
00516  4600 00 7 00000  10000          STQ     **,7          ROW I
00517  2 00001 7 00513  10001          TIX     INR1,7,1
00520  0020 00 4 00003  10000          TRA     3,4           DONE
                                  *
                                  *              CLOSED SUBROUTINE TO INTERCHANGE COLUMNS
                                  *              TWO ARGUMENTS -I,J = COLUMNS TO BE MOVED
                                  *
00521  0500 00 0 00454  10001   INRCHC CLA     ADA
00522  0402 00 0 00733  10001          SUB     =1
00523  0400 00 0 00453  10001          ADD     NSQ
00524  0601 00 0 00472  10001          STO     TEMP2         A-1+  N**2
00525  0500 60 4 00001  10000          CLA*    1,4

BINARY CARD ID. INV40020
00526  0400 00 0 00472  10001          ADD     TEMP2         ADDRESS OF COLUMN I+ N**2+1
00527  0621 00 0 00540  10001          STA     INC1
00530  0621 00 0 00543  10001          STA     INC1+3
00531  0500 60 4 00002  10000          CLA*    2,4
00532  0400 00 0 00472  10001          ADD     TEMP2         ADDRESS OF COLUMN J +  N**2+1
00533  0621 00 0 00541  10001          STA     INC1+1
00534  0621 00 0 00542  10001          STA     INC1+2
00535  0534 00 7 00453  10001          LXA     NSQ,7
00536  0534 00 6 00452  10001          LXA     N,6
00537  4634 00 6 00544  10001          SXD     INC2,6
00540  0500 00 7 00000  10000     INC1 CLA     **,7          COLUMN I
00541  0560 00 7 00000  10000          LDQ     **,7          COLUMN J
00542  0601 00 7 00000  10000          STO     **,7          COLUMN J
00543  4600 00 7 00000  10000          STQ     **,7          COLUMN I
00544  2 00000 7 00540  10001     INC2 TIX     INC1,7,0      MAIN LOOP -2N IN DECREMENT
00545  0020 00 4 00003  10000          TRA     3,4           DONE
                                  *
                                  *              CLOSED SUBROUTINE TO FIND THE LARGEST
                                  *              MAGNITUDE IN THE N-XR1+1 SQUARE SUB-MATRIX
                                  *              CURRENT REDUCTION STAGE IS IN XR1
                                  *              ROW LOCATION OF L.F. = LGSTRW
                                  *              COLUMN LOCATION OF L.F. = LGSTCL
                                  *
```

```
00546  0754 00 1 00000  10000  FINDLE PXA   0,1
00547  0402 00 0 00452  10001         SUB   N
00550  0402 00 0 00733  10001         SUB   =1

BINARY CARD ID. INV40021
00551  0734 00 3 00000  10000         PAX   0,3          COUNT OF ROWS IN XR3=N+1-XR1
00552  0621 00 0 00473  10001         STA   TEMPXR       TEMPXR=2(N+1-XR1)-TO GO INTO XR7
00553  0634 00 1 00464  10001         SXA   LGSTRW,1     START LOCATION AT PIVOT ELEMENT
00554  0634 00 1 00465  10001         SXA   LGSTCL,1
00555  0500 00 0 00456  10001         CLA   ADPVRW
00556  0621 00 0 00562  10001         STA   FLE1+1
00557  0600 00 0 00471  10001         STZ   TEMP1        KEEP LARGEST IN TEMP1
00560  0534 00 7 00473  10001         LXA   TEMPXR,7
00561  0754 00 0 00000  10000    FLE1 PXA   0,0
00562  0401 00 7 00000  10000         ADM   **,7
00563  0340 00 0 00471  10001         CAS   TEMP1
00564  0020 00 0 01003  10011         TRA   *+3          GREATER - EXCHANGE
00565  0020 00 0 01005  10011         TRA   *+5
00566  0020 00 0 01004  10011         TRA   *+4
00567  0601 00 0 00471  10001         STO   TEMP1
00570  0634 00 3 00464  10001         SXA   LGSTRW,3
00571  0634 00 7 00465  10001         SXA   LGSTCL,7
00572  2 00001 7 00561  10001         TIX   FLE1,7,1     LOOP BACK IF ROW NOT DONE
00573  4500 00 0 00562  10001         CAL   FLE1+1

BINARY CARD ID. INV40022
00574  0400 00 0 00452  10001         ADD   N
00575  0621 00 0 00562  10001         STA   FLE1+1       INCREMENT TO NEW ROW
00576  2 00001 3 00560  10001         TIX   FLE1-1,3,1   LOOP BACK IF MORE ROWS
00577  4520 00 0 00471  10001         NZT   TEMP1        DONE, IS LARGEST ELEMENT = 0
00600  0020 00 0 00411  10001         TRA   ERR2         SINGULAR MATRIX ERROR RETURN
                                *                        ADJUST LGSTRW, CL
00601  0502 00 0 00464  10001         CLS   LGSTRW
00602  0400 00 0 00452  10001         ADD   N
00603  0400 00 0 00733  10001         ADD   =1
00604  0621 00 0 00464  10001         STA   LGSTRW
00605  0502 00 0 00465  10001         CLS   LGSTCL
00606  0400 00 0 00452  10001         ADD   N
00607  0400 00 0 00733  10001         ADD   =1
00610  0621 00 0 00465  10001         STA   LGSTCL
00611  0020 00 4 00001  10000         TRA   1,4
                                *
                                *            CLOSED SUBROUTINE TO PERFORM ONE REDUCTION
                                *            STAGE - ONE STAGE CONSISTS OF PRODUCING
                                *            ZEROS ON THE PIVOT COLUMN AND A 1 IN THE
                                *            PIVOT POSITION- ZEROS AND 1 ARE ONLY UNDERSTOOD
                                *            DUE TO STORAGE OVER ITSELF
                                *            XR1=REDUCTION STAGE
                                *
00612  0534 00 6 00453  10001  REDUCE LXA   NSQ,6
00613  0634 00 1 00473  10001         SXA   TEMPXR,1
00614  0500 00 0 00454  10001         CLA   ADA
00615  0400 00 0 00452  10001         ADD   N
00616  0621 00 0 00460  10001         STA   ADOPRW        N + ADDRESS OF ROW BEING REDUCED
```

```
BINARY CARD ID. INV40023
        00617  0402 00 0 00452  10001          SUB     N
        00620  0400 00 0 00473  10001          ADD     TEMPXR
        00621  0402 00 0 00733  10001          SUB     =1
        00622  0400 00 0 00453  10001          ADD     NSQ           N**2 + ADDRESS OF ELEMENT BEING ZEROED
        00623  0621 00 0 00457  10001          STA     ADOPEL
        00624  0774 00 5 00001  10000          AXT     1,5           COUNT OF ROWS BEING REDUCED
        00625  0754 00 5 00000  10000  REDUC1  PXA     0,5
        00626  0402 00 0 00473  10001          SUB     TEMPXR
        00627  0100 00 0 00646  10001          TZE     REDUC3        DO NOT OPERATE ON PIVOT ROW
        00630  0500 60 0 00457  10001          CLA*    ADOPEL        ADOPEL HAS A TAG OF 6
        00631  0100 00 0 00646  10001          TZE     REDUC3        IF ALREADY ZERO, NO REDUCTION NECESSARY
        00632  0241 60 0 00455  10001          FDP*    ADPVEL
        00633  0131 00 0 00000  10000          XCA
        00634  0760 00 0 00002  10000          CHS                   USE NEGATIVE RATIO
        00635  0601 60 0 00447  10001          STO*    MULTPL        REDUCTION FACTOR
        00636  0534 00 7 00452  10001          LXA     N,7
        00637  0560 60 0 00456  10001  REDUC2  LDQ*    ADPVRW        ADPVRW HAS A TAG OF 7
        00640  0260 60 0 00447  10001          FMP*    MULTPL
        00641  0300 60 0 00460  10001          FAD*    ADOPRW        ADOPRW HAS A TAG OF 7

BINARY CARD ID. INV40024
        00642  0601 60 0 00460  10001          STO*    ADOPRW
        00643  2 00001 7 00637  10001          TIX     REDUC2,7,1
        00644  0500 60 0 00447  10001          CLA*    MULTPL        PUT REDUCTION FACTOR WHERE
        00645  0601 60 0 00457  10001          STO*    ADOPEL        ZERO WAS PRODUCED
        00646  0500 00 0 00460  10001  REDUC3  CLA     ADOPRW
        00647  0400 00 0 00452  10001          ADD     N
        00650  0621 00 0 00460  10001          STA     ADOPRW        GO TO NEXT ROW
        00651  1 00001 5 01001  10011          TXI     *+1,5,1       XR5 = ROW ABOUT TO BE REDUCED
        00652  2 00000 6 00625  10001  REDUC4  TIX     REDUC1,6,**    N IN DECREMENT
        00653  0020 00 4 00001  10000          TRA     1,4           DONE
                                       *
                                       *              CLOSED SUBROUTINE TO SCALE AN INCOMING
                                       *              NUMBER SO THAT IT IS LESS THAN 10. AND
                                       *              GREATER THAN OR EQUAL TO 1. DIVISIONS AND
                                       *              MULTIPLICATIONS KEPT TRACK OF IN DETXP.
                                       *              NUMBER EXPECTED IN AC AND MQ-RESULT LEFT THERE
                                       *
        00654  0601 00 0 00474  10001   SCALE STO     TEMPAC
        00655  0760 00 0 00003  10000          SSP
        00656  0340 60 0 00450  10001  SCALE1  CAS*    TEN
        00657  0020 00 0 00674  10001          TRA     DIVIDE
        00660  0020 00 0 00674  10001          TRA     DIVIDE
        00661  0340 00 0 00734  10001  TSTONE  CAS     =1.
        00662  0020 00 0 00703  10001          TRA     CHKSGN
        00663  0020 00 0 00703  10001          TRA     CHKSGN
        00664  0131 00 0 00000  10000          XCA

BINARY CARD ID. INV40025
        00665  0260 60 0 00450  10001          FMP*    TEN
        00666  0601 60 0 00447  10001          STO*    MULTPL        TEMPORARY STORAGE
        00667  0500 00 0 00451  10001          CLA     DETXP
        00670  0402 00 0 00733  10001          SUB     =1
```

```
00671  0601 00 0 00451  10001          STO     DETXP
00672  0500 60 0 00447  10001          CLA*    MULTPL
00673  0020 00 0 00661  10001          TRA     TSTONE
00674  0241 60 0 00450  10001  DIVIDE  FDP*    TEN
00675  4600 60 0 00447  10001          STQ*    MULTPL
00676  0500 00 0 00451  10001          CLA     DETXP
00677  0400 00 0 00733  10001          ADD     =1
00700  0601 00 0 00451  10001          STO     DETXP
00701  0500 60 0 00447  10001          CLA*    MULTPL
00702  0020 00 0 00656  10001          TRA     SCALE1
00703  0560 00 0 00474  10001  CHKSGN  LDQ     TEMPAC
00704  0763 00 0 00000  10000          LLS     0
00705  0560 00 0 00745  10001          LDQ     =0
00706  0020 00 4 00001  10000          TRA     1,4
                                *
                                *               FLOATING POINT SPILL ROUTINE TO ANALYZE
                                *               OVER/UNDER FLOW DURING DOUBLE PRECISION
                                *               OPERATIONS
                                *
                                *
                                *
                                *
00707  0601 00 0 00474  10001  FLPSPL  STO     TEMPAC

BINARY CARD ID. INV40026
00710  0500 00 0 00000  10000          CLA     0
00711  0622 00 0 00445  10001          STD     CELL0
00712  0500 00 0 00445  10001          CLA     CELL0
00713  0340 00 0 00746  10001          CAS     =3B17
00714  0020 00 0 00723  10001          TRA     GRT3
00715  0020 00 0 00721  10001          TRA     ALLUND
00716  0500 00 0 00474  10001          CLA     TEMPAC
00717  0560 00 0 00745  10001          LDQ     =0
00720  0020 60 0 00000  10000          TRA*    0               RETURN
00721  0754 00 0 00000  10000  ALLUND  PXA     0,0
00722  0020 00 0 41003  10011          TRA     *-3
00723  0340 00 0 00747  10001    GRT3  CAS     =9B17
00724  0020 00 0 01003  10011          TRA     *+3
00725  0020 60 0 00000  10000          TRA*    0
00726  0020 00 0 00403  10001          TRA     ERR1
00727  0340 00 0 00750  10001          CAS     =11B17
00730  0020 00 0 00403  10001          TRA     ERR1
00731  0020 00 0 00721  10001          TRA     ALLUND
00732  0020 00 0 00721  10001          TRA     ALLUND

BINARY CARD ID. INV40027
00733  000000000001     10000          *LORG
00734  201400000000     10000
00735  204500000000     10000
00736  377000000000     10000
00737  177000000000     10000
00740  000000000002     10000
00741  000000000003     10000
00742  000000000004     10000
```

```
00743  000000000012      10000
00744  000000000024      10000
00745  000000000000      10000
00746  000003000000      10000
00747  000011000000      10000
00750  000013000000      10000
                 00000  01111        END
```

```
INV4S                                                                    08/17/65
CONTROL DICTIONARY

        $CDICT INV4S                                                                    INV40028

BINARY CARD ID. INV40029
        000751000000              PREFACE     START=0,LENGTH=489,TYPE=7094,CMPLX=5
        000004000005
        314565046260      INV4S   DECK        LOC=0,LENGTH=489
        000751000000
        314565046260      INV4S   REAL        LOC=0,LENGTH=0
        000000000000

        $DKEND INV4S                                                                    INV40030
```

```
      SUBROUTINE INVBB

C     FOR WING BODY CASE, PREPARES AERODYNAMIC INFLUENCE MATRIX FOR
C     MATRIX REDUCTION

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE

      DIMENSION ABB(102,102),AWB(100)


C     READ AERODYNAMIC INFLUENCE MATRIX (INFLUENCE ON BODY DUE TO BODY
C     AND INFLUENCE ON WING DUE TO BODY) INTO CORE

      DO 100 J=1,NBODY                                                  ,1
      READ (NTAPEA) (ABB(I,J),I=1,NBODY),(AWB(I   ),I=1,NWING)

                                                                        ,2    ,3    ,4    ,5    ,6    ,7
                                                                        ,8    ,9
      WRITE (NTAPED) (AWB(I),I=1,NWING)                                 ,10   ,11   ,12   ,13   ,14
  100 CONTINUE


C     INVERT AERODYNAMIC INFLUENCE MATRIX (INFLUENCE ON BODY DUE TO BODY)
                                                                        ,15   ,16
      CALL SINVRT(ABB,102,NBODY,IRR1,IRR2,SCALE,DET,NDETXP)             ,17
      IF (IRR1) 150,200,150                                             ,18
  150 CONTINUE                                                          ,19
      WRITE (NTAPEO,6000) IRR1,IRR2,SCALE                               ,20   ,21   ,22
      REWIND NTAPEA                                                     ,23
      STOP


                                                                        ,24
  200 CONTINUE                                                          ,25
      DO 300 J=1,NBODY                                                  ,26
      WRITE (NTAPED) (ABB(I,J),I=1,NBODY)                               ,27   ,28   ,29   ,30   ,31
 300  CONTINUE                                                          ,32   ,33
      REWIND NTAPED

                                                                        ,34
 6000 FORMAT(1H1,33HERROR IN INVERSION OF BODY MATRIX,5X,
     16HIRR1 =,I3,5X,6HIRR2 =,I3,5X,7HSCALE =,E12.6)

      RETURN                                                            ,35
      END                                                               ,36
```

```
      SUBROUTINE INVRW

C     INVERT REDUCED MATRIX

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF
     1,NTAPEI,NTAPEO,NBODY,NWING,XMACH,SYM,KACE

      DIMENSION ARWW(102,102)

      DO 700 J=1,NWING                                                  ,1
      READ (NTAPEE) (ARWW(I,J),I=1,NWING)                               ,2    ,3    ,4    ,5    ,6
  700 CONTINUE                                                          ,7    ,8
      END FILE NTAPEE

C     MATRIX INVERSION                                                  ,9
      CALL SINVRT(ARWW,102,NWING,IRR1,IRR2,SCALE,DET,NDETXP)            ,10
      IF (IRR1) 750,800,750                                             ,11
  750 CONTINUE                                                          ,12
      WRITE (NTAPEO,6010) IRR1,IRR2,SCALE                               ,13   ,14   ,15
      REWIND NTAPEE                                                     ,16
      STOP
                                                                        ,17
  800 CONTINUE                                                          ,18
      DO 850 J=1,NWING                                                  ,19
      WRITE (NTAPEE) (ARWW(I,J),I=1,NWING)                              ,20   ,21   ,22   ,23   ,24
  850 CONTINUE                                                          ,25   ,26
      END FILE NTAPEE                                                   ,27
      REWIND NTAPEE
                                                                        ,28
 6010 FORMAT(1H1,36HERROR IN INVERSION OF REDUCED MATRIX,5X,
     16HIRR1 =,I3,5X,6HIRR2 =,I3,5X,7HSCALE =,E12.6)

      RETURN                                                            ,29
      END                                                               ,30
```

```
      SUBROUTINE INVW

C     FOR WING ONLY CASE, STORES MATRIX (A) AND (A) INVERSE ON TAPE

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPED,NBODY,NWING,XMACH,SYM,KACE

      DIMENSION AWW(102,102)

C     READ (A) MATRIX INTO CORE, WRITE ON TAPE

      DO 100 J=1,NWING                                                  ,1
      READ (NTAPEA) (AWW(I,J),I=1,NWING)                                ,2    ,3    ,4    ,5    ,6
      WRITE (NTAPEE) (AWW(I,J),I=1,NWING)                               ,7    ,8    ,9    ,10   ,11
  100 CONTINUE                                                          ,12   ,13
      END FILE NTAPEE                                                   ,14
      REWIND NTAPEA

C     INVERT MATRIX (A)

      CALL SINVRT(AWW,102,NWING,IRR1,IRR2,SCALE,DET,NDETXP)             ,15
      IF (IRR1) 150,200,150                                             ,16
  150 WRITE (NTAPED,6000) IRR1,IRR2,SCALE                               ,17
      REWIND NTAPEE                                                     ,18   ,19   ,20
      STOP                                                              ,21

  200 CONTINUE                                                          ,22
      DO 250 J=1,NWING                                                  ,23
      WRITE (NTAPEE) (AWW(I,J),I=1,NWING)                               ,24
  250 CONTINUE                                                          ,25   ,26   ,27   ,28   ,29
      END FILE NTAPEE                                                   ,30   ,31
      REWIND NTAPEE                                                     ,32

 6000 FORMAT(1H1,38HERROR IN INVERSION OF WING ONLY MATRIX              ,33
     1,5X,6HIRR1 =,I3,5X,6HIRR2 =,I3,5X,7HSCALE =,E12.6)

      RETURN                                                            ,34
      END                                                               ,35
```

```
CIPACK      MICHAEL SYNGE          6-9247       COMPILED   3 SEP 64
C  TO PACK A BUFFER BY MOVING ALL FULL ARRAYS TO THE BEGINNING
C  AND CONDENSING ALL THE EMPTY ARRAYS INTO ONE AT THE END.
C
      FUNCTION IPACK(BUF)
      DIMENSION BUF(1)
C
C              N       - TOTAL LENGTH OF BUFFER
C              M       - FIRST CELL OF CURRENT ARRAY
C              J       - TOTAL NUMBER OF FREE CELLS
C
      IF(ICODE(BUF)-4)90,2,90
    2 N=-IBFR(BUF(3))
      IF(N)80,80,3
    3 M=4
      J=0
    5 L=LENG(BUF(M-1))
      IF (ICODE(BUF(M-2)))10,40,10
   10 IF (J)50,50,20
C
C              MOVE UP FULL ARRAY
C
   20 I1=M-2-J
      I2=I1+L+1
      K=M-2
      DO 30I=I1,I2
      BUF(I)=BUF(K)
   30 K=K+1
      BUF(I1)=ARFUL(I1+2)
      CALL UPDATE(BUF(I1+1))
      GO TO 50
   40 J=J+L+2
   50 IF (N-L-M)100,100,70
   70 M=M+L+2
      GO TO 5
  100 IF (J)1000,1000,110
C
C              SET UP FINAL (EMPTY) ARRAY
C
  110 I1=N-J+1
      IPACK=J-2
      BUF(I1)=ARFRE(I1+2)
      BUF(I1+1)=ARSEC(IPACK,IPACK)
      GO TO 1000
C
C              ERRORS
C
   80 IPACK=-2
      GO TO 1000
   90 IPACK=-1
 1000 RETURN
C
      END
```

```
CIRLEAS     MICHAEL SYNGE          6-9247       COMPILED  21 SEP 64
C              TO RELEASE ALL EXCEPT THE                                IRLE0030
C              FIRST K WORDS OF AN ARRAY                                IRLE0040
C                                                                       IRLE0060
      FUNCTION IRLEAS(IRRAY,K)                                          IRLE0070
      DIMENSION IRRAY(1)                                                IRLE0080
C                                                                       IRLE0100
C              M       - INITIAL LENGTH OF THE ARRAY                    IRLE0110
C              K       - THE NUMBER OF WORDS TO BE RETAINED             IRLE0120
C                                                                       IRLE0130
      N=K+2                                                             IRLE0140    ,1
      IF(N-2)70,70,5                                                    IRLE0150    ,2
    5 I=32767                                                           IRLE0160    ,3
      IF (ICODE(IRRAY(I)))90,50,10                                      IRLE0170    ,4
   10 M=LENG(IRRAY(I+1))                                                IRLE0180    ,5
      IF (M-N)50,20,20                                                  IRLE0190    ,6
   20 IRRAY(I+1)=IRRAY(I+1)-262144*(M-N+2)                                           ,7
      IRRAY(N-1)=IRFRE(N-IBFR(IRRAY))                                   IRLE0210    ,8
      IRRAY(N)=IRSEC(M-N,IRLEAS)                                                    ,9
   50 IRLEAS=0                                                          IRLE0230    ,10
      GO TO 100                                                         IRLE0240    ,11
   70 IRLEAS=IDLETE(IRRAY)                                              IRLE0250    ,12
      GO TO 100                                                         IRLE0260    ,13
   90 IRLEAS=-1                                                         IRLE0270    ,14
  100 RETURN                                                            IRLE0280
C                                                                       IRLE0290
C                                                                       IRLE0300    ,15
      END                                                               IRLE0310    ,16
```

```
CIRSERV     MICHAEL SYNGE          6-9247       COMPILED  21 AUG 64
C  TO RESERVE AN ARRAY OF N CELLS IN A BUFFER. THE ARRAY CAN          IRSE0030
C  SUBSEQUENTLY BE REFERRED TO AS  BUF(NAME) .                         IRSE0040
C                                                                     IRSE0050
C                                                                     IRSE0070
      FUNCTION   IRSERV(N,BUF,NAME)                                   IRSE0080
C                                                                     IRSE0090
      M=-N                                                            IRSE0100    ,1
      IRSERV=IADARY(M,M,BUF,NAME)                                     IRSE0110    ,2
      RETURN                                                          IRSE0120
C                                                                     IRSE0130    ,3
      END                                                             IRSE0140    ,4
```

```
      SUBROUTINE KARMOR (NBODYS,NTHETA,CPCALC,VOUT,ZA,ARA,XP,ZP,XB,R,THE
     1TAB,ZDELTA,XC,YC,ZC,THETA,U,VV,WW,AN1,CPBB,BBCL,BBCD,BBCM)

C     CALCULATION OF VELOCITY COMPONENTS DUE TO BODY THICKNESS
C     BY KARMAN MOORE SOURCE METHOD

      ACOSH(U)=2.*ALOG(SQRT(ABS((U+1.)/2.))+SQRT(ABS((U-1.)/2.)))

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE


      DIMENSION XB(1),R(1),THETAB(1),ZDELTA(1),XC(1),YC(1),ZC(1)
     1,U(1),VV(1),WW(1),AN1(1),CPBB(51,10),THETA(1)

      DIMENSION ALPHAC(51),DTHETA(10),T(51),TC(51),TX(51)
     1,UB(51,10),VB(51,10),VTB(51,10)

      PI=3.1415926                                                      ,1
      BT2=XMACH*XMACH-1.                                                ,2
      BETA=SQRT(BT2)                                                    ,3
      N1=NBODYS-1                                                       ,4
      DO 60 J=1,N1                                                      ,5
   60 ALPHAC(J)=ARA-(ZDELTA(J+1)-ZDELTA(J))/(XB(J+1)-XB(J))             ,6    ,7
      DO 80 J=1,NTHETA                                                  ,8
   80 THETAB(J)=THETAB(J)/57.2957795


C     CALCULATION OF THE STRENGTH OF LINE SOURCES (T) AND DOUBLETS (TC) ,9    ,10
      DO 160 I=1,N1                                                     ,11
      RR=R(I+1)**2*ALPHAC(I)                                            ,12
      RDRDX=R(I+1)*((R(I+1)-R(I))/(XB(I+1)-XB(I)))                      ,13
      RS=RR                                                             ,14
      RDRDXS=RDRDX                                                      ,15
      DO 155 J=1,I                                                      ,16
      TRM1=XB(I+1)-XB(J)+BETA*R(J)                                      ,17
      TRM2=BETA*R(I+1)                                                  ,18
      TRM3=SQRT(TRM1**2-TRM2**2)                                        ,19
      TRM4=ACOSH(TRM1/TRM2)                                             ,20
      RN=TRM3+RDRDX*TRM4                                                ,21
      RM=RDRDX*TRM3+.5*(TRM2**2*TRM4+TRM1*TRM3)                         ,22
      IF (J-I) 150,140,150                                              ,23
  140 CONTINUE                                                          ,24
      T(J)=RDRDXS/RN                                                    ,25
      TC(J)=RS/RM                                                       ,26
      TX(J)=XB(J)-BETA*R(J)                                             ,27
      GO TO 160                                                         ,28
  150 RDRDXS=RDRDXS-RN*T (J)                                            ,29
  155 RS=RS-RM*TC(J)                                                    ,30   ,31
  160 CONTINUE
```

```
C     CALCULATION OF VELOCITY COMPONENTS ON THE BODY DUE TO THE BODY          ,32   ,33
      DO 220 J=1,NTHETA                                                       ,34
      DO 210 N=2,NBODYS                                                       ,35
      UB(N,J)=0.                                                              ,36
      VB(N,J)=0.                                                              ,37
      VTB(N,J)=0.                                                             ,38
      BR2=(BETA*R(N))**2                                                      ,39
      DO 210 K=1,N1                                                           ,40
      RPM=(XB(N)-TX(K))/SQRT(BR2)                                             ,41
      IF (RPM-1.) 210,210,200                                                 ,42
  200 CONTINUE                                                                ,43
      RPMS=SQRT(RPM**2-1.)                                                    ,44
      VB(N,J)=VB(N,J)+T(K)*BETA*RPMS-TC(K)*COS(THETAB(J))*.5*BT2*(ACOSH(
     1RPM)+RPM*RPMS)                                                          ,45
      VTB(N,J)=VTB(N,J)+TC(K)*SIN(THETAB(J))*.5*BT2*(ACOSH(RPM)-RPM*RPMS
     1)                                                                       ,46
      UB(N,J)=UB(N,J)-T(K)*ACOSH(RPM)+TC(K)*COS(THETAB(J))*RPMS*BETA          ,47
  210 CONTINUE                                                                ,48   ,49   ,50
      VB(1,J)=VB(2,J)                                                         ,51
      VTB(1,J)=VTB(2,J)                                                       ,52
      UB(1,J)=UB(2,J)                                                         ,53
  220 CONTINUE

C     CALCULATION OF VELOCITY COMPONENTS ON THE WING DUE TO THE BODY
                                                                              ,54   ,55
      IF (NWING) 230,280,230                                                  ,56
  230 CONTINUE                                                                ,57
      NS=NBODY+1                                                              ,58
      NPANEL=NBODY+NWING                                                      ,59
      II=0                                                                    ,60
      DO 260 I=NS,NPANEL                                                      ,61
      II=II+1                                                                 ,62
      U(II)=0.                                                                ,63
      THETAA=ATN1(YC(I),ZC(I))                                                ,64
      COSTHA=COS(THETAA)                                                      ,65
      SINTHA=SIN(THETAA)                                                      ,66
      R2=YC(I)**2+ZC(I)**2                                                    ,67
      BR2=BT2*R2                                                              ,68
      V=0.                                                                    ,69
      VT=0.                                                                   ,70
      DO 250 K=1,N1                                                           ,71
      RPM=(XC(I)-TX(K))/SQRT(BR2)                                             ,72
      IF (RPM-1.0) 250,250,240                                                ,73
  240 CONTINUE                                                                ,74
      RPMS=SQRT(RPM**2-1.)                                                    ,75
      U(II)=U(II)-T(K)*ACOSH(RPM)+TC(K)*COSTHA*RPMS*BETA                      ,76
      V=V+T(K)*BETA*RPMS-TC(K)*COSTHA*.5*BT2*(ACOSH(RPM)
     1+RPM*RPMS)                                                              ,77
      VT=VT+TC(K)*SINTHA*.5*BT2*(ACOSH(RPM)-RPM*RPMS)                         ,78
  250 CONTINUE
                                                                              ,79   ,80
      THETAS=THETA(I)+THETAA                                                  ,81
      VV(II)=V*SINTHA+VT*COSTHA                                               ,82
      WW(II)=V*COSTHA-VT*SINTHA                                               ,83
```

```
  260 AN1(II)=V*COS(THETAS)-VT*SIN(THETAS)                          ,84   ,85
  280 CONTINUE
                                                                    ,86
      NT=NTHETA-1                                                   ,87
      DTHETA(1)=(THETAB(2)+THETAB(1))/2.                            ,88
      DTHETA(NTHETA)=PI-(THETAB(NTHETA)+THETAB(NT))/2.              ,89
      DO 310 J=2,NT                                                 ,90
  310 DTHETA(J)=(THETAB(J+1)-THETAB(J-1))/2.

C     CALCULATION OF PRESSURES, FORCES, AND MOMENTS                 ,91   ,92
      BBCL=0.                                                       ,93
      BBCD=0.                                                       ,94
      BBCM=0.                                                       ,95
      DO 360 J=1,NTHETA                                             ,96
      DO 350 I=2,NBODYS                                             ,97
      IF (CPCALC) 330,320,330                                       ,98
  320 CONTINUE                                                      ,99
      CPBB(I,J)=-2.*UB(I,J)                                         ,100
      GO TO 340                                                     ,101
  330 CONTINUE                                                      ,102
      CPBB(I,J)=-2.*UB(I,J)+BT2*UB(I,J)**2-VTB(I,J)**2-VB(I,J)**2   ,103
  340 CONTINUE                                                      ,104
      XD=CPBB(I,J)*(R(I)**2-R(I-1)**2)*DTHETA(J)/2.                 ,105
      XL=-CPBB(I,J)*(XB(I)-XB(I-1))*(R(I)+R(I-1))*DTHETA(J)*COS(THETAB(J
     1))/2.                                                         ,106
      XM=XD*(R(I)*COS(THETAB(J))-ZP)-XL*(XB(I)-XP)                  ,107
      BBCL=BBCL+XL                                                  ,108
      BBCD=BBCD+XD                                                  ,109
      BBCM=BBCM+XM                                                  ,110
  350 CONTINUE                                                      ,111  ,112
      CPBB(1,J)=CPBB(2,J)                                           ,113
  360 CONTINUE
                                                                    ,114  ,115
      DO 590 J=1,NTHETA                                             ,116
  590 THETAB(J)=THETAB(J)*57.2957795

C     OPTIONAL PRINT OF VELOCITY COMPONENTS                         ,117  ,118
      IF (VOUT) 600,999,600                                         ,119
  600 CONTINUE                                                      ,120
      WRITE (NTAPEO,6010)                                           ,121  ,122
      WRITE (NTAPEO,6025)                                           ,123  ,124
      WRITE (NTAPEO,6015) (THETAB(J),J=1,NTHETA)                    ,125  ,126  ,127  ,128  ,129
      WRITE (NTAPEO,6020)                                           ,130  ,131
      DO 620 I=1,NBODYS                                             ,132
  620 WRITE (NTAPEO,6000) (XB(I),(UB(I,J),J=1,NTHETA))              ,133  ,134  ,135  ,136  ,137  ,138
                                                                    ,139
      WRITE (NTAPEO,6030)                                           ,140  ,141
      WRITE (NTAPEO,6015) (THETAB(J),J=1,NTHETA)                    ,142  ,143  ,144  ,145  ,146
      WRITE (NTAPEO,6020)                                           ,147  ,148
      DO 630 I=1,NBODYS                                             ,149
  630 WRITE (NTAPEO,6000) (XB(I),(VB(I,J),J=1,NTHETA))              ,150  ,151  ,152  ,153  ,154  ,155
```

```
                                                                          ,156
      WRITE (NTAPEO,6035)                                                 ,157  ,158
      WRITE (NTAPEO,6015) (THETAB(J),J=1,NTHETA)                          ,159  ,160  ,161  ,162  ,163
      WRITE (NTAPEO,6020)                                                 ,164  ,165
      DO 640 I=1,NBODYS                                                   ,166
  640 WRITE (NTAPEO,6000) (XB(I),(VTB(I,J),J=1,NTHETA))
                                                                          ,167  ,168  ,169  ,170  ,171  ,172
                                                                          ,173
 6000 FORMAT(1H ,F14.4,F15.5,9F10.5,(F29.5,9F10.5))
 6010 FORMAT(1H0//66H VELOCITY COMPONENTS ON BODY DUE TO BODY LINE SOURC
     1ES AND DOUBLETS)
 6015 FORMAT(1H0,4X,11HTHETA(DEG.),F14.4,9F10.4,(F29.4,9F10.4))
 6020 FORMAT(1H ,9X,1HX)
 6025 FORMAT(1H0,5X,8HAXIAL(U))
 6030 FORMAT(1H0,5X,10HRADIAL(VR))
 6035 FORMAT(1H0,5X,14HTANGENTIAL(VT))

  999 CONTINUE                                                            ,174
      RETURN                                                              ,175
      END                                                                 ,176
```

```
LENG                                                                    07/31/65
7094 RELMOD ASSEMBLY.

        $IBLDR LENG                                                                  LENG0000
ASSEMBLED TEXT.

        $TEXT  LENG                                                                  LENG0001

                                    *LENG       MICHAEL SYNGE          6-9247      ASSEMBLED  27 AUG 64
                                    *     TO OBTAIN THE CONTENTS OF THE DECREMENT OF A
                                    *     CODE WORD AS A FULL WORD INTEGER
                                    *
                                    *              CALLING SEQUENCE
                                    *
                                    *                  N=LENG(CODE)
                                    *
                                           ENTRY   LENG
                                    *

BINARY CARD (NOT PUNCHED)
      00000  0500 60 4 00003  10000   LENG  CLA*    3,4
      00001  4320 00 0 00004  10001         ANA     =077777000000
      00002  0771 00 0 00022  10000         ARS     18
      00003  0020 00 4 00001  10000         TRA     1,4
                                    *
      00004  077777000000     10000         *LORG
                      00000  01111         END
         CONTROL DICTIONARY

        $CDICT LENG                                                                  LENG0002

BINARY CARD (NOT PUNCHED)
            000005000000              PREFACE     START=0,LENGTH=5,TYPE=7094,CMPLX=5
            000004000005
            432545276060       LENG   DECK        LOC=0,LENGTH=5
            000005000000
            432545276060       LENG   REAL        LOC=0,LENGTH=0
            000000000000

        $DKEND LENG                                                                  LENG0003

NO MESSAGES FOR THIS ASSEMBLY
         SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

  CLASS  SYMBOL   VALUE   REFERENCES

         LENG     00000
  LCTR    BLCTR
  QUAL    UNQS
  LCTR   //
```

```
      SUBROUTINE MDMATE

C     FORMATION OF DRAG MINIMIZATION MATRIX

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE

      DIMENSION WW(110,110),XBAR(200),AREA(200)

      NPANEL=NBODY+NWING                                                ,1
      READ (NTAPEC) (I,XBAR(I),DUMMY1,DUMMY2,DUMMY3,DUMMY4,DUMMY5
     1,AREA(I),DUMMY6,DUMMY7,DUMMY8,I=1,NPANEL)                        ,2    ,3    ,4    ,5    ,6    ,7

      REWIND NTAPEC
                                                                        ,8
      DO 100 J=1,NWING                                                  ,9
      READ (NTAPEE) (WW(I,J),I=1,NWING)                                 ,10   ,11   ,12   ,13   ,14
 100  CONTINUE                                                          ,15   ,16
      REWIND NTAPEE
                                                                        ,17
      DO 200 I=1,NWING                                                  ,18
      I=I                                                               ,19
      II=I+NBODY                                                        ,20
      DO 200 J=1,I                                                      ,21
      JJ=J+NBODY                                                        ,22
      WW(I,J)=-WW(I,J)*AREA(II)-WW(J,I)*AREA(JJ)                        ,23
      WW(J,I)=WW(I,J)                                                   ,24
 200  CONTINUE
                                                                        ,25   ,26   ,27
      NN=NWING+1                                                        ,28
      NNN=NWING+2                                                       ,29
      DO 300 J=1,NWING                                                  ,30
      JJ=J+NBODY                                                        ,31
      WW(J,NN)=-AREA(JJ)                                                ,32
      WW(NN,J)=-AREA(JJ)                                                ,33
      XA=-XBAR(JJ)*AREA(JJ)                                             ,34
      WW(J,NNN)=XA                                                      ,35
      WW(NNN,J)=XA                                                      ,36
 300  CONTINUE                                                          ,37   ,38
      WW(NN,NN)=0.0                                                     ,39
      WW(NN,NNN)=0.0                                                    ,40
      WW(NNN,NN)=0.0                                                    ,41
      WW(NNN,NNN)=0.0
                                                                        ,42
      DO 325 J=1,NNN                                                    ,43
 325  WRITE (NTAPEA)(WW(I,J),I=1,NNN)                                   ,44   ,45   ,46   ,47   ,48   ,49

      END FILE NTAPEA

C     INVERT DRAG MINIMIZATION MATRIX CONSTRAINED FOR CL                ,50
      CALL SINVRT (WW,110,NN,IRR1,IRR2,SCALE,DET,NDETXP)                ,51
      IF (IRR1) 350,400,350                                             ,52
 350  CONTINUE                                                          ,53
```

```
      WRITE (NTAPEO,6000) IRR1,IRR2,SCALE                                ,54   ,55   ,56
      REWIND NTAPEA                                                      ,57
      STOP
                                                                         ,58
  400 CONTINUE                                                           ,59
      DO 450 J=1,NN                                                      ,60
  450 WRITE (NTAPEA)(WW(I,J),I=1,NN)                                     ,61   ,62   ,63   ,64   ,65   ,66

      END FILE NTAPEA                                                    ,67
      REWIND NTAPEA
                                                                         ,68
      DO 475 J=1,NNN                                                     ,69
  475 READ (NTAPEA)(WW(I,J),I=1,NNN)                                     ,70   ,71   ,72   ,73   ,74   ,75

      CALL FSF(2,NTAPEA,IRR)

C     INVERT DRAG MINIMIZATION MATRIX CONSTRAINED FOR CL AND CM          ,76
      CALL SINVRT (WW,110,NNN,IRR1,IRR2,SCALE,DET,NDETXP)                ,77
      IF (IRR1) 350,500,350
                                                                         ,78
  500 CONTINUE                                                           ,79
      DO 550 J=1,NNN                                                     ,80
  550 WRITE (NTAPEA)(WW(I,J),I=1,NNN)                                    ,81   ,82   ,83   ,84   ,85   ,86

      END FILE NTAPEA                                                    ,87
      REWIND NTAPEA

                                                                         ,88
 6000 FORMAT(1H1,46HERROR IN INVERSION OF DRAG MINIMIZATION MATRIX
     1,6HIRR1 =,I3,5X,6HIRR2 =,I3,5X,7HSCALE =,E12.6)

      RETURN                                                             ,89
      END                                                                ,90
```

```
      SUBROUTINE MEAN (X,Y,N)
      DIMENSION X(1),Y(1)

C     SUBROUTINE TO SMOOTH DATA

 10   MC=1                                                          ,1
 20   M3=N-3                                                        ,2
      DO 90 I=1,M3                                                  ,3
      I=I                                                           ,4
      A=(Y(I+2)-Y(I))/(X(I+2)-X(I))                                 ,5
      C=(Y(I+3)-Y(I+1))/(X(I+3)-X(I+1))                             ,6
      B=Y(I)-A*X(I)                                                 ,7
      D=Y(I+1)-C*X(I+1)                                             ,8
      IF (A-C) 40,30,40                                             ,9
 30   IF (B-D) 80,90,80                                             ,10
 40   XINT=(B-D)/(C-A)                                              ,11
      IF (XINT-X(I)) 80,50,50                                       ,12
 50   IF (XINT-X(I+2)) 60,60,80                                     ,13
 60   IF (XINT-X(I+1)) 80,70,70                                     ,14
 70   IF (XINT-X(I+3)) 90,90,80                                     ,15
 80   YF=A*X(I+1)+B                                                 ,16
      Y(I+1)=(Y(I+1)+YF)/2.                                         ,17
      YS=C*X(I+2)+D                                                 ,18
      Y(I+2)=(Y(I+2)+YS)/2.                                         ,19
 90   CONTINUE                                                      ,20    ,21
      MC=MC+1                                                       ,22
      IF (MC-5) 20,20,100                                           ,23
 100  RETURN                                                        ,24
      END                                                           ,25
```

```
      FUNCTION MERR(NU,IOK,M,SR)
      DIMENSION NU(3)
C
C     ERROR MESSAGE WRITER FOR GEOMD.
C     NU(1) IS AN ERROR RETURN FROM SOME SUBROUTINE, AND IT SHOULD
C     EQUAL IOK.  IF SO, MERR = 0 AND RETURN.
C     IF NOT, THE ERROR COUNTER, NU(3), IS DECREASED BY 1 AND A
C     MESSAGE IS WRITTEN IF THE TAPE NO., NU(2), .GT. 0 AND
C     NU(3) .GT. 0.  SR IS ALPHAMERIC NAME OF SUBROUTINE.
C
      MERR = 0
      IF(NU-IOK)100,500,100                                          ,1
  100 LT= NU(2)                                                      ,2
      NU(3)=NU(3)-1                                                  ,3
      IF((LT .LE. 0) .OR. (NU(3) .LE. 0)) GO TO 400                  ,4
      WRITE (LT,200) M,NU(1),SR                                      ,5    ,6    ,7
  200 FORMAT( 6H0ERROR I6,6H, CODE I6,15H IN SUBROUTINE A6,          ,8    ,9    ,10
     1 28H DURING GEOMETRIC DEFINITION)
      MERR=1                                                         ,11
      GO TO 500                                                      ,12
  400 MERR=-1                                                        ,13
  500 RETURN                                                         ,14
      END                                                            ,15
```

```
      FUNCTION NWEED(A,N,J,EP)
      DIMENSION A(J,N),EP(1)
C
C     A IS AN ARRAY OF N POINTS IN J SPACE.
C     ELIMINATE ANY POINT WHICH IS COINCIDENT OR NEARLY SO (WITHIN EP)
C        WITH THE PRECEDING POINT.
C     SHIFT REMAINING POINTS TO TOP OF ARRAY.
C     THE LAST POINT IN A AT INPUT WILL BE THE LAST POINT AT OUTPUT.
C     NWEED = NO. OF POINTS IN A AT OUTPUT.
C
      IA=1                                                              ,1
      DO1000 IB=2,N
C                                                                       ,2
      DO 100 IC=1,J                                                     ,3
      IF (ABS(A(IC,IA)-A(IC,IB))-EP(IC))100,300,300                     ,4
 100  CONTINUE
C                                                                       ,5    ,6
 200  IF (N-IB)400,400,1000                                             ,7
 300  IA=IA+1
C                                                                       ,8
 400  DO 500 IC=1,J                                                     ,9
 500  A(IC,IA)=A(IC,IB)
C                                                                       ,10   ,11
 1000 CONTINUE
C                                                                       ,12   ,13
      NWEED=IA                                                          ,14
      RETURN                                                            ,15
      END                                                               ,16
```

```
C       CONTROL LINK
        DIMENSION DICT(7)
        COMMON DAT(2),LA,LB,LC,LD,LE,LF,LI,LO,
       1 NBODY,NWING,XMACH,SYM,KACE
       2 /LGEOMD/LGDEF(3,6)
        LOGICAL LGDEF
        DATA DICT/6HDEFINE,6HDEFINE,5HPANEL,6HAERODY,6HAERODY,
       1 6HEND OF,6HEND OF/
C
C       INITIALIZE
        CALL OPCAMI
C
C       READ COMMAND CARD
C
 100    IGO=INTURP(DICT,7,LI,LO)                                          ,1
C
C       THE FOLLOWING THREE STATEMENTS CAN BE REMOVED IF THE TIME AND DATE
C       OPPOSITE EACH COMMAND ARE NOT WANTED OR IF AN APPROPRIATE
C       SUBROUTINE CLOK IS NOT AVAILABLE.  OR ELSE A DUMMY SUBROUTINE CLOK
C       WHICH RETURNS BLANKS FOR TIM AND DAYT CAN BE USED.                ,2
        CALL CLOK(TIM,DAYT)                                               ,3
        WRITE(LO,105)TIM,DAYT                                             ,4    ,5    ,6
 105    FORMAT(1H+ 101X,4HTIME 2A7)

        IF (IGO)100,100,150
C                                                                         ,7
 150    GO TO (200,200,300,400,400,500,500  ),IGO
C
C       WHEN A NEW COMMAND IS NEEDED,
C         INCREASE DIMENSION OF DICT BY 1
C         ADD COMMAND TO DATA STATEMENT
C         INCREASE 2ND ARGUEMENT OF INTURP BY 1
C         ADD THE NEW STATEMENT NO. TO THE   GO TO   AT 150
C         ADD THE STATEMENT, FOLLOW WITH   GO TO 100
C
C       A WING AND/OR BODY ARE TO BE DEFINED                              ,8
 200    CALL GEOMD(DAT,LI,LO,LD,LA)                                       ,9
        GO TO 100
C
C       CALL PANEL LINK
C       TRANSFORM BODY TO NEW COORD. SYSTEM, FLATTEN WING, INTERSECT BODY
C       AND WING.  THEN CALL PANEL LINK.                                  ,10
  300   CALL TFLAT(DAT,LI,LO,LA,LB,LD,NBODY,NU)                           ,11
        IF(NU)100,310,100
C       NBODY WAS SET TO -1 BY TFLAT IF THIS IS A BODY-ONLY CASE.
C         THEN WE DO NOT CALL THE PANEL LINK.  NBODY = 0 OTHERWISE.       ,12
  310   IF(NBODY+1)320,100,320
C       IF TFLAT DETECTS AN ERROR, IT READS DATA CARDS UNTIL IT
C         HITS AN   END OF DATA   CARD OR A   DEFINE   CARD.
C         THEN THE INPUT TAPE IS BACKSPACED 1 RECORD.                     ,13
  320   CALL PANEL                                                        ,14
        GO TO 100
C
```

```
EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT    -   INTERNAL FORMULA NUMBER(S)

C     CALL AERODYNAMIC LINK                                                  ,15
 400  CALL AERO                                                              ,16
      GO TO 100
C
C     END OF  OR  END OF  CARD WAS READ                                      ,17
 500  STOP                                                                   ,18
      END                                                                    ,19
```

```
C     INITIALIZATION FOR OPCAM
C
      SUBROUTINE OPCAMI
C
      COMMON DAT(2),LA,LB,LC,LD,LF,LF,LI,LO,
     1 NBODY,NWING,XMACH,SYM,KACE
C
      DIMENSION ICARD(14),ICODE(2)
C
      DATA  ICODE/6HEND OF,6HEND OF/
C
C     ASSIGN TAPE NUMBERS.  TAPES 1-4 AND 7-9 ARE SCRATCH TAPES.
C     LIN IS THE SYSTEM INPUT TAPE.  DATA IS TRANSFERRED FROM LIN TO LI,
C       AS SHOWN BELOW.  LO IS THE SYSTEM OUTPUT TAPE.
      LA=1                                                              ,1
      LB=2                                                              ,2
      LC=3                                                              ,3
      LD=4                                                              ,4
      LIN=5                                                             ,5
      LO=6                                                              ,6
      LE=7                                                              ,7
      LF=8                                                              ,8
      LI=9
C
      CALL DATE(DAT,DAT(2))                                             ,9
      WRITE(LO,100)DAT                                                  ,10
  100 FORMAT(1H1 56X 2A6/69H PROGRAM FOR ANALYSIS AND DESIGN OF SUPERSON ,11  ,12  ,13
     .IC WING-BODY COMBINATIONS/
     .28HODATA CARDS ARE LISTED BELOW//)
C
C     READ DATA CARDS FROM TAPE  LIN, WRITE THEM ON TAPE  LO
C     (AS PART OF JOB PRINTOUT), AND ALSO WRITE THEM ON TAPE  LI
C     WHICH WILL BE USED AS THE INPUT TAPE FOR REST OF JOB.
      REWIND LI
      DO 1000 I=1,10000                                                 ,14
      READ (LIN,200) ICARD                                              ,15
  200 FORMAT(13A6,A2)                                                   ,16  ,17  ,18
      WRITE (LI,200) ICARD
      WRITE (LO,300) ICARD                                              ,19  ,20  ,21
  300 FORMAT(1H 14A6)                                                   ,22  ,23  ,24
      DO 400 J=1,2
      IF(ICARD-ICODE(J)) 400,2000,400                                   ,25
  400 CONTINUE                                                          ,26
 1000 CONTINUE                                                          ,27  ,28
C
 2000 CONTINUE                                                          ,29  ,30
      END FILE LI                                                       ,31
      REWIND LI                                                         ,32
      WRITE(LO,5000) I                                                  ,33
 5000 FORMAT( 1H0 5X 10HA TOTAL OF I5,22H  DATA CARDS WERE READ/1H1)    ,34  ,35  ,36
C
      RETURN
      END                                                               ,37
                                                                        ,38
```

```
C                                                                       OPTI0030
      SUBROUTINE OPTIM3(X,Y,ZEE,A,B,K,L,M)                              OPTI0040
C                                                                       OPTI0050
C          INTERPOLATES BETWEEN GIVEN POINTS, CHOOSING AND INSERTING    OPTI0060
C          IN EACH INTERVAL THE NECESSARY AND SUFFICIENT NUMBER OF      OPTI0070
C          INTERMEDIATE POINTS TO MAKE THE MAXIMUM DISTANCE FROM CHORD  OPTI0080
C          TO ARC NO MORE THAN A GIVEN CONSTANT, Z.  THE FUNCTION USED  OPTI0090
C          HAS A CONTINUOUS FIRST DERIVATIVE.                           OPTI0100
C                                                                       OPTI0110
      DIMENSION X(4),Y(4),A(200),B(200),C(3,2),S(3)                     OPTI0120
C                                                                       OPTI0140
      EPS=.002                                                          OPTI0150   ,1
      CALL DVCHK (K000FX)                                               OPTI0160   ,2
       GO TO(1,1),K000FX                                                OPTI0170   ,3
    1 IF (ZEE) 2,2,3                                                    OPTI0180   ,4
    2 Z=EPS*ABS(X(K+1)-X(K))                                            OPTI0190   ,5
      GO TO 4                                                           OPTI0200   ,6
    3 Z=ZEE                                                             OPTI0210   ,7
    4 KEY=0                                                             OPTI0220   ,8
      MU=0                                                              OPTI0230   ,9
      NU=0                                                              OPTI0240   ,10
      IF (X(1)-X(4)) 12,5,10                                            OPTI0250   ,11
    5 M=0                                                               OPTI0260   ,12
      GO TO 100                                                         OPTI0270   ,13
   10 MU=1                                                              OPTI0280   ,14
      DO 11 I=1,4                                                       OPTI0290   ,15
   11 X(I)=-X(I)                                                        OPTI0300   ,16   ,17
   12 I=1                                                               OPTI0310   ,18
      J=2                                                               OPTI0320   ,19
      IF (K-2) 13,15,14                                                 OPTI0330   ,20
   13 J=1                                                               OPTI0340   ,21
      GO TO 19                                                          OPTI0350   ,22
   14 I=2                                                               OPTI0360   ,23
      GO TO 19                                                          OPTI0370   ,24
   15 DO 18 N=1,3                                                       OPTI0380   ,25
      S(N)=(Y(N+1)-Y(N))/(X(N+1)-X(N))                                  OPTI0390   ,26
      IF (N-2) 18,16,16                                                 OPTI0400   ,27
   16 IF (ABS(S(N)-S(N-1))-EPS) 17,18,18                                OPTI0410   ,28
   17 I=N-1                                                             OPTI0420   ,29
      J=N-1                                                             OPTI0430   ,30
      NU=1                                                              OPTI0440   ,31
      GO TO 19                                                          OPTI0450   ,32
   18 CONTINUE                                                          OPTI0460   ,33   ,34
   19 DO 20 N=I,J                                                       OPTI0470   ,35
      J1=N                                                              OPTI0480   ,36
      J2=N+1                                                            OPTI0490   ,37
      J3=N+2                                                            OPTI0500   ,38
      D=(Y(J2)-Y(J1))/(X(J2)-X(J1))                                     OPTI0510   ,39
      C(3,N)=((Y(J3)-Y(J1))/(X(J3)-X(J1))-D)/(X(J3)-X(J2))              OPTI0520   ,40
      C(2,N)=D-C(3,N)*(X(J2)+X(J1))                                     OPTI0530   ,41
   20 C(1,N)=Y(J1)-X(J1)*C(2,N)-X(J1)**2*C(3,N)                         OPTI0540   ,42   ,43
      DX=X(3)-X(2)                                                      OPTI0550   ,44
      D=0.0                                                             OPTI0560   ,45
```

```
      EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      E=X(K)                                                          OPTI0570    ,46
      DO 85 I=1,200                                                   OPTI0580    ,47
      L=I                                                             OPTI0590    ,48
      E=E+D                                                           OPTI0600    ,49
      IF (E+.01*(X(K+1)-X(K))-X(K+1)) 40,35,35                        OPTI0610    ,50
   35 L=L-1                                                           OPTI0620    ,51
      GO TO 90                                                        OPTI0630    ,52
   40 A(L)=E                                                          OPTI0640    ,53
      IF (NU) 45,42,45                                                OPTI0650    ,54
   42 P=AMIN1(AMAX1((X(3)-E)/DX,0.0),1.0)                             OPTI0660    ,55
      GO TO 50                                                        OPTI0670    ,56
   45 P=2-J                                                           OPTI0680    ,57
   50 Q=1.0-P                                                         OPTI0690    ,58
      F=P*(C(1,1)+E*(C(2,1)+E*C(3,1)))+Q*(C(1,2)+E*(C(2,2)+E*C(3,2)))  OPTI0700    ,59
      B(L)=F                                                          OPTI0710    ,60
      G=P*(C(2,1)+2.0*E*C(3,1))+Q*(C(2,2)+2.0*E*C(3,2))               OPTI0720    ,61
      T=ATAN(G)                                                       OPTI0730    ,62
      W=Z*COS(T)                                                      OPTI0740    ,63
      H=2.0*(P*C(3,1)+Q*C(3,2))                                       OPTI0750    ,64
   60 R=(SQRT(1.0+G**2))**3/ABS(H)                                    OPTI0760    ,65
      CALL DVCHK (K000FX)                                             OPTI0770    ,66
       GO TO(80,65),K000FX                                            OPTI0780    ,67
   65 IF (R-W) 70,70,75                                               OPTI0790    ,68
   70 D=W                                                             OPTI0800    ,69
      GO TO 85                                                        OPTI0810    ,70
   75 V=ACOS(1.0-W/R)                                                 OPTI0820    ,71
      U=T+SIGN(V,H)                                                   OPTI0830    ,72
      D=2.8284*COS(U)*SQRT(R*W)                                       OPTI0840    ,73
      GO TO 85                                                        OPTI0850    ,74
   80 IF (KEY) 90,82,90                                               OPTI0860    ,75
   82 KEY=1                                                           OPTI0870    ,76
      D=.2*(X(K+1)-X(K))                                              OPTI0880    ,77
   85 CONTINUE                                                        OPTI0890    ,78    ,79
      IF (E+D-X(K+1)) 88,90,90                                        OPTI0900    ,80
   88 M=2                                                             OPTI0910    ,81
      GO TO 91                                                                    ,82
   90 M=1                                                             OPTI0930    ,83
   91 B(1)=Y(K)                                                       OPTI0940    ,84
      IF (MU) 92,100,92                                               OPTI0950    ,85
   92 DO 95 I=1,4                                                     OPTI0960    ,86
   95 X(I)=-X(I)                                                      OPTI0970    ,87    ,88
      DO 98 I=1,L                                                     OPTI0980    ,89
   98 A(I)=-A(I)                                                      OPTI0990    ,90    ,91
  100 RETURN                                                          OPTI1000    ,92
      END                                                             OPTI1010    ,93
```

```
      SUBROUTINE OPTMW (NW,NTAPEX,A,B,CONSNT,CLBAR,XCPBAR,RFAREA,AREA,CL
     1)

C     OPTIMIZE WING ONLY CASE

      DIMENSION A(1),B(1),AREA(1),CL(1)

      NN=NW+1                                                           ,1
      B(NN)=CLBAR*RFAREA

C     TEST IF PITCHING MOMENT CONSTRAINT                                ,2
      IF (CONSNT) 100,150,100                                           ,3
  100 CONTINUE                                                          ,4
      NN=NW+2                                                           ,5
      B(NN)=XCPBAR*CLBAR*RFAREA
                                                                        ,6
  150 CONTINUE                                                          ,7
      DO 200 J=1,NW                                                     ,8
  200 CL(J)=0.                                                          ,9    ,10
      DO 350 J=1,NN                                                     ,11
      READ (NTAPEX) (A(I),I=1,NN)                                       ,12   ,13   ,14   ,15   ,16
      IF (J-NW) 350,350,250                                             ,17
  250 CONTINUE                                                          ,18
      DO 300 I=1,NW                                                     ,19
  300 CL(I)=CL(I)+A(I)*B(J)                                             ,20   ,21
  350 CONTINUE
                                                                        ,22   ,23
      RETURN                                                            ,24
      END                                                               ,25
```

```
      SUBROUTINE OPTMWB (NW,NB,NTAPEX,NTAPEY,NTAPEZ,A,B,ALPHA,ALPHAX
     1,AREA,CONSNT,CLBAR,XCPBAR,RFAREA,CL)
C     OPTIMIZE WING-BODY CASE
      DIMENSION A(1),B(1),ALPHA(1),ALPHAX(1),AREA(1),CL(1)
      IF (NTAPEZ) 50,25,50
 25   CONTINUE                                                          ,1
      DO 40 I=1,NB                                                      ,2
 40   B(I)=0.                                                           ,3
      GO TO 75                                                          ,4    ,5
 50   CONTINUE                                                          ,6
      READ (NTAPEZ) (B(I),I=1,NB)                                       ,7
 75   CONTINUE                                                          ,8    ,9    ,10   ,11   ,12
      DO 100 J=1,NW                                                     ,13
 100  ALPHA(J)=ALPHAX(J)                                                ,14
      DO 150 J=1,NB                                                     ,15   ,16
      READ (NTAPEY) (A(I),I=1,NW)                                       ,17
      DO 150 I=1,NW                                                     ,18   ,19   ,20   ,21   ,22
 150  ALPHA(I)=ALPHA(I)-A(I)*B(J)                                       ,23
      DO 200 J=1,NW                                                     ,24   ,25   ,26
 200  B(J)=AREA(J)*ALPHA(J)                                             ,27
      NN=NW+1                                                           ,28   ,29
      B(NN)=CLBAR*RFAREA                                                ,30
C     TEST IF PITCHING MOMENT CONSTRAINT
      IF (CONSNT) 275,325,275                                           ,31
 275  CONTINUE                                                          ,32
      NN=NW+2                                                           ,33
      B(NN)=XCPBAR*CLBAR*RFAREA                                         ,34
 325  CONTINUE                                                          ,35
      DO 350 J=1,NW                                                     ,36
 350  CL(J)=0.                                                          ,37
      DO 400 J=1,NN                                                     ,38   ,39
      READ (NTAPEX) (A(I),I=1,NN)                                       ,40
      DO 400 I=1,NW                                                     ,41   ,42   ,43   ,44   ,45
 400  CL(I)=CL(I)+A(I)*B(J)                                             ,46
      RETURN                                                            ,47   ,48   ,49
      END                                                               ,50
                                                                         51
```

```
      SUBROUTINE OUTB (NTAPEO,NBODY,NTHETA,NROWB,THETA,Z)

C     BODY OUTPUT FORMAT

      DIMENSION Z(1),THETA(1)
      WRITE (NTAPEO,6005) (THETA(I),I=1,NTHETA)                          ,1    ,2    ,3    ,4    ,5
      WRITE (NTAPEO,6010)                                                ,6    ,7
      DO 100 J=1,NROWB                                                   ,8
  100 WRITE (NTAPEO,6015) (J,(Z(I),I=J,NBODY,NROWB))                     ,9    ,10   ,11   ,12   ,13   ,14
                                                                         ,15
 6005 FORMAT(1H0,4X,11HTHETA(DEG.),F14.4,9F10.4,(F29.4,9F10.4))
 6010 FORMAT(1H0,6X,7HROW NO.)
 6015 FORMAT(1H ,I10,F19.5,9F10.5,(F29.5,9F10.5))
      RETURN                                                             ,16
      END                                                                ,17
```

```
      SUBROUTINE OUTPTB
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/ NPLNB,NPLNW,JLEAD,JTRAIL,IMID,NPTS(16),X(16,90),Y(16
     1,90),Z(16,90),XCEPT(21),XCEPTB(21),XCEPTW(16),YCEPTW(16),ZCEPTW(16
     2),CODEBW(16),KPANEL(15,20),XCOR(16,21),YCOR(16,21),ZCOR(16,21),XIN
     3T(15,20,2),YINT(15,20,2),ZINT(15,20,2),XCEN(15,20),YCEN(15,20),ZCE
     4N(15,20),XCON(15,20),YCON(15,20),ZCON(15,20),AREA(15,20),ARAT(15,2
     50),THETA(15,20),ALPHA(15,20),CHORD(15,20)
      DIMENSION C(3,6)
      DIMENSION ND3(6108)
      EQUIVALENCE (ND3,XCOR)

C     SUBROUTINE TO OUTPUT BODY PANEL DATA

C     DEFINE CONSTANTS
      NBODY=0                                                          ,1
      KODEB=0                                                          ,2
      KODEI=0                                                           3
      NUMBER=0                                                         ,4
      NPART1=1                                                         ,5
      NPART2=2                                                          6
      NPART3=3                                                         ,7
      E=SYM                                                            ,8
      F=0.00001                                                        ,9
      REWIND NTAPE2

C     DO-LOOP TO SET SMALL NEGATIVE AND POSITIVE
C     VALUES EQUAL TO ZERO                                             ,10
      DO 20 I=1,6108                                                   ,11
      IF (ABS(ND3(I))-F) 10,10,20                                      ,12
   10 ND3(I)=0.                                                        ,13
   20 CONTINUE

C     WRITE PRINTOUT HEADINGS                                          ,14   ,15
      WRITE (NWRITE,900)                                               ,16   ,17
      WRITE (NWRITE,910)

C     WRITE TAPE OUTPUT - NUMBER IS PANEL NUMBER
C     NPART1,NPART2 AND NPART3 INDICATE 1-, 2- AND
C     3-PART PANELS      XCOR,YCOR,ZCOR
C     ARE PANEL CORNER POINT COORDINATES                               ,18   ,19
 50   DO 130 I=1,NPER1                                                 ,20
      DO 130 J=1,NPLN1                                                 ,21
      KGO=KPANEL(I,J)                                                  ,22
      GO TO (52,53,52,53),KGO

C     SETUP OUTPUT ARRAY                                               ,23
 52   C(1,1)=XCOR(I,J+1)                                               ,24
      C(2,1)=YCOR(I,J+1)                                               ,25
```

```
      C(3,1)=ZCOR(I,J+1)                                                ,26
      C(1,2)=XCOR(I,J)                                                  ,27
      C(2,2)=YCOR(I,J)                                                  ,28
      C(3,2)=ZCOR(I,J)                                                  ,29
      C(1,3)=XINT(I,J,1)                                                ,30
      C(2,3)=YINT(I,J,1)                                                ,31
      C(3,3)=ZINT(I,J,1)                                                ,32
      GO TO 55                                                          ,33
   53 C(1,1)=XCOR(I,J)                                                  ,34
      C(2,1)=YCOR(I,J)                                                   35
      C(3,1)=ZCOR(I,J)                                                  ,36
      C(1,2)=XINT(I,J,1)                                                ,37
      C(2,2)=YINT(I,J,1)                                                ,38
      C(3,2)=ZINT(I,J,1)                                                 39
      C(2,3)=YCOR(I,J+1)                                                ,40
      C(1,3)=XCOR(I,J+1)                                                ,41
      C(3,3)=ZCOR(I,J+1)                                                ,42
   55 GO TO (57,57,58,58),KGO                                           43
   57 C(1,4)=XCOR(I+1,J)                                                ,44
      C(2,4)=YCOR(I+1,J)                                                ,45
      C(3,4)=ZCOR(I+1,J)                                                ,46
      C(1,5)=XINT(I,J,2)                                                ,47
      C(2,5)=YINT(I,J,2)                                                ,48
      C(3,5)=ZINT(I,J,2)                                                ,49
      C(1,6)=XCOR(I+1,J+1)                                              ,50
      C(2,6)=YCOR(I+1,J+1)                                              ,51
      C(3,6)=ZCOR(I+1,J+1)                                              ,52
      GO TO 60                                                          ,53
   58 C(1,4)=XINT(I,J,2)                                                ,54
      C(2,4)=YINT(I,J,2)                                                ,55
      C(3,4)=ZINT(I,J,2)                                                ,56
      C(1,5)=XCOR(I+1,J+1)                                              ,57
      C(2,5)=YCOR(I+1,J+1)                                              ,58
      C(3,5)=ZCOR(I+1,J+1)                                              ,59
      C(1,6)=XCOR(I+1,J)                                                ,60
      C(2,6)=YCOR(I+1,J)                                                ,61
      C(3,6)=ZCOR(I+1,J)
C     CHECK SUBPANELS AGAINST TOLERANCE                                 ,62
   60 KTYPE=1                                                           ,63
      GO TO (63,64,63,64),KGO                                           ,64
   63 JTEMP=J+1                                                         ,65
      GO TO 65                                                          ,66
   64 JTEMP=J                                                            67
   65 SC=SQRT((YCOR(I,JTEMP)-YINT(I,J,1))**2+(ZCOR(I,JTEMP)-
     1ZINT(I,J,1))**2)/(XCOR(I,J+1)-XCOR(I,J))                          ,68
      IF (ABS(SC)-E) 67,67,70                                           ,69
   67 KTYPE=2                                                           ,70
   70 GO TO (73,73,74,74),KGO                                           ,71
   73 JTEMP=J+1                                                          72
      GO TO 75                                                          ,73
   74 JTEMP=J                                                           ,74
   75 SC=SQRT((YCOR(I+1,JTEMP)-YINT(I,J,2))**2+(ZCOR(I+1,JTEMP)-
     1ZINT(I,J,2))**2)/(XCOR(I+1,J+1)-XCOR(I+1,J))                      ,75
```

```
      EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      IF (ABS(SC)-E) 77,77,80                                                  76
77    KTYPE=KTYPE+2                                                           ,77
80    NUMBER=NUMBER+1                                                         ,78
      GO TO (90,100,110,120),KTYPE                                            ,79
90    WRITE (NWRITE,920) NUMBER,NPART3,
     1((C(K,L),K=1,3),L=1,2),((C(K,L),K=1,3),L=1,3,2)                         ,80   ,81   ,82   ,83   ,84   ,85
                                                                              ,86   ,87   ,88   ,89   ,90   ,91
                                                                              ,92   ,93   ,94
      WRITE (NTAPE2) NUMBER,NPART3,
     1((C(K,L),K=1,3),L=1,2),((C(K,L),K=1,3),L=1,3,2)                         ,95   ,96   ,97   ,98   ,99   ,100
                                                                              ,101  ,102  ,103  ,104  ,105  ,106
                                                                              ,107  ,108  ,109
      WRITE (NWRITE,930) ((C(K,L),K=1,3),L=2,4,2),
     1((C(K,L),K=1,3),L=3,5,2)                                                ,110  ,111  ,112  ,113  ,114  ,115
                                                                              ,116  ,117  ,118  ,119  ,120  ,121
                                                                              ,122  ,123
      WRITE (NTAPE2) ((C(K,L),K=1,3),L=2,4,2),
     1((C(K,L),K=1,3),L=3,5,2)                                                ,124  ,125  ,126  ,127  ,128  ,129
                                                                              ,130  ,131  ,132  ,133  ,134  ,135
                                                                              ,136  ,137
      WRITE (NWRITE,930) ((C(K,L),K=1,3),L=4,6,2),
     1((C(K,L),K=1,3),L=5,6)                                                  ,138  ,139  ,140  ,141  ,142  ,143
                                                                              ,144  ,145  ,146  ,147  ,148  ,149
                                                                              ,150  ,151
      WRITE (NTAPE2) ((C(K,L),K=1,3),L=4,6,2),
     1((C(K,L),K=1,3),L=5,6)                                                  ,152  ,153  ,154  ,155  ,156  ,157
                                                                              ,158  ,159  ,160  ,161  ,162  ,163
                                                                              ,164  ,165
      GO TO 130                                                               166
100   WRITE (NWRITE,920) NUMBER,NPART2,
     1((C(K,L),K=1,3),L=2,4,2),((C(K,L),K=1,3),L=3,5,2)                       ,167  ,168  ,169  ,170  ,171  ,172
                                                                              ,173  ,174  ,175  ,176  ,177  ,178
                                                                              179   ,180  ,181
      WRITE (NTAPE2) NUMBER,NPART2,
     1((C(K,L),K=1,3),L=2,4,2),((C(K,L),K=1,3),L=3,5,2)                       ,182  ,183  ,184  ,185  ,186  ,187
                                                                              ,188  ,189  ,190  ,191  ,192  ,193
                                                                              ,194  ,195  ,196
      WRITE (NWRITE,930) ((C(K,L),K=1,3),L=4,6,2),
     1((C(K,L),K=1,3),L=5,6)                                                  ,197  ,198  ,199  ,200  ,201  ,202
                                                                              ,203  ,204  ,205  ,206  ,207  ,208
                                                                              ,209  ,210
      WRITE (NTAPE2) ((C(K,L),K=1,3),L=4,6,2),
     1((C(K,L),K=1,3),L=5,6)                                                  ,211  ,212  ,213  ,214  ,215  ,216
                                                                              ,217  ,218  ,219  ,220  ,221  ,222
                                                                              223   ,224
      GO TO 130                                                               ,225
110   WRITE (NWRITE,920) NUMBER,NPART2,
     1((C(K,L),K=1,3),L=1,2),((C(K,L),K=1,3),L=1,3,2)                         ,226  ,227  ,228  ,229  ,230  ,231
                                                                              ,232  ,233  ,234  ,235  ,236  ,237
                                                                              ,238  ,239  ,240
      WRITE (NTAPE2) NUMBER,NPART2,
     1((C(K,L),K=1,3),L=1,2),((C(K,L),K=1,3),L=1,3,2)                         ,241  ,242  ,243  ,244  ,245  ,246
                                                                              ,247  ,248  ,249  ,250  ,251  ,252
                                                                              ,253  ,254  ,255
```

```
      WRITE (NWRITE,930)
     1((C(K,L),K=1,3),L=2,4,2),((C(K,L),K=1,3),L=3,5,2)                 ,256  ,257  ,258  ,259  ,260  ,261
                                                                        ,262  ,263  ,264  ,265  ,266  ,267
                                                                        ,268  ,269
      WRITE (NTAPE2)
     1((C(K,L),K=1,3),L=2,4,2),((C(K,L),K=1,3),L=3,5,2)                 ,270  ,271  ,272  ,273  ,274  ,275
                                                                        ,276  ,277  ,278  ,279  ,280  ,281
                                                                         282  ,283
      GO TO 130                                                         ,284
  120 WRITE (NWRITE,920) NUMBER,NPART1,
     1((C(K,L),K=1,3),L=2,4,2),((C(K,L),K=1,3),L=3,5,2)                 ,285  ,286  ,287  ,288  ,289  ,290
                                                                         291  ,292  ,293  ,294  ,295  ,296
                                                                        ,297  ,298  ,299
      WRITE (NTAPE2) NUMBER,NPART1,
     1((C(K,L),K=1,3),L=2,4,2),((C(K,L),K=1,3),L=3,5,2)                 ,300  ,301  ,302  ,303  ,304  ,305
                                                                         306  ,307  ,308  ,309  ,310  ,311
                                                                        ,312  ,313  ,314
  130 CONTINUE                                                          ,315  ,316  ,317
      NBODY=NUMBER                                                      ,318
      NUMBER=0

C     WRITE PRINTOUT HEADINGS                                           ,319
      WRITE (NWRITE,940)                                                ,320  ,321
      WRITE (NWRITE,950)

C     WRITE TAPE OUTPUT - XCEN,YCEN,ZCEN ARE
C     PANEL CENTROID COORDINATES      XCON,YCON,ZCON
C     ARE PANEL CONTROL POINT COORDINATES      AREA IS
C     PANEL AREA      THETA AND ALPHA ARE PANEL THETA-
C     INCLINATION AND ALPHA-INCIDENCE SLOPES                            ,322  ,323
  150 DO 160 I=1,NPER1                                                  ,324
      DO 160 J=1,NPLN1                                                  ,325
      NUMBER=NUMBER+1                                                   ,326
      WRITE (NWRITE,960) NUMBER,XCEN(I,J),YCEN(I,J),
     1ZCEN(I,J),XCON(I,J),YCON(I,J),ZCON(I,J),AREA(I,J),
     2THETA(I,J),ALPHA(I,J)                                             ,327  ,328  ,329
  160 WRITE (NTAPE2) NUMBER,XCEN(I,J),YCEN(I,J),
     1ZCEN(I,J),XCON(I,J),YCON(I,J),ZCON(I,J),AREA(I,J),
     2THETA(I,J),ALPHA(I,J),CHORD(I,J)

C     WRITE TAPE OUTPUT - NUMBER OF PANELS PER
C     COLUMN AND FRACTION OF PANEL STREAMWISE
C     CHORD FOR PANEL CONTROL LOCATION                                  ,330  ,331  ,332  ,333  ,334
  200 NPAN=NBODY/NPER1                                                  ,335
      WRITE (NTAPE2) NPAN,XPER                                          ,336  ,337  ,338
  250 RETURN                                                             339
  900 FORMAT (1H1,9X,35HBODY PANEL CORNER POINT COORDINATES/10X,
     186H1 AND 2 INDICATE BODY PANEL LEADING-EDGE POINTS, 3 AND 4 INDICA
     2TE TRAILING-EDGE POINTS)
  910 FORMAT (1H0,5X,5HPANEL,   4(8X,1HX,8X,1HY,8X,1HZ)/
     15X,2HNO,2X,5HPARTS,6X,
     2    3(1H1,8X),3(1H2,8X),3(1H3,8X),3(1H4,8X)//)
  920 FORMAT (1H ,3X,I3,3X,I1,3X,12F9.5)
  930 FORMAT (1H ,13X,12F9.5)
```

```
940  FORMAT (1H1,9X,49HBODY PANEL CENTROID AND CONTROL POINT COORDINATE
    1S)
950  FORMAT (1H0,5HPANEL,6X,2(1HX,8X,1HY,8X,1HZ,8X),5X,4HAREA,9X,
    16HTHETA-,3X,6HALPHA-/13X,3(1HC,8X),3(2HCP,7X),17X,6HINCLIN,3X,
    25HINCID//)
960  FORMAT (1H ,1X,I3,4X,6F9.5,5X,F9.5,5X,2F9.5)
      END                                                                ,340
```

```
      SUBROUTINE OUTPTW
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/IJ,NPTS(16),X(90,16),Y(90,16),Z(90,16),KPNT,VALUE(5),
     1YCEPT(16),SLOPE,KPANEL(2,15),XCOR(16,16),YCOR(16,16),ZCOR(16,16),X
     2INT(2,15),YINT(2,15),ZINT(2,15),XCEN(15,15),YCEN(15,15),ZCEN(15,15
     3),XCON(15,15),YCON(15,15),ZCON(15,15),AREA(15,15),ARAT(15,15),THET
     4A(15,15),ALPHA(15,15),CHORD(15,15),XFOIL(16,25,2),ZFOIL(16,25,2),X
     5NUM(16,2),XTAB1(25),ZTAB1(25),XTAB2(25),ZTAB2(25)
      DIMENSION C(3,5)
      DIMENSION ND3(3333)
      EQUIVALENCE (ND3,XCOR)

C     SUBROUTINE TO OUTPUT WING PANEL DATA

C     DEFINE CONSTANTS
      NWING=0                                                          ,1
      KODEW=0                                                          ,2
      NUMBER=0                                                         ,3
      NPART1=1                                                         ,4
      NPART2=2                                                         ,5
      E=SYM                                                            ,6
      F=0.00001                                                        ,7
      ZHT=ZCOR(1,1)

C     DO-LOOP TO SET SMALL NEGATIVE AND POSITIVE
C     VALUES EQUAL TO ZERO                                             ,8
      DO 20 I=1,3333                                                   ,9
      IF (ABS(ND3(I))-F) 10,10,20                                      ,10
 10   ND3(I)=0.                                                        ,11
 20   CONTINUE

C     WRITE PRINTOUT HEADINGS                                          ,12   ,13
      WRITE (NWRITE,900)                                               ,14   ,15
      WRITE (NWRITE,910)

C     WRITE TAPE OUTPUT - NUMBER IS PANEL NUMBER
C     NPART1,NPART2 INDICATE 1- AND 2-PART PANELS
C     XCOR,YCOR AND ZCOR ARE PANEL CORNER POINT
C     COORDINATES

C     IF KSTART = 2, WRITE OUTPUT DATA FOR
C     INTERSECTION REGION                                              ,16   ,17
      IF (KSTART-1) 150,40,21                                          ,18
 21   DO 39 J=1,NPER1                                                  ,19
      I=1                                                              ,20
      NUMBER=NUMBER+1                                                  ,21
      KGO=KPANEL(1,J)                                                  ,22
      IF (KGO-1) 150,23,24
```

```
C     SETUP OUTPUT ARRAY                                                ,23
 23   C(1,1)=XCOR(I,J+1)                                                ,24
      C(2,1)=YCOR(I,J+1)                                                ,25
      C(3,1)=ZCOR(I,J+1)                                                ,26
      C(1,2)=XCOR(I,J)                                                  ,27
      C(2,2)=YCOR(I,J)                                                  ,28
      C(3,2)=ZCOR(I,J)                                                  ,29
      C(1,3)=XINT(1,J)                                                  ,30
      C(2,3)=YINT(1,J)                                                  ,31
      C(3,3)=ZINT(1,J)                                                  ,32
      GO TO 25                                                          ,33
 24   C(1,1)=XCOR(I,J)                                                  ,34
      C(2,1)=YCOR(I,J)                                                  ,35
      C(3,1)=ZCOR(I,J)                                                  ,36
      C(1,2)=XINT(1,J)                                                  ,37
      C(2,2)=YINT(1,J)                                                  ,38
      C(3,2)=ZINT(1,J)                                                  ,39
      C(1,3)=XCOR(I,J+1)                                                ,40
      C(2,3)=YCOR(I,J+1)                                                ,41
      C(3,3)=ZCOR(I,J+1)                                                ,42
 25   C(1,4)=XCOR(I+1,J)                                                ,43
      C(2,4)=YCOR(I+1,J)                                                ,44
      C(3,4)=ZCOR(I+1,J)                                                ,45
      C(1,5)=XCOR(I+1,J+1)                                              ,46
      C(2,5)=YCOR(I+1,J+1)                                              ,47
      C(3,5)=ZCOR(I+1,J+1)                                              ,48
      KTYPE=1                                                           ,49
      GO TO (27,28),KGO

C     CHECK SUBPANELS AGAINST TOLERANCE                                 ,50
 27   JTEMP=J+1                                                         ,51
      KTEMP=J                                                           ,52
      GO TO 30                                                          ,53
 28   JTEMP=J                                                           ,54
      KTEMP=J+1                                                         ,55
 30   SC=(YCOR(I,JTEMP)-YINT(1,J))/(XCOR(I,J+1)-
     1XCOR(I,J))                                                        ,56
      IF (ABS(SC)-E) 31,31,32                                           ,57
   31 KTYPE=2                                                           ,58
   32 IF (YCOR(I+1,KTEMP)-YCOR(I,KTEMP)) 33,33,34                       ,59
   33 KTYPE=KTYPE+2                                                     ,60
   34 GO TO (36,37,38,140),KTYPE                                        ,61
   36 WRITE (NWRITE,920) NUMBER,NPART2,((C(K,L),K=1,3),
     1L=1,2),((C(K,L),K=1,3),L=1,3,2)                                   ,62  ,63  ,64  ,65  ,66  ,67
                                                                        ,68  ,69  ,70  ,71  ,72  ,73
                                                                        ,74  ,75  ,76
      WRITE (NTAPE2) NUMBER,NPART2,((C(K,L),K=1,3),
     1L=1,2),((C(K,L),K=1,3),L=1,3,2)                                   ,77  ,78  ,79  ,80  ,81  ,82
                                                                        ,83  ,84  ,85  ,86  ,87  ,88
                                                                        ,89  ,90  ,91
      WRITE (NWRITE,930) ((C(K,L),K=1,3),L=2,4,2),
     1((C(K,L),K=1,3),L=3,5,2)                                          ,92  ,93  ,94  ,95  ,96  ,97
                                                                        ,98  ,99  ,100 ,101 ,102 ,103
                                                                        ,104 ,105
```

```
      WRITE (NTAPE2) ((C(K,L),K=1,3),L=2,4,2),
     1((C(K,L),K=1,3),L=3,5,2)                                          ,106  ,107  ,108  ,109  ,110  ,111
                                                                        ,112  ,113  ,114  ,115  ,116  ,117
                                                                        ,118  ,119
      GO TO 39                                                          ,120
   37 WRITE (NWRITE,920) NUMBER,NPART1,((C(K,L),K=1,3),L=2,4,2),
     1((C(K,L),K=1,3),L=3,5,2)                                          ,121  ,122  ,123  ,124  ,125  ,126
                                                                        ,127  ,128  ,129  ,130  ,131  ,132
                                                                        ,133  ,134  ,135
      WRITE (NTAPE2) NUMBER,NPART1,((C(K,L),K=1,3),L=2,4,2),
     1((C(K,L),K=1,3),L=3,5,2)                                          ,136  ,137  ,138  ,139  ,140  ,141
                                                                        ,142  ,143  ,144  ,145  ,146  ,147
                                                                        ,148  ,149  ,150
      GO TO 39                                                          ,151
   38 WRITE (NWRITE,920) NUMBER,NPART1,((C(K,L),K=1,3),
     1L=1,2),((C(K,L),K=1,3),L=1,3,2)                                   ,152  ,153  ,154  ,155  ,156  ,157
                                                                        ,158  ,159  ,160  ,161  ,162  ,163
                                                                        ,164  ,165  ,166
      WRITE (NTAPE2) NUMBER,NPART1,((C(K,L),K=1,3),
     1L=1,2),((C(K,L),K=1,3),L=1,3,2)                                   ,167  ,168  ,169  ,170  ,171  ,172
                                                                        ,173  ,174  ,175  ,176  ,177  ,178
                                                                        ,179  ,180  ,181
   39 CONTINUE
C     WRITE OUTPUT TAPE FOR MAIN WING REGION                            ,182  ,183
   40 DO 50 I=KSTART,KEND                                               ,184
      DO 50 J=1,NPER1                                                   ,185
      NUMBER=NUMBER+1                                                   ,186
      WRITE (NWRITE,920) NUMBER,NPART1,XCOR(I,J),YCOR(I,J),
     1ZCOR(I,J),XCOR(I+1,J),YCOR(I+1,J),ZCOR(I+1,J),XCOR(I,J+1),
     2YCOR(I,J+1),ZCOR(I,J+1),XCOR(I+1,J+1),YCOR(I+1,J+1),ZCOR(I+1,J+1)  ,187  ,188  ,189
   50 WRITE (NTAPE2) NUMBER,NPART1,XCOR(I,J),YCOR(I,J),
     1ZCOR(I,J),XCOR(I+1,J),YCOR(I+1,J),ZCOR(I+1,J),XCOR(I,J+1),
     2YCOR(I,J+1),ZCOR(I,J+1),XCOR(I+1,J+1),YCOR(I+1,J+1),ZCOR(I+1,J+1)

C     IF KEND LESS THAN NPLN1 WRITE OUTPUT DATA
C     FOR WING-TIP REGION     OTHERWISE SKIP THIS
C     SET OF INSTRUCTIONS                                               ,190  ,191  ,192  ,193  ,194
      IF (KEND-NPLN1) 60,85,150                                         ,195
   60 DO 80 J=1,NPER1                                                   ,196
      NUMBER=NUMBER+1                                                   ,197
      KGO=KPANEL(2,J)                                                   ,198
      I=NPLN1                                                           ,199
      GO TO (63,64),KGO

C     SETUP OUTPUT ARRAY                                                ,200
   63 C(1,2)=XINT(2,J)                                                  ,201
      C(2,2)=YINT(2,J)                                                  ,202
      C(3,2)=ZINT(2,J)                                                  ,203
      C(1,4)=XCOR(I+1,J+1)                                              ,204
      C(2,4)=YCOR(I+1,J+1)                                              ,205
      C(3,4)=ZCOR(I+1,J+1)                                              ,206
      C(1,5)=XCOR(I+1,J)                                                ,207
      C(2,5)=YCOR(I+1,J)                                                ,208
```

```
      WRITE (NTAPE2) NUMBER,NPART1,((C(K,L),K=1,3),                    ,317  ,318  ,319
     1L=2,5,3),((C(K,L),K=1,3),L=4,5)                                  ,320  ,321  ,322  ,323  ,324  ,325
                                                                       ,326  ,327  ,328  ,329  ,330  ,331
 80   CONTINUE                                                        ,332  ,333  ,334
 85   NWING=NUMBER                                                    ,335  ,336
      NUMBER=0                                                        ,337

C     WRITE PRINTOUT HEADINGS                                          ,338
      WRITE (NWRITE,940)                                              ,339  ,340
      WRITE (NWRITE,950)

C     WRITE TAPE OUTPUT - XCEN AND YCEN ARE
C     PANEL CENTROID COORDINATES     XCON AND YCON
C     ARE PANEL CONTROL POINT COORDINATES     AREA
C     IS PANEL AREA     ZCEN AND ZCON ARE PANEL
C     THICKNESS AND CAMBER ORDINATES     THETA AND
C     ALPHA ARE PANEL THICKNESS AND CAMBER SLOPES

C     WRITE OUTPUT TAPE                                                ,341  ,342
 90   DO 100 I=1,NPLN1                                                ,343
      DO 100 J=1,NPER1                                                ,344
      NUMBER=NUMBER+1                                                 ,345
      WRITE (NWRITE,960) NUMBER,XCEN(I,J),YCEN(I,J),
     1ZHT,XCON(I,J),YCON(I,J),ZHT,AREA(I,J),ZCEN(I,J),THETA(I,J),
     2ZCON(I,J),ALPHA(I,J)                                            ,346  ,347  ,348
 100  WRITE (NTAPE2) NUMBER,XCEN(I,J),YCEN(I,J),
     1ZHT,XCON(I,J),YCON(I,J),ZHT,AREA(I,J),THETA(I,J),ALPHA(I,J),
     2CHORD(I,J)

C     WRITE TAPE OUTPUT - NUMBER OF PANELS PER
C     COLUMN AND FRACTION OF PANEL STREAMWISE
C     CHORD FOR PANEL CONTROL LOCATION                                 ,349  ,350  ,351  ,352  ,353
 110  WRITE (NTAPE2) NPER1,XPER                                       ,354  ,355  ,356
      GO TO 150                                                       ,357
 140  WRITE (NWRITE,970)                                              ,358  ,359
 150  RETURN                                                          ,360
 900  FORMAT (1H1, 9X,35HWING PANEL CORNER POINT COORDINATES/10X,
     186H1 AND 2 INDICATE WING PANEL LEADING-EDGE POINTS, 3 AND 4 INDICA
     2TE TRAILING-EDGE POINTS)
 910  FORMAT (1H0,5X,5HPANEL,   4(8X,1HX,8X,1HY,8X,1HZ)/
     15X,2HNO,2X,5HPARTS,6X,
     2    3(1H1,8X),3(1H2,8X),3(1H3,8X),3(1H4,8X)//)
 920  FORMAT (1H ,3X,I3,3X,I1,3X,12F9.5)
 930  FORMAT (1H ,13X,12F9.5)
 940  FORMAT (1H1, 9X,49HWING PANEL CENTROID AND CONTROL POINT COORDINAT
     1ES)
 950  FORMAT (1H0,1X,5HPANEL,6X,2(1HX,8X,1HY,8X,1HZ,8X),5X,4HAREA,7X,
     11HZ,9X,6HALPHA-,2X,1HZ,10X,6HALPHA-/14X,3(1HC,8X),3(2HCP,7X),
     216X,2(5HTHICK,4X),2(6HCAMBER,4X)//)
 960  FORMAT (1H ,1X,I3,4X,6F9.5,5X,F9.5,5X,4F9.5)
 970  FORMAT (1H1,33HERROR MESSAGE - SUBROUTINE OUTPTB/10X,54HINBOARD (O
     1R OUTBOARD) PANEL COLUMN HAS ZERO-AREA PANEL/10X,45HNON-CONSTANT N
```

```
      C(3,5)=ZCOR(I+1,J)                                                ,209
      GO TO 65                                                          ,210
 64   C(1,2)=XCOR(I+1,J)                                                ,211
      C(2,2)=YCOR(I+1,J)                                                ,212
      C(3,2)=ZCOR(I+1,J)                                                ,213
      C(1,4)=XINT(2,J)                                                  ,214
      C(2,4)=YINT(2,J)                                                  ,215
      C(3,4)=ZINT(2,J)                                                  ,216
      C(1,5)=XCOR(I+1,J+1)                                              ,217
      C(2,5)=YCOR(I+1,J+1)                                              ,218
      C(3,5)=ZCOR(I+1,J+1)                                              ,219
 65   C(1,1)=XCOR(I,J)                                                  ,220
      C(2,1)=YCOR(I,J)                                                  ,221
      C(3,1)=ZCOR(I,J)                                                  ,222
      C(1,3)=XCOR(I,J+1)                                                ,223
      C(2,3)=YCOR(I,J+1)                                                ,224
      C(3,3)=ZCOR(I,J+1)                                                ,225
      KTYPE=1                                                           ,226
      GO TO (67,68),KGC

C     CHECK SUBPANELS AGAINST TOLERANCE                                 ,227
 67   JTEMP=J                                                           ,228
      KTEMP=J+1                                                         ,229
      GO TO 70                                                          ,230
 68   JTEMP=J+1                                                         ,231
      KTEMP=J                                                           ,232
 70   SC=(YCOR(I+1,JTEMP)-YINT(2,J))/(XCOR(I+1,J+1)-                    ,233
     1XCOR(I+1,J))                                                      ,234
      IF (ABS(SC)-E) 73,73,74                                           ,235
 73   KTYPE=2                                                           ,236
   74 IF (YCOR(I+1,KTEMP)-YCOR(I,KTEMP)) 75,75,76                       ,237
   75 KTYPE=KTYPE+2                                                     ,238
   76 GO TO (77,78,79,140),KTYPE                                        ,239 ,240 ,241 ,242 ,243 ,244
 77   WRITE (NWRITE,920) NUMBER,NPART2,((C(K,L),K=1,3),L=1,4)           ,245 ,246 ,247
      WRITE (NTAPE2) NUMBER,NPART2,((C(K,L),K=1,3),L=1,4)               ,248 ,249 ,250 ,251 ,252 ,253
                                                                        ,254 ,255 ,256
      WRITE (NWRITE,930) ((C(K,L),K=1,3),L=2,5,3),                      ,257 ,258 ,259 ,260 ,261 ,262
     1((C(K,L),K=1,3),L=4,5)                                            ,263 ,264 ,265 ,266 ,267 ,268
                                                                        ,269 ,270
      WRITE (NTAPE2) ((C(K,L),K=1,3),L=2,5,3),                          ,271 ,272 ,273 ,274 ,275 ,276
     1((C(K,L),K=1,3),L=4,5)                                            ,277 ,278 ,279 ,280 ,281 ,282
                                                                        ,283 ,284
      GO TO 80                                                          ,285
 78   WRITE (NWRITE,920) NUMBER,NPART1,((C(K,L),K=1,3),L=1,4)           ,286 ,287 ,288 ,289 ,290 ,291
                                                                        ,292 ,293 ,294
      WRITE (NTAPE2) NUMBER,NPART1,((C(K,L),K=1,3),L=1,4)               ,295 ,296 ,297 ,298 ,299 ,300
                                                                        ,301 ,302 ,303
      GO TO 80                                                          ,304
   79 WRITE (NWRITE,920) NUMBER,NPART1,((C(K,L),K=1,3),                 ,305 ,306 ,307 ,308 ,309 ,310
     1L=2,5,3),((C(K,L),K=1,3),L=4,5)                                   ,311 ,312 ,313 ,314 ,315 ,316
```

```
EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

ZUMBER OF PANELS FOR THAT COLUMN)
 END                                                                    ,361
```

```
      SUBROUTINE OUTW INTAPEO,NWING,NCOLW,NROWW,Z)

C     WING OUPUT FORMAT

      DIMENSION Z(1)
      WRITE (NTAPEO,6005) (I,I=1,NCOLW)                                  ,1    ,2    ,3    ,4    ,5
      WRITE (NTAPEO,6010)                                                ,6    ,7
      DO 100 J=1,NROWW                                                   ,8
  100 WRITE (NTAPEO,6015) (J,(Z(I),I=J,NWING,NROWW))                     ,9    ,10   ,11   ,12   ,13   ,14
                                                                         ,15
 6005 FORMAT(1H0,2X,16HSPANWISE STATION,I8,9I10,(18X,I8,9I10))
 6010 FORMAT(1H ,2X,17HCHORDWISE STATION)
 6015 FORMAT(1H ,I10,F19.5,9F10.5,(F29.5,9F10.5))
      RETURN                                                             ,16
      END                                                                ,17
```

```
      SUBROUTINE PANEL
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,KMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/ ND3(11000)
      DIMENSION TABLE(3)

C     CONTROL PROGRAM FOR PANELING SUBROUTINES
      DATA TABLE /10HBODY PANEL,10HWING PANEL,6HPANEND/

C     DEFINE TAPE UNITS AND CONSTANTS
C     ND1, ND4 ARE INPUT SCRATCH TAPES      NTAPE2 IS
C     OUTPUT SCRATCH TAPE
      ND4=ND2(1)                                                        ,1
      NBODY=0                                                           ,2
      NWING=0                                                           ,3
      KINT=0                                                            ,4
      REWIND ND1                                                        ,5
      REWIND ND4                                                        ,6
      REWIND NTAPE2

C     READ CARD INPUT - X- AND Y-PERCENT VALUES FOR
C     BODY AND WING PANEL CONTROL POINT LOCATION                        ,7
      READ (NREAD,890) XPER,YPER

C     DEFINE KODEC      IF KODEC = 0, Y-PRIME FOR
C     CONTROL POINT TO BE SAME AS THAT FOR CENTROID                     ,8     ,9     ,10
      KODEC=1                                                           ,11
      IF (YPER) 150,5,10                                                ,12
 5    KODEC=0                                                           ,13
 10   N=3

C     READ CARD INPUT FOR CONTROL CARD                                  ,14
 15   KCODE=INTURP(TABLE,N,NREAD,NWRITE)                                ,15
      IF (KCODE) 15,15,20

C     CALL SUBROUTINES                                                  ,16
 20   GO TO (50,100,140),KCODE                                          ,17
 50   CALL BODY                                                         ,18
      IF (KODEB) 150,55,150                                             ,19
 55   IF (KODEI) 150,15,150                                             ,20
 100  CALL WING                                                         ,21
      IF (KODEW) 150,15,150                                             ,22
 140  WRITE (NWRITE,930)                                                ,23    ,24
 150  REWIND NTAPE2                                                     ,25
      RETURN                                                            ,26
 890  FORMAT (6F10.0)
 930  FORMAT (1H1)
       END                                                              ,27
```

```
      SUBROUTINE PARTV

C     PARTITION VELOCITY COMPONENTS MATRIX

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE

      DIMENSION U(100),V(100),W(100),UU(100),VV(100),WW(100)


      DO 100 J=1,NBODY                                                  ,1
      READ (NTAPEB) (U(I),V(I),W(I),I=1,NBODY),(UU(I),VV(I),WW(I),I=1,
     1NWING)                                                            ,2    ,3    ,4    ,5    ,6    ,7
                                                                        ,8    ,9
      WRITE (NTAPED) (U(I),V(I),W(I),I=1,NBODY)                         ,10   ,11   ,12   ,13   ,14
      WRITE (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)                      ,15   ,16   ,17   ,18   ,19
  100 CONTINUE                                                          ,20   ,21
      END FILE NTAPED                                                   ,22
      REWIND NTAPEA
                                                                        ,23
      DO 200 J=1,NBODY                                                  ,24
      READ (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)                       ,25   ,26   ,27   ,28   ,29
      WRITE (NTAPED) (UU(I),VV(I),WW(I),I=1,NWING)                      ,30   ,31   ,32   ,33   ,34
  200 CONTINUE                                                          ,35   ,36
      END FILE NTAPED                                                   ,37
      REWIND NTAPEA
                                                                        ,38
      DO 300 J=1,NWING                                                  ,39
      READ (NTAPEB) (U(I),V(I),W(I),I=1,NBODY),(UU(I),VV(I),WW(I),I=1,NW
     1ING)                                                              ,40   ,41   ,42   ,43   ,44   ,45
                                                                        ,46   ,47
      WRITE (NTAPED) (U(I),V(I),W(I),I=1,NBODY)                         ,48   ,49   ,50   ,51   ,52
      WRITE (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)                      ,53   ,54   ,55   ,56   ,57
  300 CONTINUE                                                          ,58   ,59
      REWIND NTAPEA                                                     ,60
      REWIND NTAPEB                                                     ,61
      END FILE NTAPED
                                                                        ,62
      DO 400 J=1,NWING                                                  ,63
      READ (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)                       ,64   ,65   ,66   ,67   ,68
      WRITE (NTAPED) (UU(I),VV(I),WW(I),I=1,NWING)                      ,69   ,70   ,71   ,72   ,73
  400 CONTINUE                                                          ,74   ,75
      END FILE NTAPED                                                   ,76
      REWIND NTAPEA                                                     ,77
      REWIND NTAPED
                                                                        ,78
      RETURN                                                            ,79
      END                                                               ,80
```

```
      SUBROUTINE POLXN(C,P,M,N,EP,MAX,Q,LQ,L)
      DIMENSION C(1),P(M,N),Q(M,MAX),LQ(2,1)
C
C     GIVEN A POLYGONAL ARC, P, OF N POINTS IN M-SPACE, FIND ITS
C     INTERSECTIONS, Q, WITH PLANE C.
C
C     L = NO. OF INTERSECTIONS.
C     LQ CONTAINS POINT NUMBERS OF POINTS ADJACENT TO EACH Q.
C
      L=0                                                               ,1
      T=0.                                                              ,2
      DO 20 I=1,M                                                       ,3
 20   T=T+C(I)**2
C                                                                       ,4    ,5
      IF ( T .LE.0. .OR. M .LE. 0 .OR. MAX .LE. 0) GO TO 2000
C     SCALE EP FOR LATER USE                                            ,6    ,7    ,8
 100  E=ABS(EP)*SQRT(T)
C                                                                       ,9
      DO 1000 K=1,N
C                                                                       ,10
      FK=C(M+1)                                                         ,11
      DO 150 I=1,M                                                      ,12
 150  FK=FK+C(I)*P(I,K)
C                                                                       ,13   ,14
      IF (ABS(FK)-E)500,500,600
C                                                                       ,15
 500  U=0.                                                              ,16
      V=-1.                                                             ,17
      K1=K                                                              ,18
      GO TO 900
C                                                                       ,19
 600  IF (K-1)1000,1000,700
C                                                                       ,20
 700  IF (FK*FP)800,1000,1000
C                                                                       ,21
 800  T=FK-FP                                                           ,22
      U=FK/T                                                            ,23
      V=FP/T                                                            ,24
      K1=K-1
C                                                                       ,25
 900  L=L+1                                                             ,26
      DO 920 I=1,M                                                      ,27
 920  Q(I,L)=U*P(I,K-1)-V*P(I,K)
C                                                                       ,28   ,29
      LQ(1,L)=K1                                                        ,30
      LQ(2,L)=K
C                                                                       ,31
      IF (L-MAX)1000,9000,9000
C                                                                       ,32
 1000 FP=FK
C                                                                       ,33   ,34
      GO TO 9000
C                                                                       ,35
```

```
EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

 2000 L=-1
C                                                                    ,36
 9000 RETURN                                                         ,37
      END                                                            ,38
```

```
      SUBROUTINE QRAT(X,Y,IXY,KODE,XI,YI,C,NU)
      DIMENSION X(1),Y(1),C(3)
C
C     FIND COEFFICIENTS OF QUADRATIC THRU 3 POINTS.
C     1ST PT IS (X,Y), 2ND PT IS (X(IXY+1),Y(IXY+1)),
C     3RD PT IS(X(2*IXY+1),Y(2*IXY+1))
C     IF KODE .GT. 0, EVALUATE AT X=XI
C
C     USING 32769 INSTEAD OF 1, ALLOWS IXY TO BE NEGATIVE.
      JXY=IXY+32769                                                     ,1
      KXY=IXY+JXY                                                       ,2
      DX21=X(JXY)-X                                                     ,3
      DY21=Y(JXY)-Y                                                     ,4
      DX32=X(KXY)-X(JXY)                                                ,5
      DY32=Y(KXY)-Y(JXY)                                                ,6
      DX31=X(KXY)-X                                                     ,7
      DY31=Y(KXY)-Y                                                     ,8
      SX31=X(KXY)+X
C                                                                       ,9
 100  IF (DX21*DX32*DX31)300,200,300                                    ,10
 200  NU=1                                                              ,11
      GO TO 1000
C                                                                       ,12
 300  C(3)=(DY32/DX32-DY21/DX21)/DX31                                   ,13
      C(2)=DY31/DX31-C(3)*SX31                                          ,14
      C(1)=Y(JXY)-(C(2)+C(3)*X(JXY))*X(JXY)                             ,15
      NU=0
C                                                                       ,16
 400  IF (KODE)1000,1000,500                                            ,17
 500  YI=C(1)+(C(2)+C(3)*XI)*XI
C                                                                       ,18
 1000 RETURN                                                            ,19
      END                                                               ,20
```

```
      SUBROUTINE READ (NTAPEI,NTAPEO,NTAPEC,NM,S)
      DIMENSION S(1),DICT(2)

C     CONTROLS DATA READ IN OPTION FOR SPECIFYING WING CAMBER AND
C     PRESSURE DISTRIBUTION, ALSO BODY CAMBER

      DATA DICT/6HCONSTA,5HGIVEN/

      IGO=INTURP(DICT,2,NTAPEI,NTAPEO)                                  ,1
      IGO=IGO+1                                                         ,2
      GO TO (50,60,70),IGO                                              ,3
   50 CONTINUE                                                          ,4
      READ (NTAPEI,5000) (S(I),I=1,NM)                                  ,5    ,6    ,7    ,8    ,9
      GO TO 999                                                         ,10
   60 CONTINUE                                                          ,11
      READ (NTAPEI,5000) SCONST                                         ,12   ,13   ,14
      DO 65 I=1,NM                                                      ,15
   65 S(I)=SCONST                                                       ,16   ,17
      GO TO 999                                                         ,18
   70 CONTINUE                                                          ,19
      READ (NTAPEC) (S(I),I=1,NM)
                                                                        ,20   ,21   ,22   ,23   ,24
 5000 FORMAT(7F10.0)

  999 CONTINUE                                                          ,25
      RETURN                                                            ,26
      END                                                               ,27
```

```
      SUBROUTINE REDUCE

C     REDUCE AERODYMANIC INFLUENCE COEFFICIENTS MATRIX AND FORM THE
C     (D) AND (E) MATRICES

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF
     1,NTAPEI,NTAPEC,NBODY,NWING,XMACH,SYM,KACE

      DIMENSION ABB(100,100),AWB(100,100),ABW(100),AWW(100),E(100),D(100
     1)

      EQUIVALENCE (E,D)

      DO 100 J=1,NBODY                                                  ,1
      READ (NTAPEC) (AWB(I,J),I=1,NWING)                                ,2    ,3    ,4    ,5    ,6
  100 CONTINUE
                                                                        ,7    ,8
      DO 200 J=1,NBODY                                                  ,9
      READ (NTAPED) (ABB(I,J),I=1,NBODY)                                ,10   ,11   ,12   ,13   ,14
  200 CONTINUE                                                          ,15   ,16
      REWIND NTAPED

C     FORM (D) MATRIX                                                   ,17
      DO 300 K=1,NBODY                                                  ,18
      DO 250 I=1,NWING                                                  ,19
      D(I)=0.0                                                          ,20
      DO 250 J=1,NBODY                                                  ,21
      D(I)=D(I)+AWB(I,J)*ABB(J,K)                                       ,22
  250 CONTINUE                                                          ,23   ,24   ,25
      WRITE (NTAPEF) (D(I),I=1,NWING)                                   ,26   ,27   ,28   ,29   ,30
  300 CONTINUE                                                          ,31   ,32
      END FILE NTAPEF

                                                                        ,33
      DO 350 J=1,NBODY                                                  ,34
      WRITE (NTAPEF) (ABB(I,J),I=1,NBODY)                               ,35   ,36   ,37   ,38   ,39
  350 CONTINUE

C     MATRIX REDUCTION                                                  ,40   ,41
      DO 600 J=1,NWING                                                  ,42
      READ (NTAPEA)(ABW(I),I=1,NBODY),(AWW(I),I=1,NWING)
C     FORM (E) MATRIX                                                   ,43   ,44   ,45   ,46   ,47   ,48
                                                                        ,49   ,50
      DO 400 M=1,NBODY                                                  ,51
      E(M)=0.0                                                          ,52
      DO 400 N=1,NBODY                                                  ,53
      E(M)=E(M)-ABB(M,N)*ABW(N)                                         ,54
  400 CONTINUE                                                          ,55   ,56   ,57
      DO 500 I=1,NWING                                                  ,58
      DO 500 M=1,NBODY                                                  ,59
      AWW(I)=AWW(I)+AWB(I,M)*E(M)                                       ,60
  500 CONTINUE                                                          ,61   ,62   ,63
```

```
EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      WRITE (NTAPEF) (E(I),I=1,NBODY)                                  ,64   ,65   ,66   ,67   ,68
      WRITE (NTAPEE) (AWW(I),I=1,NWING)                                ,69   ,70   ,71   ,72   ,73
  600 CONTINUE
                                                                       ,74   ,75
      REWIND NTAPEA                                                    ,76
      REWIND NTAPEE                                                    ,77
      END FILE NTAPEF                                                  ,78
      REWIND NTAPEF
                                                                       ,79
      RETURN                                                           ,80
      END                                                              ,81
```

```
      SUBROUTINE RICH3A(NP,P,Q,KD,MAX,CHD,NA,A,NU)
      DIMENSION P(3),Q(3),A(3),NU(3),T(12),U(3),V(3)
C
C     P = ARRAY OF NP POINTS, STORED (X1,Y1,Z1,X2,Y2,Z2,....)
C     Q = SCRATCH ARRAY OF LENGTH 3*NP.  IT MAY BE THE SAME ARRAY
C         AS P IF OK TO DESTROY P.
C
C     RICH3A PERFORMS A 3D INTERPOLATION TO PRODUCE ARRAY A
C     WHICH CONTAINS P WITH INTERSPERSED POINTS.
C
C     NA = NO OF POINTS IN A.
C     MAX = DIMENSION OF A.
C     KD = 0 IF ENRICHING IS DONE IN PLANE THRU 1ST PT, LAST PT,
C       AND PT FARTHEST FROM LINE THRU END PTS.
C     KD = 1 OR 2 IF IN PLANE THRU END PTS WHICH IS NORMAL TO
C       XY PLANE
C     KD = 1 IF PROJECTION OF A ON XY PLANE IS TO BE STRAIGHT LINES
C       BETWEEN POINTS OF P (WHEN PROJECTED TO XY PLANE).
C     KD = 2 OTHERWISE.
C     CHD = CHORD HEIGHT TOLERANCE (SEE ENRYCH)
C
C     INPUTS ARE NP,P,KD,MAX,CHD,NU(2),NU(3)
C     OUTPUTS ARE Q,NA,A,NU(1),NU(3)
C
      MU=0                                                              ,1
      NA=0                                                              ,2
      NP2=NP+NP                                                         ,3
      NP3=NP2+NP                                                        ,4
      MAXR=MAX-NP2
C                                                                       ,5
      IF(NP .LE. 3  .OR.  CHD .LE. 0.) GO TO 7000                       ,6     ,7     ,8
 20   IF(KD)30,30,90
C     FIND ROTATION MATRIX TO SYSTEM WHICH HAS ORIGIN AT 1ST PT,
C       +X AXIS PASSING THRU LAST PT, Z AXIS IN PLANE DESCRIBED FOR KD=0
C
C     FIND THE POINT STARTING IN P(3*JP-2) WHICH IS FARTHEST FROM
C       THE LINE THRU END PTS, AND ITS DISTANCE = DIST.                 ,9
 30   CALL DMAXL(P,NP,DIST,JP,U,NU)
C     U IS UNIT VECTOR FROM 1ST PT TO LAST PT.                          ,10
      IF(NU)9000,40,9000                                                ,11
 40   IF(DIST-CHD)7000,7000,50                                          ,12
 50   D=UVECN(P,P(3*JP-2),V,0,3)                                        ,13
      I=1                                                               ,14
      NU=21                                                             ,15
      CALL VCROS(V,U,T(4),D,I)                                          ,16
      IF(I)9000,60,9000
C     T(4)=V CROSS U IS UNIT VECTOR ALONG NEW Y AXIS                    ,17
 60   I=1                                                               ,18
      CALL VCROS(U,T(4),T(7),D,I)                                       ,19
      CALL TRAV(T,U,3)
C     ROTATION MATRIX NOW FORMED.                                       ,20
      T(12)=P(3)                                                        ,21
      GO TO 130
```

```
C
C       KD=1 OR 2
C       T = TRANSFORMATION MATRIX (SEE TROTPT).                          ,22
 90     DO 100 I=1,12                                                    ,23
 100    T(I)=0.
C                                                                        ,24    ,25
        T(1)=P(NP3-2)-P(1)                                               ,26
        T(2)=P(NP3-1)-P(2)                                               ,27
        D=SQRT(T**2+T(2)**2)                                             ,28
        IF (D)110,110,120                                                ,29
 110    NU=7                                                             ,30
        GO TO 9000
C                                                                        ,31
 120    T(1)=T(1)/D                                                      ,32
        T(2)=T(2)/D                                                      ,33
        T(9)=1.                                                           34
        T(4)=-T(2)                                                       ,35
        T(5)=T(1)
C                                                                        ,36
 130    T(10)=P(1)                                                       ,37
        T(11)=P(2)
C       TRANSLATION MATRIX, IN T(10),T(11),T(12),  NOW FORMED
C
C       TRANSFORM POINTS IN P TO Q                                       ,38
        JX=-2                                                            ,39
        DO 200 I=1,NP                                                     40
        JX=JX+3                                                          ,41
 200    CALL TROTPT(T,T(10),2,P(JX),Q(JX))
C                                                                        ,42    ,43
        IF (MAXR)400,400,500                                             ,44
 400    NU=1                                                              45
        GO TO 9000
C                                                                        ,46
 500    IA=MAXR                                                           47
        IB=IA+NP                                                         ,48
        IXQ=IA+1                                                          49
        IYQ=IB+1                                                         ,50
        DO 600 I1=3,NP3,3                                                ,51
        IA=IA+1                                                           52
        IB=IB+1                                                          ,53
        A(IA)=Q(I1-2)                                                     54
 600    A(IB)=Q(I1)
C                                                                        ,55    ,56
        NO=MAXR                                                          ,57
 700    CALL ENRYCH(A(IXQ),A(IYQ),NP,CHD,A,NO,NU)                        ,58
        IF (NU)800,1000,800                                              ,59
 800    IF (NU-3)9000,900,9000                                           ,60
 900    NU=-1                                                            ,61
 1000   NA=NO/2                                                          ,62
        IF (MAX-NO-NA)1100,1200,1200                                     ,63
 1100   NU=6                                                             ,64
        GO TO 9000
C                                                                        ,65
 1200   IA=NO                                                             66
```

```
      DO 1300 I1=1,NO                                                   ,67
      IB=IA+IA/2                                                        ,68
      A(IB)=A(IA)                                                        69
 1300 IA=IA-1
C                                                                       ,70    ,71
      IA=-2                                                              72
 1500 DO 1700 I=1,NA                                                    ,73
      IA=IA+3                                                           ,74
      CALL BITURP(Q,Q(2),3,NP,KD,A(IA),A(IA+1),NU)                      ,75
      IF (NU)1600,1700,1600                                             ,76
 1600 NU=NU+10                                                           77
      GO TO 9000                                                        ,78
 1700 CALL TROTPT(T,T(10),1,A(IA),A(IA))                                ,79    ,80
      GO TO 8000
C
C     NO INTERPOLATION BECAUSE NP .LT. 4 OR CHD .LE. 0 OR PTS. COLLINEAR ,81
 7000 NA=NP                                                             ,82
      CALL TRAV(A,P,NP3)
C                                                                       ,83
 8000 NU=MU
C                                                                       ,84
 9000 RETURN                                                            ,85
      END                                                               ,86
```

```
      SUBROUTINE RICHNA (A,MAX,M,N,P,EPS,CHD,KOD,NA,NU)
      DIMENSION A(3,1),P(1),NU(3),EP(3)
C
C     GIVEN M ARRAYS OF 3D POINTS, WHERE EACH ARRAY HAS N POINTS, STORED
C     IN A SINGLE ARRAY A OF LENGTH MAX.
C     USE RICH3A ON EACH ARRAY SO THAT EACH ARRAY WILL CONTAIN MORE
C     POINTS.  UPON RETURN, THE DENSE ARRAYS WILL BE STORED INA,
C     HEADED BY A TABLE OF CONTENTS.  THE ORIGINAL ARRAYS
C     ARE DESTROYED.
C
      DO 10 I=1,3                                                       ,1
 10   EP(I)=EPS                                                         ,2    ,3
      NA=0                                                              ,4
      MN=M*N                                                            ,5
      IA=MAX-MN                                                         ,6
      IAI=IA+1                                                          ,7
      MAXR=IA-M                                                         ,8
      IDA=M+1                                                           ,9
      IF(KOD)100,100,20                                                 ,10
 20   EP(3)=1.E+30
C
C     MOVE ARRAY TO BOTTOM OF A.                                        ,11
 100  DO 200 J=1,MN                                                     ,12
 150  IA=IA+1                                                           ,13
      DO 200 I=1,3                                                      ,14
 200  A(I,IA)=A(I,J)
C                                                                       ,15   ,16   ,17
 300  DO 1000 IB=1,M                                                    ,18
      IF(KOD)305,360,305
C
C     CHECK FOR POINTS WHICH HAVE X,Y NEARLY SAME AS PREV. POINT AND
C     Z DIFFERENT.  WRITE COMMENT IF ANY FOUND.  THESE POINTS WILL BE
C     REMOVED LATER BY NWEED, BUT USER SHOULD KNOW ABOUT IT.
C                                                                       ,19
 305  LTAPE=NU(2)                                                       ,20
      IF (LTAPE)360,360,310                                             ,21
 310  LF=IAI+1                                                          ,22
      LL=IAI+N-1                                                        ,23
      DO 350 L=LF,LL                                                    ,24
      DO 320 I=1,2                                                      ,25
      IF (ABS(A(I,L)-A(I,L-1))-EPS)320,350,350                          ,26
 320  CONTINUE                                                          ,27   ,28
      IF (ABS(A(3,L)-A(3,L-1))-EPS)350,350,330                          ,29
 330  LM=L-1                                                            ,30
      WRITE (LTAPE,340)IB,((A(I,J),I=1,3),J=LM,L)                       ,31   ,32   ,33   ,34   ,35   ,36
                                                                        ,37   ,38   ,39
 340  FORMAT(37H0CHECK THESE 2 POINTS ON PERCENT LINE I4/
     1 8H X Y Z =2(3F10.4,5X))
 350  CONTINUE
C
C     REMOVE POINTS TOO CLOSE TO PRECEEDING POINT                       ,40   ,41
 360  NS=NWEED(A(1,IAI),N,3,EP)                                         ,42
      IF (NS-1)400,400,500
```

```
C       ALL POINTS MUST BE COINCIDENT (WITHIN EP)                          ,43
 400    NU=-1                                                              ,44
        GO TO 9000
C                                                                          ,45
 500    CALL RICH3A(NS,A(1,IAI),A(1,IAI),KOD,3*MAXR,CHD,ND,A(1,IDA),NU)    ,46
        IF (NU)9000,600,9000
C       MAKE ENTRIES IN TABLE OF CONTENTS                                  ,47
 600    A(1,IB)=P(IB)                                                      ,48
        A(2,IB)=ND                                                         ,49
        A(3,IB)=(IDA-1)*3+1
C                                                                          ,50
        IAI=IAI+N                                                          ,51
        IDA=IDA+ND                                                         ,52
        MAXR=MAXR+N-ND                                                     ,53
        NA=NA+ND
C                                                                          ,54
 1000   CONTINUE
C                                                                          ,55    ,56
 9000   RETURN                                                             ,57
        END                                                                ,58
```

```
      SUBROUTINE RITE (NFMT,NTAPEO,NP,NROW,NCOL,THETA,U,V,W)

C     PRINT OUT VELOCITY COMPONENTS

      DIMENSION NFMT(1),U(1),V(1),W(1)

      IF (NFMT(1)) 5,999,5                                          ,1
    5 CONTINUE                                                      ,2
      IGO=NFMT(1)                                                   ,3
      GO TO (10,20,30,40,50,60,70),IGO                              ,4
   10 WRITE (NTAPEO,6030)                                           ,5    ,6
      GO TO 90                                                      ,7
   20 WRITE (NTAPEO,6040)                                           ,8    ,9
      GO TO 90                                                      ,10
   30 WRITE (NTAPEO,6045)                                           ,11   ,12
      GO TO 90                                                      ,13
   40 WRITE (NTAPEO,6050)                                           ,14   ,15
      GO TO 100                                                     ,16
   50 WRITE (NTAPEO,6055)                                           ,17   ,18
      GO TO 100                                                     ,19
   60 WRITE (NTAPEO,6060)                                           ,20   ,21
      GO TO 100                                                     ,22
   70 WRITE (NTAPEO,6065)                                           ,23   ,24
      GO TO 100


C     PRINT OUT VELOCITY COMPONENTS AT BODY PANELS                  ,25
   90 CONTINUE                                                      ,26
      WRITE (NTAPEO,6010)                                           ,27   ,28
      CALL OUTB (NTAPEO,NP,NCOL,NROW,THETA,U)                       ,29
      WRITE (NTAPEO,6015)                                           ,30   ,31
      CALL OUTB (NTAPEO,NP,NCOL,NROW,THETA,V)                       ,32
      WRITE (NTAPEO,6020)                                           ,33   ,34
      CALL OUTB (NTAPEO,NP,NCOL,NROW,THETA,W)                       ,35
      GO TO 999
C     PRINT OUT VELOCITY COMPONENTS AT WING PANELS
                                                                    ,36
  100 CONTINUE                                                      ,37
      WRITE (NTAPEO,6010)                                           ,38   ,39
      CALL OUTW (NTAPEO,NP,NCOL,NROW,U)                             ,40
      WRITE (NTAPEO,6015)                                           ,41   ,42
      CALL OUTW (NTAPEO,NP,NCOL,NROW,V)                             ,43
      WRITE (NTAPEO,6020)                                           ,44   ,45
      CALL OUTW (NTAPEO,NP,NCOL,NROW,W)

                                                                    ,46
 6000 FORMAT(1H ,10F11.5)
 6010 FORMAT(1H0,5X,8HAXIAL(U))
 6015 FORMAT(1H0,5X,13HTRANSVERSE(V))
 6020 FORMAT(1H0,5X,11HVERTICAL(W))
 6030 FORMAT(1H0//76H VELOCITY COMPONENTS ON BODY PANELS DUE TO BODY PAN
     1EL PRESSURE SINGULARITIES)
 6040 FORMAT(1H0//76H VELOCITY COMPONENTS ON BODY PANELS DUE TO WING PAN
```

```
    1EL PRESSURE SINGULARITIES)
6045 FORMAT(1H0//55H VELOCITY COMPONENTS ON BODY PANELS DUE TO WING SOU
    1RCES)
6050 FORMAT(1H0//76H VELOCITY COMPONENTS ON WING PANELS DUE TO WING PAN
    1EL PRESSURE SINGULARITIES)
6055 FORMAT(1H0//55H VELOCITY COMPONENTS ON WING PANELS DUE TO WING SOU
    1RCES)
6060 FORMAT(1H0//76H VELOCITY COMPONENTS ON WING PANELS DUE TO BODY PAN
    1EL PRESSURE SINGULARITIES)
6065 FORMAT(1H0//73H VELOCITY COMPONENTS ON WING PANELS DUE TO BODY LIN
    1E SOURCES AND DOUBLETS)

 999 CONTINUE                                                          ,47
     RETURN                                                            ,48
     END                                                               ,49
```

```
C     ROUTINE TO ADJUST STORAGE FOR INV4S AND CALL IT
C
      SUBROUTINE SINVRT(A,K,N,IR1,IR2,SCALE,DET,NDETXP)
C
C     A       = MATRIX TO BE INVERTED
C     K       = ROW DIMENSION STATEMENT ENTRY FOR A
C     N       = NUMBER OF ROWS,COLUMNS IN A
C     IR1     = 0 IF INVERSION SUCCESSFUL
C             = 1 IF OVERFLOW OCCURS
C             = 2 IF MATRIX IS SINGULAR
C             = 3 IF SCALED INVERSE CANNOT BE RE-SCALED
C             = 4 IF ROWS AND COLUMNS CANNOT BE REARRANGED
C                 (MACHINE ERROR)
C             = 10+I IF CODE 3 OCCURRED AFTER CODE I
C             = 20+I IF CODE 4 OCCURRED AFTER CODE I
C     IR2     = CURRENT REDUCTION STAGE, IF IR1=1
C             = RANK, IF IR1=2
C             = 0, OTHERWISE
C     SCALE   = 0, IF IR1 NOT = 3
C             = SCALE FACTOR IF IR1=3
C     DET     = 0, IF IR1=2
C             = DECIMAL PART OF DETERMINANT IF IR1 NOT = 2
C               (A NUMBER GT OR E TO 1.0 AND LT 10.0
C     NDETXP  = 0, IF IR1=2
C             = EXPONENT PART OF DETERMINANT IF IR1 NOT = 2
C                 (DETERMINANT= DET*10.0**NDTXP)
C
C     NEEDS N**2+2N+3 CORE LOCATIONS
C
      DIMENSION A(K,10)
C
C     TRANSPOSE TO ROW-WISE STORAGE
C
      NM1=N-1                                                          ,1
      DO 10 J=1,NM1                                                    ,2
      L=J+1                                                            ,3
      DO 10 I=L,N                                                      ,4
      TEMP=A(I,J)                                                      ,5
      A(I,J)=A(J,I)                                                    ,6
 10   A(J,I)=TEMP
C
C     SQEEZE OUT ALL UNUSED STORAGE                                    ,7    ,8    ,9
      IF (N-K) 25,100,100                                              ,10
 25   L=N+1                                                            ,11
      DO 50 J=2,N                                                      ,12
      DO 50 I=1,N                                                      ,13
      A(L,1)=A(I,J)                                                    ,14
 50   L=L+1
C
C     CALL INVERSION ROUTINE
C                                                                      ,15   ,16   ,17
 100  CALL INV4S(A,N,IR1,IR2,SCALE,DET,NDETXP)
```

```
C     IF ANY OTHER ERROR BUT 3, DO NOT REARRANGE
C
      IF (IR1) 150,150,125                                              ,18
  125 IF (IR1-3) 7500,150,7500                                          ,19
C
C     EXPAND STORAGE TO NORMAL
C
  150 IF (N-K) 175,275,275                                              ,20
  175 L=N**2                                                            ,21
      J=N                                                               ,22
  180 I=N                                                               ,23
  190 A(I,J)=A(L,1)                                                     ,24
      A(L,1)=0.                                                         ,25
      L=L-1                                                             ,26
      IF (I-1) 225,225,200                                              ,27
  200 I=I-1                                                             ,28
      GO TO 190                                                         ,29
  225 IF (J-2)275,275,250                                               ,30
  250 J=J-1                                                             ,31
      GO TO 180                                                         ,32
C
C     TRANSPOSE BACK TO COLUMN-WISE STORAGE
C
  275 DO 300 J=1,NM1                                                    ,33
      L=J+1                                                             ,34
      DO 300 I=L,N                                                      ,35
      TEMP=A(I,J)                                                       ,36
      A(I,J)=A(J,I)                                                     ,37
  300 A(J,I)=TEMP                                                       ,38
 7500 RETURN                                                            ,39   ,40   ,41
      END                                                               ,42
                                                                        ,43
```

```
      SUBROUTINE SLOPEW
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/IJ,NPTS(16),X(90,16),Y(90,16),Z(90,16),KPNT,VALUE(5),
     1YCEPT(16),SLOPE,KPANEL(2,15),XCOR(16,16),YCOR(16,16),ZCOR(16,16),X
     2INT(2,15),YINT(2,15),ZINT(2,15),XCEN(15,15),YCEN(15,15),ZCEN(15,15
     3),XCON(15,15),YCON(15,15),ZCON(15,15),AREA(15,15),ARAT(15,15),THET
     4A(15,15),ALPHA(15,15),CHORD(15,15),XFOIL(16,25,2),ZFOIL(16,25,2),X
     5NUM(16,2),XTAB1(25),ZTAB1(25),XTAB2(25),ZTAB2(25)
      DIMENSION C(4),P(3,50),R(3,50),Q(3,5),LOC(3)
      DIMENSION CLENG(15),NC(3)

C     SUBROUTINE TO CALCULATE WING PANEL THICKNESS
C     AND CAMBER SLOPES
      IF (KOPTF) 500,470,10

C     DEFINE CONSTANTS
C     M,MAX,E AND C ARE CONSTANTS AND ARRAYS
C     USED BY SUBROUTINE PCLXN     IXY,KXY AND NC ARE
C     USED BY SUBROUTINE BITURP                                          ,1
 10   M=3                                                                ,2
      MAX=1                                                              ,3
      E=0.001                                                            ,4
      NPER2=NPER-2                                                       ,5
      NPERT=NPER+1                                                       ,6
      CONST=10000.                                                       ,7
      IXY=1                                                              ,8
      KXY=2                                                              ,9
      ZHT=ZCOR(1,1)                                                      ,10
      C(1)=0.                                                            ,11
      C(2)=-1.                                                           ,12
      C(3)=0.                                                            ,13
      NC(1)=0                                                            ,14
      NC(2)=6                                                            ,15
      NC(3)=500

C     DEFINE ARRAYS FOR WING LEADING AND TRAILING
C     EDGES     P IS ARRAY FOR LEADING EDGE AND R
C     IS ARRAY FOR TRAILING EDGE                                         ,16
 20   N1=NPTS(1)                                                         ,17
      DO 30 I=1,N1                                                       ,18   ,19
      P(1,I)=X(I,1)                                                      ,20
      P(2,I)=Y(I,1)                                                      ,21
 30   P(3,I)=Z(I,1)                                                      ,22   ,23
      N2=NPTS(NPER)                                                      ,24
      DO 40 I=1,N2                                                       ,25   ,26
      R(1,I)=X(I,NPER)                                                   ,27
      R(2,I)=Y(I,NPER)                                                   ,28
 40   R(3,I)=Z(I,NPER)                                                   ,29   ,30
      IF (KSTART-1) 500,150,50
```

```
C     CALCULATE PANEL SLOPES ON INBOARD COLUMN IF
C     WING AND BODY INTERSECT IN NON-STREAMWISE LINE                    ,31
 50   I=1                                                               ,32
 60   NUM1=XNUM(1,1)

C     DEFINE ARRAYS FOR AIRFOIL COORDINATES     XTAB1,ZTAB1
C     ARE ARRAYS FOR UPPER SURFACE COORDINATES     XTAB2,
C     ZTAB2 ARE ARRAYS FOR LOWER SURFACE COORDINATES                    ,33
      DO 65 J=1,NUM1                                                    ,34   ,35
      XTAB1(J)=XFOIL(I,J,1)                                             ,36
 65   ZTAB1(J)=ZFOIL(I,J,1)                                             ,37   ,38
      CALL CHECK (XTAB1,NUM1,2,L,NWRITE)                                ,39
      IF (L) 450,70,450                                                 ,40
 70   NUM2=XNUM(I,2)                                                    ,41
      DO 75 J=1,NUM2                                                    ,42   ,43
      XTAB2(J)=XFOIL(I,J,2)                                             ,44
 75   ZTAB2(J)=ZFOIL(I,J,2)                                             ,45   ,46
      CALL CHECK (XTAB2,NUM2,2,L,NWRITE)                                ,47
      IF (L) 450,77,450                                                 ,48
   77 DO 100 J=1,NPER1

C     CALCULATE CHORD LENGTH
C     XLEAD AND XTRAIL ARE WING LEADING- AND TRAILING-
C     EDGE INTERCEPTS OF CHORD THROUGH CONTROL POINT
C     CLENG IS CHORD LENGTH                                             ,49   ,50
      C(4)=YCON(I,J)                                                    ,51
      CALL POLXN (C,P,M,N1,E,MAX,Q,LOC,NINT)                            ,52
      IF (NINT-1) 480,80,490                                            ,53
 80   XLEAD=Q(1,1)                                                      ,54
      CALL POLXN (C,R,M,N2,E,MAX,Q,LOC,NINT)                            ,55
      IF (NINT-1) 480,90,490                                            ,56
 90   XTRAIL=Q(1,1)                                                     ,57
      CLENG(J)=XTRAIL-XLEAD

C     CALCULATE CAMBER AND THICKNESS ORDINATES
C     ZCON AND ZCEN ARE Z-CAMBER AND Z-THICKNESS
C     ORDINATES                                                         ,58
      FRAC=(XCON(I,J)-XLEAD)/CLENG(J)                                   ,59
      FRAT=(XCEN(I,J)-XLEAD)/CLENG(J)                                   ,60
 91   CALL BITURP (XTAB1,ZTAB1,IXY,NUM1,KXY,FRAC,ZR,NC)                 ,61
      IF (NC(1)) 460,92,92                                              ,62
 92   ZCU=ZR                                                            ,63
      CALL BITURP (XTAB2,ZTAB2,IXY,NUM2,KXY,FRAC,ZR,NC)                 ,64
      IF (NC(1)) 460,93,93                                              ,65
 93   ZCL=ZR                                                            ,66
      ZCON(I,J)=((ZCU+ZCL)/2.)*CLENG(J)+ZHT                             ,67
      CALL BITURP (XTAB1,ZTAB1,IXY,NUM1,KXY,FRAT,ZR,NC)                 ,68
      IF (NC(1)) 460,97,97                                              ,69
 97   ZTU=ZR                                                            ,70
      CALL BITURP (XTAB2,ZTAB2,IXY,NUM2,KXY,FRAT,ZR,NC)                 ,71
      IF (NC(1)) 460,98,98                                              ,72
 98   ZTL=ZR                                                            ,73
      ZCEN(I,J)=((ZTU-ZTL)/2.)*CLENG(J)+ZHT                             ,74
```

```
 100  CONTINUE

C     CALCULATE CAMBER AND THICKNESS SLOPES
C     ALPHA AND THETA ARE ALPHA-CAMBER AND -THICKNESS
C     SLOPES
C     NOTE THAT IF SINGLE AIRFOIL IS INPUT ON
C     CARDS CONSTANT ARRAY FOR ALPHA-CAMBER AND
C     -THICKNESS IS CALCULATED                                          ,75   ,76
      IF (NUMS-1) 500,150,103                                           ,77
  103 XTAB1(1)=XLEAD                                                    ,78
      ZTAB1(1)=ZFOIL(I,1,1)*CLENG(NPER1)+ZHT                            ,79
      XTAB2(1)=XLEAD                                                    ,80
      ZTAB2(1)=ZHT                                                      ,81
      DO 110 J=2,NPER                                                   ,82   ,83
      XTAB1(J)=XCON(I,J-1)                                              ,84
      ZTAB1(J)=ZCON(I,J-1)                                              ,85
      XTAB2(J)=XCEN(I,J-1)                                              ,86
 110  ZTAB2(J)=ZCEN(I,J-1)                                              ,87   ,88
      XTAB1(NPERT)=XTRAIL                                               ,89
      ZTAB1(NPERT)=ZFOIL(I,NUM1,1)*CLENG(NPER1)+ZHT                     ,90
      XTAB2(NPERT)=XTRAIL                                               ,91
      ZTAB2(NPERT)=ZHT                                                  ,92
  120 DO 130 J=2,NPER                                                   ,93   ,94
      EPS=CLENG(J-1)/CONST                                              ,95
      XR=XTAB1(J)-EPS                                                   ,96
      XS=XTAB1(J)+EPS                                                   ,97
  123 CALL BITURP (XTAB1,ZTAB1,IXY,NPERT,KXY,XR,ZR,NC)                  ,98
      IF (NC(1)) 460,124,124                                            ,99
  124 CALL BITURP (XTAB1,ZTAB1,IXY,NPERT,KXY,XS,ZS,NC)                  ,100
      IF (NC(1)) 460,125,125                                            ,101
 125  ALPHA(I,J-1)=ATAN((ZS-ZR)/(XS-XR))                               ,102
      XR=XTAB2(J)-EPS                                                   ,103
      XS=XTAB2(J)+EPS                                                   ,104
  127 CALL BITURP (XTAB2,ZTAB2,IXY,NPERT,KXY,XR,ZR,NC)                  ,105
      IF (NC(1)) 460,128,128                                            ,106
  128 CALL BITURP (XTAB2,ZTAB2,IXY,NPERT,KXY,XS,ZS,NC)                  ,107
      IF (NC(1)) 460,129,129                                            ,108
 129  THETA(I,J-1)=ATAN((ZS-ZR)/(XS-XR))                               ,109
  130 CONTINUE

C     CALCULATE PANEL SLOPES FOR MAIN WING REGION                       ,110  ,111
 150  DO 250 I=KSTART,KEND

C     CALCULATE CHORD LENGTH
C     XLEAD AND XTRAIL ARE WING LEADING- AND TRAILING-
C     EDGE INTERCEPTS OF CHORD THROUGH CONTROL POINT
C     CLENG IS CHORD LENGTH                                             ,112  ,113
      C(4)=YCON(I,1)                                                    ,114
      CALL POLXN (C,P,M,N1,E,MAX,Q,LOC,NINT)                            ,115
      IF (NINT-1) 480,153,490                                           ,116
  153 XLEAD=Q(1,1)                                                      ,117
      CALL POLXN (C,R,M,N2,E,MAX,Q,LOC,NINT)                            ,118
      IF (NINT-1) 480,157,490                                           ,119
  157 XTRAIL=Q(1,1)                                                     ,120
```

```
      CLENG(1)=XTRAIL-XLEAD                                          ,121
      EPS=CLENG(1)/CONST                                             ,122
      IF (NUMS-1) 500,160,170                                        ,123
  160 IR=1                                                           ,124
      GO TO 190                                                      ,125
  170 IR=I+KSTART-1                                                  ,126
  190 NUM1=XNUM(IR,1)

C     DEFINE ARRAYS FOR AIRFOIL COORDINATES     XTAB1,ZTAB1
C     ARE ARRAYS FOR UPPER SURFACE COORDINATES     XTAB2,
C     ZTAB2 ARE ARRAYS FOR LOWER SURFACE COORDINATES                 ,127
      DO 195 J=1,NUM1                                                ,128  ,129
      XTAB1(J)=XFOIL(IR,J,1)                                         ,130
  195 ZTAB1(J)=ZFOIL(IR,J,1)                                         ,131  ,132
      CALL CHECK (XTAB1,NUM1,2,L,NWRITE)                             ,133
      IF (L) 450,200,450                                             ,134
 200  NUM2=XNUM(IR,2)                                                ,135
      DO 205 J=1,NUM2                                                ,136  ,137
      XTAB2(J)=XFOIL(IR,J,2)                                         ,138
  205 ZTAB2(J)=ZFOIL(IR,J,2)                                         ,139  ,140
      CALL CHECK (XTAB2,NUM2,2,L,NWRITE)                             ,141
      IF (L) 450,207,450

C     CALCULATE CAMBER AND THICKNESS ORDINATES
C     ZCON AND ZCEN ARE Z-CAMBER AND Z-THICKNESS
C     ORDINATES                                                      ,142
  207 DO 220 J=1,NPER1                                               ,143  ,144
      FRAC=(XCON(I,J)-XLEAD)/CLENG(1)                                ,145
      FRAT=(XCEN(I,J)-XLEAD)/CLENG(1)                                ,146
  210 CALL BITURP (XTAB1,ZTAB1,IXY,NUM1,KXY,FRAC,ZR,NC)              ,147
      IF (NC(1)) 460,211,211                                         ,148
 211  ZCU=ZR                                                         ,149
      CALL BITURP (XTAB2,ZTAB2,IXY,NUM2,KXY,FRAC,ZR,NC)              ,150
      IF (NC(1)) 460,212,212                                         ,151
 212  ZCL=ZR                                                         ,152
      ZCON(I,J)=((ZCU+ZCL)/2.)*CLENG(1)+ZHT                          ,153
      CALL BITURP (XTAB1,ZTAB1,IXY,NUM1,KXY,FRAT,ZR,NC)              ,154
      IF (NC(1)) 460,217,217                                         ,155
 217  ZTU=ZR                                                         ,156
      CALL BITURP (XTAB2,ZTAB2,IXY,NUM2,KXY,FRAT,ZR,NC)              ,157
      IF (NC(1)) 460,218,218                                         ,158
 218  ZTL=ZR                                                         ,159
      ZCEN(I,J)=((ZTU-ZTL)/2.)*CLENG(1)+ZHT                          ,160
  220 CONTINUE

C     CALCULATE CAMBER AND THICKNESS SLOPES
C     ALPHA AND THETA ARE ALPHA-CAMBER AND -THICKNESS
C     SLOPES
C     NOTE THAT IF SINGLE AIRFOIL IS INPUT ON
C     CARDS CONSTANT ARRAY FOR ALPHA-CAMBER AND
C     -THICKNESS IS CALCULATED                                       ,161  ,162
      IF (I-KSTART) 500,230,223                                      ,163
 223  IF (NUMS-1) 500,225,230                                        ,164
  225 K=KSTART                                                       ,165
```

```
      DO 227 J=1,NPER1                                                  ,166  ,167
      ALPHA(I,J)=ALPHA(K,J)                                             ,168
  227 THETA(I,J)=THETA(K,J)                                             ,169  ,170
      GO TO 250                                                         ,171
  230 XTAB1(1)=XLEAD                                                    ,172
      ZTAB1(1)=ZFOIL(IR,1,1)*CLENG(1)+ZHT                               ,173
      XTAB2(1)=XLEAD                                                    ,174
      ZTAB2(1)=ZHT                                                      ,175
      DO 231 J=2,NPER                                                   ,176  ,177
      XTAB1(J)=XCON(I,J-1)                                              ,178
      ZTAB1(J)=ZCON(I,J-1)                                              ,179
      XTAB2(J)=XCEN(I,J-1)                                              ,180
  231 ZTAB2(J)=ZCEN(I,J-1)                                              ,181  ,182
      XTAB1(NPERT)=XTRAIL                                               ,183
      ZTAB1(NPERT)=ZFOIL(IR,NUM1,1)*CLENG(1)+ZHT                        ,184
      XTAB2(NPERT)=XTRAIL                                               ,185
      ZTAB2(NPERT)=ZHT                                                  ,186
      DO 240 J=2,NPER                                                   ,187  ,188
      XR=XTAB1(J)-EPS                                                   ,189
      XS=XTAB1(J)+EPS                                                   ,190
  233 CALL BITURP (XTAB1,ZTAB1,IXY,NPERT,KXY,XR,ZR,NC)                  ,191
      IF (NC(1)) 460,234,234                                            ,192
  234 CALL BITURP (XTAB1,ZTAB1,IXY,NPERT,KXY,XS,ZS,NC)                  ,193
      IF (NC(1)) 460,235,235                                            ,194
  235 ALPHA(I,J-1)=ATAN((ZS-ZR)/(XS-XR))                                ,195
      XR=XTAB2(J)-EPS                                                   ,196
      XS=XTAB2(J)+EPS                                                   ,197
  237 CALL BITURP (XTAB2,ZTAB2,IXY,NPERT,KXY,XR,ZR,NC)                  ,198
      IF (NC(1)) 460,238,238                                            ,199
  238 CALL BITURP (XTAB2,ZTAB2,IXY,NPERT,KXY,XS,ZS,NC)                  ,200
      IF (NC(1)) 460,239,239                                            ,201
  239 THETA(I,J-1)=ATAN((ZS-ZR)/(XS-XR))                                ,202
  240 CONTINUE                                                          ,203  ,204
  250 CONTINUE                                                          ,205  ,206
      IF (KSTART-1) 500,257,253                                         ,207
  253 IF (NUMS-1) 500,254,257                                           ,208
  254 K=KSTART                                                          ,209
      DO 255 J=1,NPER1                                                  ,210  ,211
      ALPHA(1,J)=ALPHA(K,J)                                             ,212
  255 THETA(1,J)=THETA(K,J)

C     CALCULATE PANEL SLOPES FOR WING TIP REGION
C     IF OUTBOARD WING EDGE IS NON-STREAMWISE LINE                      ,213  ,214
  257 IF (KEND-NPLN1) 258,510,500                                       ,215
  258 I=NPLN1                                                           ,216
      IF (SLOPE) 260,500,280                                            ,217
  260 DO 270 J=1,NPER                                                   ,218  ,219
      JN=NPER+1-J                                                       ,220
      R(1,J)=XCOR(I+1,JN)                                               ,221
      R(2,J)=YCOR(I+1,JN)                                               ,222
  270 R(3,J)=ZCOR(I+1,JN)                                               ,223  ,224
      GO TO 310                                                         ,225
  280 DO 290 J=1,NPER                                                   ,226  ,227
      P(1,J)=XCOR(I+1,J)                                                ,228
```

```
      P(2,J)=YCOR(I+1,J)                                                ,229
  290 P(3,J)=ZCOR(I+1,J)                                                ,230  ,231
  310 NUM1=XNUM(NUMS,1)

C     DEFINE ARRAYS FOR AIRFOIL COORDINATES     XTAB1,ZTAB1
C     ARE ARRAYS FOR UPPER SURFACE COORDINATES     XTAB2,
C     ZTAB2 ARE ARRAYS FOR LOWER SURFACE COORDINATES                    ,232
      DO 315 J=1,NUM1                                                   ,233  ,234
      XTAB1(J)=XFOIL(NUMS,J,1)                                          ,235
  315 ZTAB1(J)=ZFOIL(NUMS,J,1)                                          ,236  ,237
      CALL CHECK (XTAB1,NUM1,2,L,NWRITE)                                ,238
      IF (L) 450,320,450                                                ,239
  320 NUM2=XNUM(NUMS,2)                                                 ,240
      DO 325 J=1,NUM2                                                   ,241  ,242
      XTAB2(J)=XFOIL(NUMS,J,2)                                          ,243
  325 ZTAB2(J)=ZFOIL(NUMS,J,2)                                          ,244  ,245
      CALL CHECK (XTAB2,NUM2,2,L,NWRITE)                                ,246
      IF (L) 450,327,450                                                ,247
  327 DO 380 J=1,NPER1

C     CALCULATE CHORD LENGTH
C     XLEAD AND XTRAIL ARE WING LEADING- AND TRAILING-
C     EDGE INTERCEPTS OF CHORD THROUGH CONTROL POINT
C     CLENG IS CHORD LENGTH                                             ,248  ,249
      C(4)=YCON(I,J)                                                    ,250
      IF (SLOPE) 330,500,340                                            ,251
  330 NL=N1                                                             ,252
      NT=NPER                                                           ,253
      GO TO 350                                                         ,254
  340 NL=NPER                                                           ,255
      NT=N2                                                             ,256
  350 CALL POLXN (C,P,M,NL,E,MAX,Q,LOC,NINT)                            ,257
      IF (NINT-1) 480,360,490                                           ,258
  360 XLEAD=Q(1,1)                                                      ,259
      IF (J-1) 500,363,364                                              ,260
  363 XLS=XLEAD                                                         ,261
  364 CALL POLXN (C,R,M,NT,E,MAX,Q,LOC,NINT)                            ,262
      IF (NINT-1) 480,367,490                                           ,263
  367 XTRAIL=Q(1,1)                                                     ,264
      IF (J-NPER1) 370,368,500                                          ,265
  368 XTS=XTRAIL                                                        ,266
      CLS=XTS-XLS                                                       ,267
  370 CLENG(J)=XTRAIL-XLEAD

C     CALCULATE CAMBER AND THICKNESS ORDINATES
C     ZCON AND ZCEN ARE Z-CAMBER AND Z-THICKNESS
C     ORDINATES                                                         ,268
      FRAC=(XCON(I,J)-XLEAD)/CLENG(J)                                   ,269
      FRAT=(XCEN(I,J)-XLEAD)/CLENG(J)                                   ,270
  371 CALL BITURP (XTAB1,ZTAB1,IXY,NUM1,KXY,FRAC,ZR,NC)                 ,271
      IF (NC(1)) 460,372,372                                            ,272
  372 ZCU=ZR                                                            ,273
      CALL BITURP (XTAB2,ZTAB2,IXY,NUM2,KXY,FRAC,ZR,NC)                 ,274
      IF (NC(1)) 460,373,373                                            ,275
```

```
 373  ZCL=ZR                                                          ,276
      ZCON(I,J)=((ZCU+ZCL)/2.)*CLENG(J)+ZHT                           ,277
      CALL BITURP (XTAB1,ZTAB1,IXY,NUM1,KXY,FRAT,ZR,NC)               ,278
      IF (NC(1)) 460,377,377                                          ,279
 377  ZTU=ZR                                                          ,280
      CALL BITURP (XTAB2,ZTAB2,IXY,NUM2,KXY,FRAT,ZR,NC)               ,281
      IF (NC(1)) 460,378,378                                          ,282
 378  ZTL=ZR                                                          ,283
      ZCEN(I,J)=((ZTU-ZTL)/2.)*CLENG(J)+ZHT                           ,284
  380 CONTINUE

C     CALCULATE CAMBER AND THICKNESS SLOPES
C     ALPHA AND THETA ARE ALPHA-CAMBER AND -THICKNESS
C     SLOPES
C     NOTE THAT IF SINGLE AIRFOIL IS INPUT ON
C     CARDS CONSTANT ARRAY FOR ALPHA-CAMBER AND
C     -THICKNESS IS CALCULATED                                        ,285  ,286
      IF (NUMS-1) 500,383,387                                         ,287
  383 K=KSTART                                                        ,288
      L=KEND                                                          ,289
      DO 385 J=1,NPER1                                                ,290  ,291
      ALPHA(L,J)=ALPHA(K,J)                                           ,292
  385 THETA(L,J)=THETA(K,J)                                           ,293  ,294
      GO TO 510                                                       ,295
  387 XTAB1(1)=XLS                                                    ,296
      ZTAB1(1)=ZFOIL(NUMS,1,1)*CLS+ZHT                                ,297
      XTAB2(1)=XLS                                                    ,298
      ZTAB2(1)=ZHT                                                    ,299
      DO 390 J=2,NPER                                                 ,300  ,301
      XTAB1(J)=XCON(I,J-1)                                            ,302
      ZTAB1(J)=ZCON(I,J-1)                                            ,303
      XTAB2(J)=XCEN(I,J-1)                                            ,304
 390  ZTAB2(J)=ZCEN(I,J-1)                                            ,305  ,306
      XTAB1(NPERT)=XTS                                                ,307
      ZTAB1(NPERT)=ZFOIL(I,NUM1,1)*CLS+ZHT                            ,308
      XTAB2(NPERT)=XTS                                                ,309
      ZTAB2(NPERT)=ZHT                                                ,310
 400  DO 410 J=2,NPER                                                 ,311  ,312
      EPS=CLENG(J-1)/CONST                                            ,313
      XR=XTAB1(J)-EPS                                                 ,314
      XS=XTAB1(J)+EPS                                                 ,315
 403  CALL BITURP (XTAB1,ZTAB1,IXY,NPERT,KXY,XR,ZR,NC)                ,316
      IF (NC(1)) 460,404,404                                          ,317
 404  CALL BITURP (XTAB1,ZTAB1,IXY,NPERT,KXY,XS,ZS,NC)                ,318
      IF (NC(1)) 460,405,405                                          ,319
 405  ALPHA(I,J-1)=ATAN((ZS-ZR)/(XS-XR))                             ,320
      XR=XTAB2(J)-EPS                                                 ,321
      XS=XTAB2(J)+EPS                                                 ,322
 407  CALL BITURP (XTAB2,ZTAB2,IXY,NPERT,KXY,XR,ZR,NC)                ,323
      IF (NC(1)) 460,408,408                                          ,324
 408  CALL BITURP (XTAB2,ZTAB2,IXY,NPERT,KXY,XS,ZS,NC)                ,325
      IF (NC(1)) 460,409,409                                          ,326
 409  THETA(I,J-1)=ATAN((ZS-ZR)/(XS-XR))                              ,327
  410 CONTINUE                                                        ,328  ,329
```

```
  420 GO TO 510

C     ERROR MESSAGES                                                    ,330
  450 WRITE (NWRITE,940)                                                ,331  ,332
      GO TO 495                                                         ,333
 460  WRITE (NWRITE,930)                                                ,334  ,335
      GO TO 495                                                         ,336
  470 WRITE (NWRITE,900)                                                ,337  ,338
      GO TO 495                                                         ,339
  480 WRITE (NWRITE,910)                                                ,340  ,341
      GO TO 495                                                         ,342
  490 WRITE (NWRITE,920)                                                ,343  ,344
      GO TO 495                                                         ,345
 495  KODEW=1                                                           ,346
      GO TO 500                                                         ,347
  500 RETURN

C     USE SUBROUTINE MEAN TO SMOOTH CAMBER AND
C     THICKNESS SLOPES                                                  ,348
 510  IF (NUMS-1) 500,520,550                                           ,349
 520  K=KSTART                                                          ,350
      DO 530 J=1,NPER1                                                  ,351  ,352
      XTAB1(J)=XCON(K,J)                                                ,353
      ZTAB1(J)=ALPHA(K,J)                                               ,354
      XTAB2(J)=XCEN(K,J)                                                ,355
 530  ZTAB2(J)=THETA(K,J)                                               ,356  ,357
 533  CALL MEAN (XTAB1,ZTAB1,NPER1)                                     ,358
 537  CALL MEAN (XTAB2,ZTAB2,NPER1)                                     ,359
      DO 540 I=1,NPLN1                                                  ,360  ,361
      DO 540 J=1,NPER1                                                  ,362  ,363
      ALPHA(I,J)=ZTAB1(J)                                               ,364
 540  THETA(I,J)=ZTAB2(J)                                               ,365  ,366  ,367
      GO TO 500                                                         ,368
 550  DO 580 I=1,NPLN1                                                  ,369  ,370
      DO 560 J=1,NPER1                                                  ,371  ,372
      XTAB1(J)=XCON(I,J)                                                ,373
      ZTAB1(J)=ALPHA(I,J)                                               ,374
      XTAB2(J)=XCEN(I,J)                                                ,375
 560  ZTAB2(J)=THETA(I,J)                                               ,376  ,377
 563  CALL MEAN (XTAB1,ZTAB1,NPER1)                                     ,378
 567  CALL MEAN (XTAB2,ZTAB2,NPER1)                                     ,379
      DO 570 J=1,NPER1                                                  ,380  ,381
      ALPHA(I,J)=ZTAB1(J)                                               ,382
 570  THETA(I,J)=ZTAB2(J)                                               ,383  ,384
 580  CONTINUE                                                          ,385  ,386
      GO TO 500                                                         ,387
  900 FORMAT(1H1,9X,33HERROR MESSAGE - SUBROUTINE SLOPEW/10X,
     119HPROGRAM LOGIC ERROR/10X,
     234HINCORRECT COMPUTED GO TO STATEMENT)
  910 FORMAT(1H1,9X,33HERROR MESSAGE - SUBROUTINE SLOPEW/10X,
     124HSUBROUTINE POLXN FAILURE/10X,
     255HWING CHORD LENGTH NOT CALCULATED  NO INTERSECTION FOUND)
  920 FORMAT(1H1,9X,33HERROR MESSAGE - SUBROUTINE SLOPEW/10X,
     124HSUBROUTINE POLXN FAILURE/10X,
```

```
   262HWING CHORD LENGTH NOT CALCULATED  MULTIPLE INTERSECTIONS FOUND)
930 FORMAT(1H1,9X,33HERROR MESSAGE - SUBROUTINE SLOPEW/10X,
   125HSUBROUTINE BITURP FAILURE/10X,
   233HINTERPOLATED VALUE NOT CALCULATED)
940 FORMAT(1H1,9X,33HERROR MESSAGE - SUBROUTINE SLOPEW/10X,
   121HINPUT CARD DATA ERROR/10X,
   147HX-COORDINATES OF AIRFOIL NOT STRICTLY MONOTONIC/10X,
   219HINCREASING SEQUENCE)
     END                                                          .388
```

```
      SUBROUTINE TDUMP(LO,LTAPE,LGDEF)
      DIMENSION R(2000),IR(10)
      LOGICAL LGDEF(3,6)
C
C     USED BY GEOMD TO DUMP CONTENTS OF PERCENT LINE TAPE
C
      DO 10 M=1,6                                                    ,1
      IF(LGDEF(3,M)) GO TO 50                                        ,2    ,3    ,4
 10   CONTINUE                                                       ,5    ,6
      GO TO 9000
C                                                                    ,7
 50   WRITE(LO,100)                                                  ,8    ,9
 100  FORMAT(10H1TAPE DUMP)
      REWIND LTAPE                                                   ,10
      JR=0                                                           ,11
      DO 1000 K=1,9                                                  ,12
      JR=JR+1                                                        ,13
      READ (LTAPE) IR                                                ,14   ,15   ,16
      WRITE(LO,200)JR,IR                                             ,17   ,18   ,19
 200  FORMAT(7HORECORD I3/10I6)
      JR=JR+1                                                        ,20
      L=IR(2)                                                        ,21
      READ(LTAPE)(R(I),I=1,L)                                        ,22   ,23   ,24   ,25   ,26
      WRITE(LO    ,300)JR,(R(I),I=1,L)                               ,27   ,28   ,29   ,30   ,31   ,32

 300  FORMAT(7HORECORD I3/(1H 9F13.4))
 1000 CONTINUE                                                       ,33   ,34
 9000 RETURN                                                         ,35
      END                                                            ,36
```

```
      SUBROUTINE TFLAT(DAT,LI,LO,LA,LB,LD,NBODY,IOK)
C
C     USED BY OPCAM TO TRANSFORM BODY AND WING TO A NEW
C     COORDINATE SYSTEM, FLATTEN THE WING, AND INTERSECT
C     THE TRANSFORMED BODY WITH THE TRANSFORMED AND FLATTENED WING
C
      DIMENSION DAT(2),STA(51),OLDSTA(51),
     , B(4368),RT(3,4),RX(3,51),W(4368),NU(3),BCODE(2),S(1000),
     , BW(3,100),ALFBET(2),NAME(3),RZ(51,2)
     ,,IREC(10),JREC(10),DUM(9)
     ,,BTITL(12),WTITL(12),BAX(2)
      LOGICAL LGDEF,BODY,WING
      COMMON /LGEOMD/LGDEF(3,6)
      DATA IREC,JREC,DUM,NU/10*0,1,9,1,7*0,9*0.,0,6, 30/,
     , NAME/6HDEFINE,6HDEFINE,6HEND OF/
     , ,BAX/0.,0./,BTITL(1)/
     ,72HTRANSFORMED BODY (PRIMED-AXIS SYSTEM)
     ,          /,WTITL(1)/
     ,72HPOINTS ON FLAT TRANSFORMED WING (PRIMED-AXIS SYSTEM)
     ,          /
C INPUTS ARE
C     DAT = DATE
C     LI = INPUT TAPE
C     LO = OUTPUT TAPE
C     LA = TAPE WRITTEN BY GEOMETRIC DEFINITION LINK, AND
C          USED BY TFLAT
C     LB = TAPE TO BE WRITTEN FOR AERO LINK
C     LD = TAPE TO BE WRITTEN FOR PANEL LINK
C OUTPUT IS
C     IOK = 0 IF TFLAT IS SUCCESSFUL
C
 100  WRITE(LO,110)DAT                                                  ,1     ,2     ,3
 110  FORMAT(1H1 55X 2A6/68H PROGRAM ENTERED LINK TO TRANSFORM BODY AND/
     ,OR WING BEFORE PANELLING)

C     CLEAR STORAGE (PROGRAM DOES NOT REQUIRE THIS, BUT HELPFUL FOR DUMPS)
      DO 120 I=1,4368                                                   ,4
      B(I)=-0.                                                          ,5
 120  W(I)=-0.                                                          ,6     ,7
      CALL TRAV(RX,B,153)                                               ,8
      CALL TRAV(S,B,1000)                                               ,9
      CALL TRAV(BW,B,300)                                               ,10
      CALL TRAV(RZ,B,102)
                                                                        ,11
      NU(2)=LO                                                          ,12
      NBODY=0                                                           ,13
      REWIND LA                                                         ,14
      REWIND LD                                                         ,15
      READ(LI,210) XNRX,CHD,BCODE,EPRC                                  ,16    ,17    ,18
 210  FORMAT(6F10.0)
      NRX=XNRX                                                          ,19
      KODEBR=BCODE(1)                                                   ,20
      KODEBS=BCODE(2)
```

```
C     NRX = NO. OF POINTS DESIRED IN RX ARRAY (RADIUS AND Z VS. X)
C       LINEAR INTERPOLATION, =2 FOR BIQUADRATIC INTERPOLATION.
C     KODEBR IS USED BY BITURP TO GET RX ARRAY
C     KODEBS IS USED BY WBXX
C     CHD IS USED BY TFLAT1
C     EPRC IS USED BY TFLAT1.  Z-CENTROID AND RADIUS SET =0 IF LESS THAN EPRC.
C                                                                          ,21
      MU = 2                                                               ,22
      IF(NRX .GT. 2 .AND. NRX .LT. 50) GO TO 400                           ,23   ,24   ,25
      NRX=50                                                               ,26
 400  CONTINUE
C
C     TEST FOR ERROR IN PREVIOUS LINK                                      ,27
      BODY=.FALSE.                                                         ,28
      WING=.FALSE.                                                         ,29
      IF(.NOT. LGDEF(1,3)) GO TO 500
C     BODY WAS REQUESTED.  SEE IF PROCESSED.                               ,30   ,31   ,32
      IF(.NOT. LGDEF(3,3)) GO TO 700                                       ,33   ,34   ,35
      BODY= .TRUE.
C     CHECK WING                                                           ,36
 500  IF(.NOT. LGDEF(1,1)) GO TO 600
C     UPPER WING REQUESTED.  SEE IF PROCESSED.                             ,37   ,38   ,39
      IF(.NOT. LGDEF(3,1)) GO TO 700                                       ,40   ,41   ,42
      WING= .TRUE.
C     CHECK THAT AT LEAST A BODY OR A WING IS DEFINED                      ,43
 600  IF(BODY .OR. WING) GO TO 800
C                                                                          ,44   ,45   ,46
 700  WRITE(LO,710)                                                       ,47   ,48
 710  FORMAT(23HOERROR IN PREVIOUS LINK)
      GO TO 8000
C                                                                          ,49
 800  IF(BODY) GO TO 1000
C     NO BODY, BUT THERE IS A WING.  WRITE DUMMY RECORDS FOR BODY
C       MERIDIAN LINES ON TAPE LD.                                         ,50   ,51   ,52
      WRITE(LD)JREC                                          LD 1        ,53   ,54   ,55
      WRITE(LD)DUM                                           LD 2
C     SKIP RECORDS 1-4 ON TAPE LA                                          ,56   ,57   ,58
      CALL FSR(4,LA,IOK)                                     LA 1-4      ,59
      GO TO 3000
C
C     READ BODY DEFINITION FROM TAPE LA                                    ,60
 1000 CONTINUE                                                             ,61
      READ(LA)IREC                                           LA 1        ,62   ,63   ,64
      NSTA=IREC(2)                                                        ,65
      MU = 4                                                              ,66
      IF((NSTA .LT. 2) .OR. (NSTA .GT. 51) .OR. (IREC(1) .NE. 1) .OR.
     , (IREC(3) .NE. 0))  GO TO 8000                                      ,67   ,68   ,69
      READ(LA)(STA(I),I=1,NSTA)                              LA 2
C     SAVE ORIGINAL BODY STATIONS FOR LATER PRINTOUT                       ,70   ,71   ,72   ,73   ,74
      CALL TRAV(OLDSTA,STA,NSTA)                                          ,75
      READ(LA)IREC                                           LA 3        ,76   ,77   ,78
      MU = 6                                                              ,79
      IF(IREC(1) .NE. 3 .OR. IREC(3) .NE. 0 .OR. IREC(2) .GT. 4368)
     , GO TO 8000                                                         ,80   ,81   ,82
```

```
              EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT    -   INTERNAL FORMULA NUMBER(S)

      NBPCT=IREC(4)
C     NBPCT = NO. OF BODY MERIDIAN LINES                                          ,83
      NBEL=IREC(2)                                                                ,84
      READ(LA)(B(I),I=1,NBEL)                                   LA 4
C
C     FIND TRANSFORMATION RT TO NEW COORD. SYSTEM.
C     TRANSFORM AND SHEAR BODY MERIDIAN LINES B
C     TRANSFORM BODY STATIONS
C     FIND RADIUS AND Z-COORD. OF CENTROID VS. X-PRIME                            ,85   ,86   ,87   ,88   ,89
 2000 MU = 8                                                                      ,90
      CALL TFLAT1(CHD,NSTA,STA,NBPCT,B,S,W,RT,ALFBET,RZ,EPRC,NU)                  ,91
      IF(NU)8000,2050,8000                                                        ,92
 2050 CONTINUE
C                                                                                 ,93
      WRITE(LO,2100)(RT(I,4),I=1,3),(RT(I,1),I=1,3),
     , (I,OLDSTA(I),STA(I),RZ(I,1),RZ(I,2),I=1,NSTA)                              ,94   ,95   ,96   ,97   ,98   ,99
                                                                                  ,100  ,101  ,102  ,103  ,104
 2100 FORMAT(51HOTHE ORIGIN OF THE NEW BODY COORDINATE SYSTEM IS AT /
     , 5X12H(X,Y,Z) =  ( F10.4,1H,F10.4,1H,F10.4,1H)/
     , 52HOA UNIT VECTOR ALONG THE X-PRIME AXIS HAS COMPONENTS /
     , 5X12H(X,Y,Z) =  ( F10.4,1H,F10.4,1H,F10.4,1H)/
     , /1HO 12X 8HORIGINAL 5X 12HCIRCULARIZED 7X 4HBODY 8X 6HZ-BODY
     , /6X 3HNO. 2X 12HBODY STATION 3X 12HBODY STATION 6X 6HRADIUS
     , 6X 8HCENTROID//(1H I7,F13.4,F15.4,2F14.4))
C
C     WRITE TRANSFORMED BODY MERIDIAN LINES ON OUTPUT TAPE
      CALL TFLATM(B,NBPCT,BAX,BTITL,DAT,LO,STA)
C
C     WRITE BODY RECORDS ON TAPE LD                                               ,105
      IREC(1)=1                                                                   ,106
      WRITE(LD)IREC                                             LD 1              ,107  ,108  ,109
      WRITE(LD)(B(I),I=1,NBEL)                                  LD 2
C
C     BODY DONE.  PROCESS WING.
C                                                                                 ,110  ,111  ,112  ,113  ,114
      IF(WING) GO TO 3000
C     NO. WING.  WRITE DUMMY WING RECORDS ON TAPE LD                              ,115  ,116  ,117
      JREC(3)=1                                                                   ,118
      DO 2500 I=3,5,2                                                             ,119
      JREC(1)=I                                                                   ,120
      WRITE(LD)JREC                                             LD 3,5            ,121  ,122  ,123
 2500 WRITE(LD)DUM                                              LD 4,6            ,124  ,125  ,126  ,127
      ZAV=0.                                                                      ,128
      GO TO 5000
C
C     WING IS TO BE PROCESSED.  READ WING DEFINITION FROM TAPE LA                 ,129
 3000 CONTINUE                                                                    ,130
      CALL FSR(2,LA,IOK)                                        LA 5-6            ,131
      READ(LA)IREC                                              LA 7              ,132  ,133  ,134
      MU = 10                                                                     ,135
      IF(IREC(1) .NE. 7 ) GO TO  8000                                             ,136  ,137  ,138
      NWPCL=IREC(4)                                                               ,139
      NWEL=IREC(2)
C     NWPCL = NO. OF WING PERCENT LINES                                           ,140
```

```
      MU = 12                                                           ,141
      IF(NWEL .GT. 4368) GO TO 8000
C     RECORD 3 ON LD IS SAME AS 7 ON LA (EX. WORD 1).  WRITE IT.        ,142  ,143  ,144
      IREC(1)=3                                                         ,145
      WRITE(LD)IREC                                                LD 3
C                                                                       ,146  ,147  ,148
      READ(LA) (W(I),I=1,NWEL)                                     LA 8
C
C     WE DO NOT TRANSFORM WING IF THERE IS NO BODY                      ,149  ,150  ,151  ,152  ,153
      IF(.NOT. BODY) GO TO 4000
C
C     TRANSFORM WING                                                    ,154  ,155  ,156
 3050 CALL TRAPCT(W,NWPCL,RT)
C
C     FIND WING-BODY INTERSECTIONS                                      ,157
 3100 MU = 14                                                           ,158
      CALL WBXX(B,NBPCT,W,NWPCL,1.E-4,KODEBS,S,BW,NU)                   ,159
      IF(NU)8000,3200,8000
C     INTERSECTION POINTS ARE IN BW.  USE AVERAGE OF Z FOR
C     FIRST AND LAST INTERSECTION POINTS.                               ,160
 3200 ZAV=(BW(3,1)+BW(3,NWPCL))*.5
C     CHECK THAT BOTH INTERSECTION POINTS WERE OK.  IF ANY WING
C     PERCENT LINE FAILED TO PIERCE BODY, THEN COORD. OF 1.E+30
C     WAS INSERTED (BY WBX) FOR X,Y AND Z.                              ,161
      MU = 16                                                           ,162
      IF(ZAV-1.E+5)4200,8000,8000
C
C     THIS IS WING WITHOUT BODY.  FOR HEIGHT OF FLAT WING
C     USE AVERAGE OF Z FOR 1ST POINT OF 1ST PERCENT LINE AND
C     1ST PT. OF LAST LINE.                                             ,163
 4000 CONTINUE                                                          ,164
      MU = 18                                                           ,165
      JZAV=W(3)                                                         ,166
      ZAVZER=W(JZAV+2)                                                  ,167
      KZAV=W(3*NWPCL)                                                   ,168
      ZAVHUN=W(KZAV+2)                                                  ,169
      ZAV=.5*(ZAVZER+ZAVHUN)
C
C     THE FLAT WING WILL HAVE Z=ZAV FOR ALL POINTS                      ,170
 4200 CONTINUE                                                          ,171
      MU = 20                                                           ,172
      J=3*NWPCL+3                                                       ,173
      DO 4220 I=J,NWEL,3                                                ,174
 4220 W(I)=ZAV
C
C     WRITE FLAT, TRANSFORMED WING ON OUTPUT TAPE                       ,175  ,176
      CALL TFLATW(W,NWPCL,WTITL,DAT,LO)
C
C     WRITE FLAT WING PERCENT LINES ON LD                               ,177
      WRITE(LD) (W(I),I=1,NWEL)                                    LD 4
C                                                                       ,178  ,179  ,180  ,181  ,182
      IF(BODY) GO TO 4500
C     WING ALONE.  WRITE DUMMY RECORDS ON LD FOR WING-BODY
C     INTERSECTION POINTS.                                              ,183  ,184  ,185
```

```
      MU = 22                                                           ,186
      JREC(1)=5                                                         ,187
      WRITE(LD)JREC                                          LD 5       ,188  ,189  ,190
      WRITE(LD)DUM                                           LD 6       ,191  ,192  ,193
      IOK=0                                                             ,194
      GO TO 5000
C
C     FIND INTERSECTIONS OF FLAT WING WITH BODY                         ,195
 4500 CONTINUE                                                          ,196
      MU = 24                                                           ,197
      CALL WBXX(B,NBPCT,W,NWPCL,1.E-4,KODEBS,S,BW,NU)                   ,198
      IF(NU)8000,4700,8000
C
C     WRITE WING-BODY INTERSECTION POINTS ON OUTPUT TAPE                ,199
 4700 CALL TFLATX(LO,DAT,NWPCL,W,BW,NU)                                 ,200
      MU = 26                                                           ,201
      IF(NU)8000,4800,8000
C
C     WRITE INTERSECTICN POINTS ON LD                                   ,202
 4800 MU = 28                                                           ,203
      IREC(1)=5                                                         ,204
      IREC(2)=3*NWPCL                                                   ,205
      IREC(3)=0                                                         ,206
      WRITE(LD)IREC                                          LD 5       ,207  ,208  ,209
      WRITE(LD) ((BW(I,J),I=1,3),J=1,NWPCL)                  LD 6
C                                                                       ,210  ,211  ,212  ,213  ,214
                                                                        ,216  ,217
 5000 CONTINUE                                                          ,218
      END FILE LD                                                       ,219
      REWIND LD
C                                                                       ,220
 6000 IF(.NOT. BODY)GO TO 9000
C
C     WRITE TAPE LB FOR AERO AND PANEL LINKS.                           ,221  ,222  ,223
      REWIND LB
C     RECORD 1 = XYZ CCMPONENTS OF UNIT VECTOR ALONG X-PRIME AXIS,
C           FOLLOWED BY ZAV = Z-COORD. OF FLAT WING,
C           FOLLOWED BY TRANSFORMATION CONSTANTS TO CONVERT OLD
C           STATIONS TO NEW STATIONS -
C               NEW X=ALFBET(1)*(OLD X)+ALFBET(2)
C                                                                       ,224
      WRITE(LB)(RT(I,1),I=1,3),ZAV,ALFBET                    LB 1
C
C     PREPARE TO FIND POINTS ON CURVE OF BODY RADIUS VS. X,
C     AND CURVE OF Z-COORD. OF CENTROID VS. X.
C     THERE ARE NX POINTS.  IF THERE IS NO WING,THEN
C     THE PTS. ARE EQUALLY SPACED IN X.  IF THERE IS A WING
C     THEN THE POINTS WILL BE EQUALLY SPACED FROM NOSE TO
C     INTERSECTION OF LEADING EDGE WITH BODY, AND EQUALLY
C     SPACED FROM THERE ON AFT.  THESE TWO SPACINGS
C     WILL BE APPROX. EQUAL.
C                                                                       ,225  ,226  ,227  ,228  ,229

      MU = 30                                                           ,231
```

```
                EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      BLENG=STA(NSTA)                                                  ,232
      IF(WING)GO TO 6100
C                                                                      ,233  ,234  ,235
      XNRX=NRX-1                                                       ,236
      DO 6050 I=1,NRX                                                  ,237
      XIM=I-1                                                          ,238
 6050 RX(1,I)=(XIM/XNRX)*BLENG                                         ,239  ,240
      INDEX=NRX                                                        ,241
      GO TO 6150
C                                                                      ,242
 6100 BLN=BW(1,1)                                                      ,243
      BLR=BLENG-BLN                                                    ,244
      NRXN=(BLN/BLENG)*FLOAT(NRX-1)+1.5                                ,245
      NRXR=NRX-NRXN+1                                                  ,246
      XNRX=NRXN-1                                                      ,247
      DO 6110 I=1,NRXN                                                 ,248
      XIM=I-1                                                          ,249
 6110 RX(1,I)=(XIM/XNRX)*BLN                                           ,250  ,251
      XNRX=NRXR-1                                                      ,252
      DO 6120 I=2,NRXR                                                 ,253
      XIM=I-1                                                          ,254
      J=NRXN+I-1                                                       ,255
 6120 RX(1,J)=BLN+(XIM/XNRX)*BLR                                       ,256  ,257
      INDEX=NRXN
C                                                                      ,258
 6150 CONTINUE
C
C     THE X VALUES ARE NOW IN THE RX ARRAY.
C     THE RADIUS FOR EACH GIVEN STATION STARTS IN RZ(1,1).
C     THE Z-COORD. OF CENTROID FOR EACH STA STARTS IN RZ(1,2)
C     INTERPOLATE TO GET R,Z CORRESPONDING TO X-VALUES IN RX.
C                                                                      ,259
      MU = 32                                                          ,260
      DO 6300 I=1,NRX
C                                                                      ,261
      DO 6250 J=1,2                                                    ,262
      CALL BITURP(STA,RZ(1,J),1,NSTA,KCDEBR,RX(1,I),RX(J+1,I),NU)      ,263
      IF(NU)8000,6250,6200                                             ,264
 6200 IF(NU-2)6250,6250,8000                                           ,265
 6250 CONTINUE
C                                                                      ,266  ,267
 6300 CONTINUE
C     WRITE RECORDS 2,3                                                ,268  ,269
      WRITE(LB)NRX,INDEX                                   LB 2        ,270  ,271  ,272
      WRITE(LB) ((RX(I,J),I=1,3),J=1,NRX)                  LB 3        ,273  ,274  ,275  ,276  ,277  ,277
                                                                       ,279  ,280
      END FILE LB                                                      ,281
      REWIND LB
C                                                                      ,282
      WRITE(LO,6400) DAT,(J,(RX(I,J),I=1,3),J=1,NRX)                   ,283  ,284  ,285  ,286  ,287  ,287
                                                                       ,289  ,290  ,291  ,292
 6400 FORMAT(1H1 53X 2A6/38H BODY RADIUS AND Z-COORDINATE OF BODY
     ,23HCENTROID VERSUS X-PRIME/5H0 NO. 10X 7HX-PRIME 18X
     ,6HRADIUS 13X 7HZ-PRIME /(1H I3,F18.4,F24.4,F20.4))
```

```
C
C
      IOK=0                                                             ,293
      IF(BODY .AND. .NOT. WING) NBODY=-1                                ,294  ,295  ,296
      GO TO 9000
C
C     ERROR.  SKIP OVER DATA CARDS TO NEXT CASE, OR
C     UNTIL   END OF DATA   CARD IS FOUND                               ,297
 8000 CONTINUE                                                          ,298
      IOK=MU                                                            ,299
      IF(MU-2)8100,8100,8005                                            ,300
 8005 WRITE(LO,8010)MU,NU(1),(IREC(I),I=1,4)                            ,301  ,302  ,303  ,304  ,305  ,306

 8010 FORMAT(/12HOTFLAT ERROR I3,7H,  CODE I3,
     1 /7H IREC = 4I10)
      IF(MU .LT. 14  .OR.  MU .GT. 18) GO TO 8100                       ,307  ,308  ,309
      WRITE(LO,8020)                                                    ,310  ,311
 8020 FORMAT(45HOTRANSFORMED WING (BEFORE FLATTENING) FOLLOWS )
      CALL TFLATW(W,NWPCL,WTITL,DAT,LO)
C                                                                       ,312
 8100 READ(LI,8110)KIND                                                 ,313  ,314  ,315
 8110 FORMAT(A6)
      DO 8200 I=1,3                                                     ,316
      IF(KIND-NAME(I))8200,8300,8200                                    ,317
 8200 CONTINUE                                                          ,318  ,319
      GO TO 8100
C                                                                       ,320
 8300 BACKSPACE LI
C                                                                       ,321
 9000 CONTINUE                                                          ,322
      DO 9100 J=1,6                                                     ,323
      DO 9100 I=1,3                                                     ,324
 9100 LGDEF(I,J) = .FALSE.                                              ,325  ,326  ,327
      RETURN                                                            ,328
      END                                                               ,329
```

```
      SUBROUTINE TELAT1(CHD,NSTA,STA,NBPCT,B,S,E,RT,ALFBET,RZ,EPRC,NU)
C
      DIMENSION RZ(51,2),STA(51),B(2448),RT(3,4),ALFBET(2),NU(3),S(100,2
     ,), E(2000),EP(5),BEND(2,2),AREA(2),RAD(2),UNITZ(3)
C
C     USED BY TELAT TO TRANSFORM BODY MERIDIAN LINES TO A
C       NEW COORDINATE SYSTEM WITH ORIGIN AT THE CENTROID OF
C       THE FIRST BODY SECTION.  THE NEW +X AXIS PASSES THRU
C       THE CENTROID OF THE LAST BODY SECTION.
C       IT IS ASSUMED THAT BODY MERIDIAN LINES ARE GIVEN FOR A HALF-BODY
C         AND THAT THE BODY IS SYMMETRICAL ABOUT A PLANE PARALLEL TO THE
C         XZ PLANE.
C     TRANSFORMED BODY MERIDIAN LINES ARE SCALED (INDIVIDUALLY) SO
C       THAT 1ST PT IN EACH LINE HAS X=0 AND EACH LINE HAS SAME
C       X-LENGTH (LENGTH BETWEEN END CENTROIDS).
C INPUTS ARE
C     CHD = CHORD-HEIGHT TOLERANCE.  TO FIND THE CENTROID
C       OF A BODY SECTION, WE CUT EACH MERIDIAN LINE WITH THE
C       X-VALUE OF THE SECTION.  THIS GIVES US A POLYGON OF THE
C       SECTION IN 2-DIMENSIONS (Y,Z).  IF CHD .GT. 0, WE THEN
C       INTERPOLATE ADDITIONAL POINTS ON THE POLYGON TO BETTER
C       APPROXIMATE A SMOOTH BODY.  CHD IS USED BY S/R ENRYCH.
C     NSTA = NO. OF STATIONS
C     NBPCT = NO. OF BODY MERIDIAN LINES
C INPUT-OUTPUTS ARE
C     STA = ARRAY OF BODY STATIONS.  ON OUTPUT, SCALED TO AGREE
C       WITH TRANSFORMED BODY.
C     B = ARRAY OF BODY MERIDIAN LINES (WITH HEADER).
C     EPRC IS USED BY BODCR TO SET Z-CENTROID AND RADIUS =0 IF .LE. EPRC.
C OUTPUTS ARE
C     RT = ROTATION, TRANSLATION MATRIX OF THE TRANSFORMATION
C       FROM ORIGINAL SYSTEM TO NEW SYSTEM.
C       COL. 1 IS UNIT VECTOR ALONG NEW +X AXIS
C       COL. 2 IS UNIT VECTOR ALONG NEW +Y AXIS
C       COL. 3 IS UNIT VECTOR ALONG NEW +Z AXIS
C       COL. 4 IS ORIGIN OF NEW XYZ SYSTEM
C              ALL WITH RESPECT TO ORIGINAL XYZ SYSTEM
C     ALFBET = PAIR OF CONSTANTS USED TO RELATE OLD STATION
C       VALUES TO NEW.
C       NEW X = ALFBET(1)*(OLD X)+ ALFBET(2)
C     RZ = ARRAYS OF BODY RADIUS AND Z-COORD. OF CENTROID AT
C       EACH BODY STATION
C SCRATCH ARRAYS ARE  S, E
C
      DATA EP/5*.005/
      EP(5)=ABS(EPRC)
C
C     BEND (BODY ENDS) ARRAY WILL BE COORD. OF CENTROIDS
C     OF THE END SECTIONS OF THE BODY                                   ,1
 100  BEND(1,1)=STA(1)                                                  ,2
      BEND(1,2)=STA(NSTA)
C
C     FIND EQUIV. RADIUS AND CENTROID OF BODY END SECTIONS              ,3
```

```
      DO 200 I=1,2                                                      ,4
      MU=I                                                              ,5
      CALL BODCR(B,NBPCT,BEND(1,I),EP,CHD,2000,S(1,I),E,NE,
     , BEND(2,I),BEND(3,I),AREA(I),RAD(I),NU)                           ,6
      IF(NU)8000,200,8000                                               ,7
 200  CONTINUE
C
C     FIND TRANSFORMATION MATRICES TO NEW SYSTEM                        ,8    ,9
 300  BLENG=UVECN(BEND,BEND(1,2),RT,0,3)                                ,10
      DATA UNITZ/0.,0.,1./
      IZXP=1                                                            ,11
      CALL VCROS(UNITZ,RT,RT(1,2),DZXP,IZXP)                            ,12
      IXPYP=1                                                           ,13
      CALL VCROS(RT,RT(1,2),RT(1,3),DXPYP,IXPYP)                        ,14
      MU=3                                                              ,15
      IF(BLENG .EQ. 0. .OR. IZXP .NE. 0 .OR. IXPYP .NE. 0) GO TO 8000   ,16   ,17   ,18
      CALL TRAV(RT(1,4),BEND,3)
C     RT = TRANSFORMATION ARRAY   IS COMPLETE.
C     BLENG = LENGTH OF BODY ALONG NEW X AXIS
C
C     TRANSFORM BODY MERIDIAN LINES TO NEW SYSTEM                       ,19
 500  CALL TRAPCT(B,NBPCT,RT)
C
C     NOW SCALE SO ALL MERIDIAN LINES WILL START AT
C       X=0, END AT X=BLENG.                                            ,20
      MU=4                                                              ,21
 600  CALL BSCALE(B,NBPCT,BLENG,NU)                                     ,22
      IF(NU)8000,1000,8000
C
C     SCALE BODY STA ARRAY TO CORRESPOND TO NEW COORD. SYSTEM           ,23
 1000 MU=5                                                              ,24
      D=STA(NSTA)-STA(1)                                                ,25
      IF(D)1200,8000,1200                                               ,26
 1200 ALFBET(1)=BLENG/D                                                 ,27
      ALFBET(2)=-ALFBET*STA(1)                                          ,28
      DO 1250 I=1,NSTA                                                  ,29
 1250 STA(I)=ALFBET*STA(I)+ALFBET(2)
C
C     FIND RADIUS AND Z-COORD. OF CENTROID AT EACH STATION.             ,30   ,31
 2000 MU=6                                                              ,32
      DO 2100 I=1,NSTA                                                  ,33
      CALL BODCR(B,NBPCT,STA(I),EP,CHD,2000,S,E,NE,
     , Y,RZ(I,2),A,RZ(I,1),NU)                                          ,34
      IF(NU)8000,2100,8000                                              ,35
 2100 CONTINUE
C                                                                       ,36   ,37
      NU=0                                                              ,38
      GO TO 9000
C
C     ERROR                                                             ,39
 8000 CONTINUE                                                          ,40
      LTAPE=NU(2)                                                       ,41
      IF(LTAPE .LE. 0)GO TO 8500                                        ,42   ,43   ,44
      WRITE(LTAPE,8100)MU,NU(1)                                         ,45   ,46   ,47
```

```
EXTERNAL FORMULA NUMBER   --   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

 8100 FORMAT(/13H0TFLAT1 ERROR I3,7H,  CODE I3,5X
     1 49HFAILURE IN TRANSFORMING BODY MERIDIAN LINES, ETC.)
      IF(MU .NE. 6)GO TO 8500                                          ,48   ,49   ,50
      WRITE(LTAPE,8200)STA(I)                                          ,51   ,52   ,53
 8200 FORMAT(8H STATION F10.4)
 8500 NU=MU
C
 9000 RETURN                                                           ,54
      END                                                              ,55
                                                                       ,56
```

```
C     SAME AS BODY1M EXCEPT DOES NOT IDENTIFY POINTS WHICH LIE
C     IN A BODY DEFINING SECTION.
C
C     USED BY TFLAT TO PRINT BODY MERIDIAN LINES.
C     ALSO COMPUTES RHO, THETA FOR EACH POINT.
C     THE RHO, THETA ORIGIN IS ON BODY AXIS.
C
      SUBROUTINE TFLATM(B,NB,AXIS,TITLE,DAYT,LO,STA)
C
      DIMENSION B(1),AXIS(2),TITLE(12),DAYT(2),STA(1),XYZ(3)
      LOGICAL NEWPCT,WTITLE
      DATA MAXLIN,DEGRAD/45,0206712273406/
C
C     B IS ARRAY OF NB MERIDIAN LINES, HEADED BY INDEX.
C     STA IS ARRAY OF BODY STATIONS (NOT USED BY TFLATM)
C     MAXLIN = MAX NO. OF LINES PER PAGE
C     MAXEL IS MAX NO. OF XYZ ELEMENTS THAT WILL GO ON A PAGE,
C       ASSUMING NO HEADINGS.  WE ALLOW FOR HEADINGS LATER.
C
      MAXEL=3*(MAXLIN-2)                                                ,1
      NELREM=0
C                                                                       ,2
      DO 4000 K=1,NB
C     WORKING ON KTH MERIDIAN LINE                                      ,3
      JK=3*K                                                            ,4
      NPTS=B(JK-1)                                                      ,5
      LOCST=B(JK)                                                       ,6
      LOCEND=LOCST+3*NPTS-1
C     THE X OF THE FIRST POINT IN THE MERIDIAN LINE IS
C       IN B(LOCST).
C     THE Z OF THE LAST POINT IS IN B(LOCEND).
C     FOR PAGE CONTROL WE WRITE A BLOCK OF POINTS FROM
C       B(LA) TO B(LB).                                                 ,7
      LA=LOCST                                                          ,8
      NEWPCT=.TRUE.                                                     ,9
      ISTA=1
C
C     WE KEEP COMING BACK TO HERE UNTIL ALL POINTS ARE WRITTEN          ,10
  200 NELREM=NELREM-9
C     THIS ALLOWS FOR HEADING                                           ,11
      IF(NELREM-6)300,400,400
C     NOT ENOUGH ROOM FOR 2 POINTS, SO WILL START NEW PAGE              ,12
  300 WTITLE=.TRUE.                                                     ,13
      NELREM=MAXEL-6                                                    ,14
      GO TO 500                                                         ,15
  400 WTITLE=.FALSE.
C                                                                       ,16
  500 LB=MINO (LOCEND,LA+NELREM-1)                                      ,17
      NL=(LB-LA+1)/3                                                    ,18
      IF(.NOT. WTITLE) GO TO 600                                        ,19   ,20   ,21
      WRITE(LO,550)TITLE,DAYT                                           ,22   ,23   ,24
  550 FORMAT(1H1 12A6/54X 2A6)
  600 IF(NEWPCT) GO TO 700
```

```
C     WRITE HEADING                                                    ,25   ,26   ,27
      WRITE(LO,650)K                                                   ,28   ,29   ,30
  650 FORMAT(19HOBODY MERIDIAN LINE I4,14H   (CONTINUED))
      GO TO 800                                                        ,31
  700 WRITE(LO,750)K                                                   ,32   ,33   ,34
  750 FORMAT(19HOBODY MERIDIAN LINE I4)
      NUM=0                                                            ,35
  800 WRITE(LO,850)                                                    ,36   ,37
  850 FORMAT(14X1HX10X1HY10X1HZ12X3HRHO7X5HTHETA)
C
C     READY TO WRITE POINTS, COMPUTING RHO-THETA AS WE GO
      DO 2000 L=1,NL                                                   ,38
      NUM=NUM+1                                                        ,39
      CALL TRAV(XYZ,B(LA),3)                                           ,40
      DY=XYZ(2)-AXIS(1)                                                ,41
      DZ=XYZ(3)-AXIS(2)                                                ,42
      RHO=SQRT(DY*DY+DZ*DZ)                                            ,43
      IF(RHO)1100,1100,1200                                            ,44
 1100 THETA=0.                                                         ,45
      GO TO 1300                                                       ,46
 1200 THETA = ATN1(ABS(DY),DZ)*DEGRAD
C                                                                      ,47
 1300 CONTINUE
C                                                                      ,48
 1500 WRITE(LO,1550)NUM,XYZ,RHO,THETA                                  ,49   ,50   ,51
 1550 FORMAT(I4,4X,3F11.4,3X,2F11.4)
      LA=LA+3                                                          ,52
 2000 CONTINUE
C                                                                      ,53   ,54
      NELREM=NELREM-3*NL                                               ,55
      IF(LB-LOCEND)3000,4000,4000                                      ,56
 3000 NEWPCT=.FALSE.                                                   ,57
      LA=LB+1                                                          ,58
      GO TO 200
C                                                                      ,59
 4000 CONTINUE
C                                                                      ,60   ,61
      RETURN                                                           ,62
      END                                                              ,63
```

```
      SUBROUTINE TFLATW(B,NB,TITLE,DAYT,LO)

C     USED BY TFLAT TO PRINT WING PERCENT LINES.
C       (ADAPTED FROM BODY1M)

      DIMENSION B(1),TITLE(12),DAYT(2),XYZ(3)
      LOGICAL NEWPCT,WTITLE
      DATA MAXLIN/45/
C
C     B IS ARRAY OF NB MERIDIAN LINES, HEADED BY INDEX.
C     MAXLIN = MAX NO. OF LINES PER PAGE
C     MAXEL IS MAX NO. OF XYZ ELEMENTS THAT WILL GO ON A PAGE,
C       ASSUMING NO HEADINGS.  WE ALLOW FOR HEADINGS LATER.
C
      MAXEL=3*(MAXLIN-2)                                                ,1
      NELREM=0
C                                                                       ,2
      DO 4000 K=1,NB
C     WORKING ON KTH MERIDIAN LINE                                      ,3
      JK=3*K                                                            ,4
      PCT=B(JK-2)                                                       ,5
      NPTS=B(JK-1)                                                      ,6
      LOCST=B(JK)                                                       ,7
      LOCEND=LOCST+3*NPTS-1
C     THE X OF THE FIRST POINT IN THE MERIDIAN LINE IS
C       IN B(LOCST).
C     THE Z OF THE LAST POINT IS IN B(LOCEND).
C     FOR PAGE CONTROL WE WRITE A BLOCK OF POINTS FROM
C       B(LA) TO B(LB).                                                 ,8
      LA=LOCST                                                          ,9
      NEWPCT=.TRUE.                                                     ,10
      ISTA=1
C
C     WE KEEP COMING BACK TO HERE UNTIL ALL POINTS ARE WRITTEN          ,11
  200 NELREM=NELREM-9
C     THIS ALLOWS FOR HEADING                                           ,12
      IF(NELREM-6)300,400,400
C     NOT ENOUGH ROOM FOR 2 POINTS, SO WILL START NEW PAGE              ,13
  300 WTITLE=.TRUE.                                                     ,14
      NELREM=MAXEL-6                                                    ,15
      GO TO 500                                                         ,16
  400 WTITLE=.FALSE.
C                                                                       ,17
  500 LB=MINO (LOCEND,LA+NELREM-1)                                      ,18
      NL=(LB-LA+1)/3                                                    ,19
      IF(.NOT. WTITLE) GO TO 600                                        ,20   ,21   ,22
      WRITE(LO,550)TITLE,DAYT                                           ,23   ,24   ,25
  550 FORMAT(1H1 12A6/54X 2A6)
  600 IF(NEWPCT) GO TO 700
C     WRITE HEADING                                                     ,26   ,27   ,28
      WRITE(LO,650) K,PCT                                               ,29   ,30   ,31
  650 FORMAT(24HOWING PERCENT CHORD LINE I3,1H,F9.4, 14H   (CONTINUED))
      GO TO 800                                                         ,32
```

```
  700 WRITE(LO,750) K,PCT                                          ,33   ,34   ,35
  750 FORMAT(24HOWING PERCENT CHORD LINE I3,1H,F9.4)
      NUM=0                                                        ,36
  800 WRITE(LO,850)                                                ,37   ,38
  850 FORMAT(14X 1HX 10X 1HY 10X 1HZ)
C
C     READY TO WRITE POINTS
      DO 2000 L=1,NL                                               ,39
      NUM=NUM+1                                                    ,40
      CALL TRAV(XYZ,B(LA),3)
C                                                                  ,41
      WRITE(LO,1550)NUM,XYZ                                        ,42   ,43   ,44
 1550 FORMAT(1H I3,4X,3F11.4)
      LA=LA+3
C                                                                  ,45
 2000 CONTINUE
C                                                                  ,46   ,47
      NELREM=NELREM-3*NL                                           ,48
      IF(LB-LOCEND)3000,4000,4000                                  ,49
 3000 NEWPCT=.FALSE.                                               ,50
      LA=LB+1                                                      ,51
      GO TO 200
C                                                                  ,52
 4000 CONTINUE
C                                                                  ,53   ,54
      RETURN                                                       ,55
      END                                                          ,56
```

```
      SUBROUTINE TFLATX(LO,DAT,NW,W,WB,NU)
C
C     USED BY TFLAT TO PRINT WING-BODY INTERSECTIONS.
C     ADAPTED FROM PART OF WBXUL
C     W = ARRAY OF WING PERCENT LINES (WITH HEADER)
C     WB = ARRAY OF WING-BODY INTERSECTION POINTS
C     NW = NO OF WING PERCENT LINES
C
      DIMENSION DAT(2),W(1),WB(1),PT(3),BAXIS(2)
C
      DATA BAXIS/0.,0./,DEGRAD/0206712273406/
C
C
      NU=0                                                              ,1
      WRITE(LO,100)DAT,BAXIS                                            ,2    ,3    ,4
  100 FORMAT(1H1 53X 2A6/23H INTERSECTIONS OF FLAT
     , 42HWING PERCENT CHORD LINES WITH BODY SURFACE /
     , 41H RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( F9.4,1H,F9.4,1H)/
     , 21HO(PRIMED-AXIS SYSTEM)/
     , 13HONO.  PERCENT 6X 1HX 9X 1HY 9X 1HZ 10X 3HRHO 6X 5HTHETA//)
C
      IPCT=1                                                            ,5
      IWBX=1
C                                                                       ,6
      DO 1000 J=1,NW                                                    ,7
      PCT=W(IPCT)                                                       ,8
      IPCT=IPCT+3                                                       ,9
      CALL TRAV(PT,WB(IWBX),3)                                          ,10
      IWBX=IWBX+3                                                       ,11
      IF(PT .GT. 1.E+25) GO TO 900                                      ,12   ,13   ,14
      DY=PT(2)                                                          ,15
      DZ=PT(3)                                                          ,16
      RHO=SQRT(DY*DY+DZ*DZ)                                             ,17
      IF(RHO)820,820,830                                                ,18
  820 THETA=0.                                                          ,19
      GO TO 850                                                         ,20
  830 THETA=ATN1(ABS(DY),DZ)*DEGRAD                                     ,21
  850 WRITE(LO,860)J,PCT,PT,RHO,THETA                                   ,22   ,23   ,24
  860 FORMAT(1H I2,F10.4,1X 3F10.4,2X 2F10.4)
      GO TO 1000                                                        ,25
  900 WRITE(LO,910)J,PCT                                                ,26   ,27   ,28
  910 FORMAT(1H I2,F10.4,6X 15HNO INTERSECTION)
      NU=1                                                              ,29
 1000 CONTINUE
C                                                                       ,30   ,31
      RETURN                                                            ,32
      END                                                               ,33
```

```
      SUBROUTINE THETAB
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NOMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/ NPLNB,NPLNW,JLEAD,JTRAIL,IMID,NPTS(16),X(16,90),Y(16
     1,90),Z(16,90),XCEPT(21),XCEPTB(21),XCEPTW(16),YCEPTW(16),ZCEPTW(16
     2),CODEBW(16),KPANEL(15,20),XCOR(16,21),YCOR(16,21),ZCOR(16,21),XIN
     3T(15,20,2),YINT(15,20,2),ZINT(15,20,2),XCEN(15,20),YCEN(15,20),ZCE
     4N(15,20),XCON(15,20),YCON(15,20),ZCON(15,20),AREA(15,20),ARAT(15,2
     50),THETA(15,20),ALPHA(15,20),CHORD(15,20)
      DIMENSION C(4),P(3,5),Q(3,5),LOC(3)

C     SUBROUTINE TO CALCULATE BODY PANEL
C     THETA ANGLES

C     DEFINE FUNCTIONS TO BE USED IN CALCULATIONS
C     DETR CALCULATES MATRIX DETERMINANT
C     ANGL CALCULATES THETA-INCLINATION ANGLE USING
C     ARCCOSINE FUNCTION ROUTINE
      DETR(A1,A2,A3,B1,B2,B3)=A1*(B2-B3)-A2*(B1-B3)+
     1A3*(B1-B2)
      ANGL(C1,C2)=ACOS(C1/SQRT(C1*C1+C2*C2))

C     DEFINE CONSTANTS
      E=0.001                                                          ,1
      F=0.000001

C     CALCULATE PANEL THETA ANGLES                                     ,2
 50   DO 70 I=1,NPER1                                                  ,3
      DO 65 J=1,NPLN1                                                  ,4
      IF (ABS(ZCOR(I+1,J+1)-ZCOR(I,J+1))-F) 55,55,60                   ,5
 55   THETA(I,J)=0.                                                    ,6
      GO TO 65                                                         ,7
 60   C1= DETR(XCOR(I,J),XCOR(I,J+1),XCOR(I+1,J+1),
     1YCOR(I,J),YCOR(I,J+1),YCOR(I+1,J+1))                             ,8
      C2=-DETR(XCOR(I,J),XCOR(I,J+1),XCOR(I+1,J+1),
     1ZCOR(I,J),ZCOR(I,J+1),ZCOR(I+1,J+1))                             ,9
      THETA(I,J)=(ZCOR(I+1,J+1)-ZCOR(I,J+1))/
     1ABS(ZCOR(I+1,J+1)-ZCOR(I,J+1))*ANGL(C1,C2)                       ,10
 65   CONTINUE                                                         ,11   ,12
 70   CONTINUE                                                         ,13   ,14
 100  RETURN                                                           ,15
      END                                                              ,16
```

```
      SUBROUTINE TRAPCT(B,N,RT)
      DIMENSION B(3,1),RT(3,4)
C
C     GIVEN AN ARRAY B OF N BODY MERIDIAN LINES OR WING PERCENT LINES,
C     (WITH HEADER WHICH DESCRIBES EACH PERCENT LINE), APPLY THE
C     TRANSFORMATION RT TO EACH POINT.
C
C     1ST POINT (X,Y,Z) STARTS IN B(1,N+1).
      J=N+1
C     NPT = TOTAL NO. OF POINTS IN B.                                  ,1
      NPT=0                                                            ,2
      DO 100 I=1,N                                                     ,3
      NP=B(2,I)                                                        ,4
  100 NPT=NPT+NP
C     TRANSFORM TO NEW COORD. SYSTEM                                   ,5    ,6
      DO 200 I=1,NPT                                                   ,7
      CALL TROTPT(RT,RT(1,4),2,B(1,J),B(1,J))                          ,8
  200 J=J+1
C                                                                      ,9    ,10
      RETURN                                                           ,11
      END                                                              ,12
```

```
      SUBROUTINE TRAV(A,B,N)
      DIMENSION A(1),B(1)
C
      DO 100 I=1,N                                                  ,1
 100  A(I)=B(I)
C                                                                   ,2     ,3
      RETURN                                                        ,4
      END                                                           ,5
```

```
EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE TROTPT(R,T,N,P,Q)
C
      DIMENSION R(3,3),T(3),P(3),Q(3),A(3)
C
C     TRANSLATION AND ROTATION OF A POINT
C     P IS A POINT (X,Y,Z) IN FIRST SYSTEM, Q IN 2ND
C     N = 1 IF Q GIVEN, P WANTED.  N = 2 IF P GIVEN, Q WANTED
C     R AND T ARE ROTATION, TRANSL. MATRICES TO MAP Q INTO P
C     P = R Q +T
C
      IF (N-1)20, 1,20                                                  ,1
    1 DO 5 I=1,3                                                        ,2
    5 A(I)=Q(I)                                                         ,3     ,4
   10 DO 15 I=1,3                                                       ,5
      U=0.                                                              ,6
      DO 12 J=1,3                                                       ,7
   12 U=U+R(I,J)*A(J)                                                   ,8     ,9
   15 P(I)=U+T(I)                                                       ,10    ,11
      GO TO 50                                                          ,12
   20 DO 25 I=1,3                                                       ,13
   25 A(I)=P(I)-T(I)                                                    ,14    ,15
      DO 40 I=1,3                                                       ,16
      U=0.                                                              ,17
      DO 30 J=1,3                                                       ,18
   30 U=U+R(J,I)*A(J)                                                   ,19    ,20
   40 Q(I)=U                                                            ,21    ,22
   50 RETURN                                                            ,23
      END                                                               ,24
```

```
      SUBROUTINE TVEL

C     COMPUTE VELOCITY COMPONENTS INDUCED BY WING SOURCES

C     COMPUTES VELOCITY COMPONENTS DUE TO WING SOURCES

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE

      COMMON /BLOCK/X(200,3,4),Y(200,3,4),Z(200,3,4),XBAR(200)
     1,YBAR(200),ZBAR(200),XC(200),YC(200),ZC(200),THETA(200),ALPHAS(200
     2),AREA(200),CHORD(200),NPART(200),NPANEL,ISYM,A(200),U(200),V(200)
     3,VPM(200),VV(200),W(200),WPM(200),WW(200),VPMM(200),WPMM(200),ALPH
     1AC(100),ALPHAT(100)

      DIMENSION B(200),C(200),D(200),AN(200)

      EQUIVALENCE (VV,B),(WPM,C),(WW,D),(VPMM,AN)

      NS=NBODY+1                                                        ,1
      NPANEL=NBODY+NWING
                                                                        ,2
      DO 100 J=1,NPANEL                                                 ,3
      U(J)=0.                                                           ,4
      V(J)=0.                                                           ,5
      W(J)=0.                                                           ,6
      AN(J)=0.                                                          ,7
  100 CONTINUE
                                                                        ,8    ,9
      DO 200 J=1,NWING                                                  ,10
      READ (NTAPEB) (A(I),B(I),C(I),I=1,NPANEL)                         ,11   ,12   ,13   ,14   ,15
      READ (NTAPEA) (D(I),I=1,NPANEL)                                   ,16   ,17   ,18   ,19   ,20
      DO 200 I=1,NPANEL                                                 ,21
      U(I)=U(I)+A(I)*ALPHAT(J)                                          ,22
      V(I)=V(I)+B(I)*ALPHAT(J)                                          ,23
      W(I)=W(I)+C(I)*ALPHAT(J)                                          ,24
      AN(I)=AN(I)+D(I)*ALPHAT(J)                                        ,25
  200 CONTINUE                                                          ,26   ,27   ,28
      REWIND NTAPEA                                                     ,29
      REWIND NTAPEB
                                                                        ,30
      IF (NBODY) 300,400,300

C     WRITE VELOCITY COMPONENTS ON BODY PANELS INDUCED BY WING SOURCES  ,31
  300 CONTINUE                                                          ,32
      WRITE (NTAPEC) (U(I),V(I),W(I),I=1,NBODY)                         ,33   ,34   ,35   ,36   ,37
      END FILE NTAPEC

C     WRITE VELOCITY COMPONENTS ON WING PANELS INDUCED BY WING SOURCES  ,38
  400 CONTINUE                                                          ,39
      WRITE (NTAPEC) (U(I),V(I),W(I),I=NS,NPANEL)                       ,40   ,41   ,42   ,43   ,44
      END FILE NTAPEC
                                                                        ,45
```

```
      IF (NBODY) 450,500,450

C     WRITE NORMAL COMPONENT OF VELOCITY OF VELOCITY ON BODY PANELS
C     INDUCED BY WING SOURCES                                          ,46
 450  CONTINUE                                                         ,47
      WRITE (NTAPEC) (AN(I),I=1,NBODY)                                 ,48   ,49   ,50   ,51   ,52
      END FILE NTAPEC                                                  ,53
 500  CONTINUE                                                         ,54
      REWIND NTAPEC
                                                                       ,55
      RETURN                                                           ,56
      END                                                              ,57
```

```
UFOOL                                                                    07/31/65
7094 RELMOD ASSEMBLY.

        $IBLDR UFOOL                                                                    UF000000
ASSEMBLED TEXT.

        $TEXT  UFOOL                                                                    UF000001

                                  *FOOL       MICHAEL SYNGE          6-9247       ASSEMBLED  26 JAN 65
                                         ENTRY   FOOL
                                   IFOOL ENTRY   FOOL
                                  *

BINARY CARD (NOT PUNCHED)
      00000  0020 00 4 00001  10000   FOOL  TRA     1,4
                                  *
                      00000  01111         END
CONTROL DICTIONARY

        $CDICT UFOOL                                                                    UF000002

BINARY CARD (NOT PUNCHED)
          000001000000              PREFACE     START=0,LENGTH=1,TYPE=7094,CMPLX=5
          000004000005
          642646464360       UFOOL  DECK        LOC=0,LENGTH=1
          000001000000
          264646436060       FOOL   REAL        LOC=0,LENGTH=0
          000000000000
          312646464360       IFOOL  REAL        LOC=0,LENGTH=0
          000000000000

        $DKEND UFOOL                                                                    UF000003

NO MESSAGES FOR THIS ASSEMBLY
        SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

  CLASS  SYMBOL   VALUE   REFERENCES

         FOOL     00000
  LCTR    BLCTR
  QUAL    UNDS
  LCTR   //
```

```
$IBLDR UN07                                                          UN070000
$FILE  UN07    (UNIT071,C(1),READY,NOLIST,INOUT,BLK=256,HIGH,BIN     UN070001
```

```
$IBLDR UN08                                                          UN080000
$FILE  UN08    (UNIT08),C(2),READY,NOLIST,INOUT,BLK=256,HIGH,BIN     UN080001
```

```
$IBLDR UN09                                                          UN090000
$FILE  UN09    (UNIT09),A(1),READY,NOLIST,INOUT,BLK=14,HIGH,BCD      UN090001
```

```
UPDATE
7094 RELMOD ASSEMBLY.                                                    07/31/65

        $IBLDR UPDATE                                                                    UPDA0000
ASSEMBLED TEXT.

        $TEXT  UPDATE                                                                    UPDA0001

                                  *UPDATE     MICHAEL SYNGE          6-9247      ASSEMBLED  21 SEP 64
                                  *     TO UPDATE THE CONTENTS OF AN ARRAY NAME WHEN THE
                                  *     ARRAY IS MOVED (E.G. BY PACKING).
                                  *
                                  *              CALLING SEQUENCE
                                  *
                                  *                  CALL UPDATE(BUF(M-1))
                                  *
                                  *              WHERE M IS THE FIRST WORD OF THE ARRAY
                                  *              (AFTER MOVING) IN THE BUFFER  BUF .
                                  *
                                         ENTRY   UPDATE
                                  *

BINARY CARD (NOT PUNCHED)
      00000  0500 60 4 00003  10000  UPDATE CLA*    3,4
      00001  0621 00 0 00007  10001         STA     NAME
      00002  0500 00 4 00003  10000         CLA     3,4
      00003  0402 00 0 00011  10001         SUB     =1
      00004  0621 00 0 01001  10011         STA     *+1
      00005  0500 00 0 00000  10000         CLA     **
      00006  4320 00 0 00012  10001         ANA     =077777
      00007  0621 00 0 00000  10000   NAME  STA     **
      00010  0020 00 4 00001  10000         TRA     1,4
                                  *
      00011  000000000001     10000         *LORG
      00012  000000077777     10000
                      00000   01111         END
          CONTROL DICTIONARY

        $CDICT UPDATE                                                                    UPDA0002

BINARY CARD (NOT PUNCHED)
            000013000000              PREFACE     START=0,LENGTH=11,TYPE=7094,CMPLX=5
            000004000005
            644724216325       UPDATE DECK        LOC=0,LENGTH=11
            000013000000
            644724216325       UPDATE REAL        LOC=0,LENGTH=0
            000000000000

        $DKEND UPDATE                                                                    UPDA0003

NO MESSAGES FOR THIS ASSEMBLY
```

REFERENCES TO DEFINED SYMBOLS.

CLASS	SYMBOL	VALUE	REFERENCES
	NAME	00007	1
LCTR	BLCTR		
QUAL	UNQS		
LCTR	//		
	UPDATE	00000	

```
      FUNCTION UVECN(A,B,V,K,N)
      DIMENSION A(1),B(1),V(1)
C     GIVEN VECTORS A,B
C     FIND DISTANCE (= UVECN) FROM A TO B.
C     IF DISTANCE NOT ZERO AND K=0, FIND V, A UNIT
C       VECTOR FROM A TOWARD B.
C     OTHERWISE, V=B-A.
C
      D=0.                                                          ,1
      DO 100 I=1,N                                                  ,2
      V(I)=B(I)-A(I)                                                ,3
 100  D=D+V(I)**2
C
      IF (D)500,500,200                                             ,4     ,5
 200  D=SQRT(D)                                                     ,6
      IF (K)500,300,500                                             ,7
 300  DO 400 I=1,N                                                  ,8
 400  V(I)=V(I)/D                                                   ,9
 500  UVECN=D                                                       ,10    ,11
      RETURN                                                        ,12
      END                                                           ,13
                                                                    ,14
```

```
      SUBROUTINE VCROS(A,B,C,D,K)
      DIMENSION A(3),B(3),C(3),X(3),Y(3),Z(3)
C
C     FINDS THE CROSS PRODUCT, VECTOR C, OF VECTORS A AND B.
C     C = A X B.  IF K NOT 0 THEN C IS NORMALIZED. =0, NOT.
C     D = MAGNITUDE OF C BEFORE NORMALIZED.
C     ON OUTPUT, K=0 IF SUCCESS, =1 IF NORM. REQUESTED BUT D=0.
C     C MAY BE STORED IN A OR B IF DESIRED.
C
      DO 10 I=1,3                                                       ,1
      X(I)=A(I)                                                         ,2
   10 Y(I)=B(I)
                                                                        ,3     ,4
      Z(1)=X(2)*Y(3)-X(3)*Y(2)                                          ,5
      Z(2)=X(3)*Y(1)-X(1)*Y(3)                                          ,6
      Z(3)=X(1)*Y(2)-X(2)*Y(1)                                          ,7
      E=SQRT (VDOTP(Z,Z))                                               ,8
      D=E                                                               ,9
      IF (K)20,50,20                                                    ,10
   20 IF (E)40,30,40                                                    ,11
   30 K=1                                                               ,12
      GO TO 50                                                          ,13
   40 K=0                                                               ,14
      GO TO 100                                                         ,15
   50 E=1.                                                              ,16
  100 DO 110 I=1,3                                                      ,17
  110 C(I)=Z(I)/E                                                       ,18    ,19
      RETURN                                                            ,20
      END                                                               ,21
```

```
EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      FUNCTION VDOTN(A,B,N)
      DIMENSION A(N),B(N)
C     DOT PRODUCT OF TWO VECTORS IN N-SPACE
      VDOTN=0.                                                          ,1
      DO 100 I=1,N                                                      ,2
 100  VDOTN=VDOTN+A(I)*B(I)                                             ,3     ,4
      RETURN                                                            ,5
      END                                                               ,6
```

```
      FUNCTION VDOTP(P,Q)
C
      DIMENSION P(3),Q(3)
C     FINDS THE DOT PRODUCT OF VECTORS P AND Q
      VDOTP=P*Q+P(2)*Q(2)+P(3)*Q(3)                                    ,1
      RETURN                                                           ,2
      END                                                              ,3
```

```
      SUBROUTINE WBX(B,NB,W,NW,EPS,KODE,S,BW,NU)
C
      DIMENSION B(1),W(1),S(2,1),BW(3,1),NU(3)
     1 Q(3,2),R(10),DUM(3)
     2,WGPTS(3,2),UVEC(3),WP(3,2)
     3,PTCNWG(3)
      LOGICAL INTD
C
C     USED BY WBXUL TO FIND THE INTERSECTIONS OF THE UPPER OR LOWER
C       WING PERCENT LINES WITH THE BODY.
C     B IS THE ARRAY OF NB BODY PERCENT LINES.
C     W IS THE ARRAY OF NW WING PERCENT LINES.
C
C     A PERCENT LINE OF N POINTS HAS 3*N ELEMENTS
C             X1,Y1,Z1, X2,Y2,Z2, ........., XN,YN,ZN
C     THE FIRST 3*J CELLS OF AN ARRAY B OF J PERCENT LINES FORMS A TABLE OF
C     CONTENTS TO THE REST OF THE ARRAY.  ELEMENTS 1,4,7,...   OF B ARE LABELS
C     FOR PERCENT LINES 1,2,3,...   ELEMENTS 2,5,8,... ARE THE NUMBER OF
C     3-DIMENSIONAL POINTS IN PERCENT LINES 1,2,3,...   ELEMENTS 3,6,9,...   ARE
C     THE STARTING LOCATIONS IN B OF PERCENT LINES 1,2,3,...
C     FOR EXAMPLE, IF B(7) = 30., B(8) = 12. AND B(9) = 49., THEN THE PERCENT
C     LINE WITH THE LABEL 30. CONTAINS 12 POINTS AND STARTS IN B(49).
C
C     EPS IS THE TOLERANCE IN LOCATING INTERSECTION POINTS.
C       A WARNING MESSAGE IS WRITTEN IF THE ERROR IS GREATER THAN EPS.
C     KODE =1 IF THE BODY IS POLYGONAL IN CROSS SECTON, = 2 IF SMOOTH.
C     S IS A SCRATCH ARRAY AT LEAST 2*NB CELLS LONG.  USED FOR 2-DIMENSIONAL
C       CROSS SECTIONS THRU THE BODY.
C     BW IS AN OUTPUT ARRAY OF BODY-WING INTERSECTION POINTS.  IF THE JTH
C       WING PERCENT LINE DOES NOT INTERSECT THE BODY, THEN
C       BW(1,J) = BW(2,J) = BW(3,J) = 1.E+30.
C
C     NU IS AN ERROR ARRAY WHICH IS PASSED DOWN TO LOWER LEVEL SUBROUTINES.
C     NU(1), AN OUTPUT, IS ZERO FOR SUCCESS, NON-ZERO FOR FAILURE.
C     NU(2), AN INPUT, MAY BE INTERPRETED AS A  TAPE NUMBER ON WHICH TO
C       WRITE AN ERROR MESSAGE, IF GREATER THAN ZERO.
C     NU(3), BOTH AN INPUT AND AN OUTPUT, IS AN ERROR COUNTER AND ERROR MESSAGE
C       LIMITER.  SOME LOWER LEVEL SUBROUTINES MAY SUBTRACT 1 FROM NU(3) IF AN
C       ERROR OCCURS AND THEN WRITE AN ERROR MESSAGE IF NU(3) AND NU(2) ARE BOTH
C       GREATER THAN ZERO.
C
C     INPUTS ARE B,NB,W,NW,EPS,KODE,NU(2),NU(3). OUTPUTS BW,S,BW,NU(1),NU(3)
C
      DATA SR,DUM/3HWBX,3*1.E+3C/
C
      R(10)=1.E-7
C
C     CUT THE BODY WITH A STATION PLANE THRU EACH WING PERCENT POINT IN TURN
C     UNTIL THE INTERSECTION POINT IS ISOLATED BETWEEN 2 ADJACENT WING POINTS.
C     THEN ITERATE UNTIL THE X-VALUE OF THE INTERSECTION IS FOUND (WITHIN EPS).
C                                                                          ,1
 100  DO 6000 IW=1,NW
C     FOR EACH WING PERCENT LINE IN W.
```

```
C                                                                       ,2
      INTD=.FALSE.                                                      ,3
      L=IW*3
C     NWP IS NO. OF PTS. IN THE IW PERCENT LINE AND LWP IS THE LOCATION OF THE
C     FIRST POINT IN W.  LWP IS UPDATED IN THE LOOP BELOW.              ,4
      NWP=W(L-1)                                                        ,5
      LWP=W(L)
C                                                                       ,6
 200  DO 1200 K=1,NWP
C     FOR EACH POINT W(LWP) IN THE PERCENT LINE.
C
C     FIND THE HORIZONTAL DISTANCE D FROM POINT W(LWP) TO THE BODY AND THE
C     CORRESPONDING  POINT Q ON THE BODY.                               ,7
      EPC=AMAX1(ABS(W(LWP)),1.)*1.E-6                                   ,8
      CALL WBXC(B,NB,EPC,W(LWP),KODE,S,D,Q,NU)                          ,9
      IF(NU)250,500,300                                                 ,10
 250  MU=6
C     NU = -5 MEANS BODY IS NOT DEFINED AT STATION X                    ,11
      IF(NU+5)9000,400,9000                                             ,12
 300  MU=7
C     NU = 1 OR 2 MEANS WING POINT IS ABOVE OR BELOW THE BODY           ,13
      IF(NU-2)310,310,9000                                              ,14
 310  MU=8                                                              ,15
      IF(D)510,9000,510
C     SEE IF PREVIOUS POINT WAS OK.                                     ,16
  400 IF(INTD) GO TO 1250                                               ,17   ,18   ,19
      GO TO 1000                                                        ,20
 500  IF(D .EQ. 0.) GO TO 1500                                          ,21   ,22   ,23
 510  IF(INTD) GO TO 600                                                ,24   ,25   ,26
      INTD=.TRUE.                                                       ,27
      GO TO 700
C
C     SEE IF THIS POINT AND PREVIOUS POINT ARE ON OPPOSITE SIDES OF BODY SURFACE   ,28
  600 IF(D*DP)650,700,700
C     OPPOSITE SIDES
C     THE INTERSECTION PT. IS BETWEEN W(LWPP) AND W(LWP)                ,29
 650  R(4)=D                                                            ,30
      GO TO 1600
C
C     SAME SIDE OR ELSE FIRST TIME WBXC SUCCEEDED.                      ,31
 700  DP=D
C                                                                       ,32
 1000 LWPP=LWP                                                          ,33
      LWP=LWP+3
C                                                                       ,34
 1200 CONTINUE
C                                                                       ,35   ,36
      GO TO 1300
C
C     THE X-VALUE (=W(LWP)) OF THIS WING PERCENT POINT IS OUTSIDE
C     THE RANGE OF BODY X-VALUES.  HOWEVER, THE PREVIOUS WING
C     PERCENT POINT WAS OK.  ONLY 1 OF THE BODY END POINTS CAN
C     LIE BETWEEN W(LWPP) AND W(LWP).  FIND THIS POINT, INTERSECT
C     THE WING PERCENT LINE SEGMENT WITH THE CORRESPONDING
```

```
      SUBROUTINE WBX(B,NB,W,NW,EPS,KODE,S,BW,NU)
C
      DIMENSION B(1),W(1),S(2,1),BW(3,1),NU(3)
     1 Q(3,2),R(10),DUM(3)
     2,WGPTS(3,2),UVEC(3),WP(3,2)
     3,PTCNWG(3)
      LOGICAL INTD
C
C     USED BY WBXUL TO FIND THE INTERSECTIONS OF THE UPPER OR LOWER
C       WING PERCENT LINES WITH THE BODY.
C     B IS THE ARRAY OF NB BODY PERCENT LINES.
C     W IS THE ARRAY OF NW WING PERCENT LINES.
C
C     A PERCENT LINE OF N POINTS HAS 3*N ELEMENTS
C             X1,Y1,Z1, X2,Y2,Z2, ........., XN,YN,ZN
C     THE FIRST 3*J CELLS OF AN ARRAY B OF J PERCENT LINES FORMS A TABLE OF
C     CONTENTS TO THE REST OF THE ARRAY.  ELEMENTS 1,4,7,...   OF B ARE LABELS
C     FOR PERCENT LINES 1,2,3,...   ELEMENTS 2,5,8,... ARE THE NUMBER OF
C     3-DIMENSIONAL POINTS IN PERCENT LINES 1,2,3,...   ELEMENTS 3,6,9,...   ARE
C     THE STARTING LOCATIONS IN B OF PERCENT LINES 1,2,3,...
C     FOR EXAMPLE, IF B(7) = 30., B(8) = 12. AND B(9) = 49., THEN THE PERCENT
C     LINE WITH THE LABEL 30. CONTAINS 12 POINTS AND STARTS IN B(49).
C
C     EPS IS THE TOLERANCE IN LOCATING INTERSECTION POINTS.
C       A WARNING MESSAGE IS WRITTEN IF THE ERROR IS GREATER THAN EPS.
C     KODE =1 IF THE BODY IS POLYGONAL IN CROSS SECTON, = 2 IF SMOOTH.
C     S IS A SCRATCH ARRAY AT LEAST 2*NB CELLS LONG.  USED FOR 2-DIMENSIONAL
C       CROSS SECTIONS THRU THE BODY.
C     BW IS AN OUTPUT ARRAY OF BODY-WING INTERSECTION POINTS.  IF THE JTH
C       WING PERCENT LINE DOES NOT INTERSECT THE BODY, THEN
C       BW(1,J) = BW(2,J) = BW(3,J) = 1.E+30.
C
C     NU IS AN ERROR ARRAY WHICH IS PASSED DOWN TO LOWER LEVEL SUBROUTINES.
C     NU(1), AN OUTPUT, IS ZERO FOR SUCCESS, NON-ZERO FOR FAILURE.
C     NU(2), AN INPUT, MAY BE INTERPRETED AS A  TAPE NUMBER ON WHICH TO
C       WRITE AN ERROR MESSAGE, IF GREATER THAN ZERO.
C     NU(3), BOTH AN INPUT AND AN OUTPUT, IS AN ERROR COUNTER AND ERROR MESSAGE
C       LIMITER.  SOME LOWER LEVEL SUBROUTINES MAY SUBTRACT 1 FROM NU(3) IF AN
C       ERROR OCCURS AND THEN WRITE AN ERROR MESSAGE IF NU(3) AND NU(2) ARE BOTH
C       GREATER THAN ZERO.
C
C     INPUTS ARE B,NB,W,NW,EPS,KODE,NU(2),NU(3). OUTPUTS BW,S,BW,NU(1),NU(3)
C
      DATA SR,DUM/3HWBX,3*1.E+30/
C
      R(10)=1.E-7
C
C     CUT THE BODY WITH A STATION PLANE THRU EACH WING PERCENT POINT IN TURN
C     UNTIL THE INTERSECTION POINT IS ISOLATED BETWEEN 2 ADJACENT WING POINTS.
C     THEN ITERATE UNTIL THE X-VALUE OF THE INTERSECTION IS FOUND (WITHIN EPS).
C                                                                            ,1
 100  DO 6000 IW=1,NW
C     FOR EACH WING PERCENT LINE IN W.
```

```
C                                                                       ,2
      INTD=.FALSE.                                                      ,3
      L=IW*3
C     NWP IS NO. OF PTS. IN THE IW PERCENT LINE AND LWP IS THE LOCATION OF THE
C     FIRST POINT IN W.  LWP IS UPDATED IN THE LOOP BELOW.              ,4
      NWP=W(L-1)                                                        ,5
      LWP=W(L)
C                                                                       ,6
 200  DO 1200 K=1,NWP
C     FOR EACH POINT W(LWP) IN THE PERCENT LINE.
C
C     FIND THE HORIZONTAL DISTANCE D FROM POINT W(LWP) TO THE BODY AND THE
C     CORRESPONDING  POINT Q ON THE BODY.                               ,7
      EPC=AMAX1(ABS(W(LWP)),1.)*1.E-6                                   ,8
      CALL WBXC(B,NB,EPC,W(LWP),KODE,S,D,Q,NU)                          ,9
      IF(NU)250,500,300                                                 ,10
 250  MU=6
C     NU = -5 MEANS BODY IS NOT DEFINED AT STATION X                    ,11
      IF(NU+5)9000,400,9000                                             ,12
 300  MU=7
C     NU = 1 OR 2 MEANS WING POINT IS ABOVE OR BELOW THE BODY           ,13
      IF(NU-2)310,310,9000                                              ,14
 310  MU=8                                                              ,15
      IF(D)510,9000,510
C     SEE IF PREVIOUS POINT WAS OK.                                     ,16
  400 IF(INTD) GO TO 1250                                               ,17   ,18   ,19
      GO TO 1000                                                        ,20
 500  IF(D .EQ. 0.) GO TO 1500                                          ,21   ,22   ,23
 510  IF(INTD) GO TO 600                                                ,24   ,25   ,26
      INTD=.TRUE.                                                       ,27
      GO TO 700
C
C     SEE IF THIS POINT AND PREVIOUS POINT ARE ON OPPOSITE SIDES OF BODY SURFACE   ,28
  600 IF(D*DP)650,700,700
C     OPPOSITE SIDES
C     THE INTERSECTION PT. IS BETWEEN W(LWPP) AND W(LWP)                ,29
 650  R(4)=D                                                            ,30
      GO TO 1600
C
C     SAME SIDE OR ELSE FIRST TIME WBXC SUCCEEDED.                      ,31
 700  DP=D
C                                                                       ,32
 1000 LWPP=LWP                                                          ,33
      LWP=LWP+3
C                                                                       ,34
 1200 CONTINUE
C                                                                       ,35   ,36
      GO TO 1300
C
C     THE X-VALUE (=W(LWP)) OF THIS WING PERCENT POINT IS OUTSIDE
C     THE RANGE OF BODY X-VALUES.  HOWEVER, THE PREVIOUS WING
C     PERCENT POINT WAS OK.  ONLY 1 OF THE BODY END POINTS CAN
C     LIE BETWEEN W(LWPP) AND W(LWP).  FIND THIS POINT, INTERSECT
C     THE WING PERCENT LINE SEGMENT WITH THE CORRESPONDING
```

```
C     BODY STATION PLANE TO GET A POINT ON THE WING P.C. LINE.
C     CALL THIS POINT  PTONWG.
C     THE DISTANCE FROM THIS PT. TO THE BODY SURFACE WILL BE IN
C     R(2)=R(4).  THE X-VALUE WILL BE IN R(1)=R(3).
C                                                                          ,37
 1250 CALL WBXD(R,B,NB,KODE,PTONWG,Q,S,W(LWPP),NU)                         ,38
      MU=5                                                                ,39
      IF(NU)9000,1260,9000                                                ,40
 1260 D=R(2)                                                              ,41
      IF(D*DP)1600,1600,1300
C
C     NO INTERSECTION.  LEAVE PERCENT LINE UNCHANGED.  STORE X,Y,Z = 1.E+30
C     AS INTERSECTION POINT.                                               ,42
 1300 KOUT=0                                                              ,43
      CALL TRAV(BW(1,IW),DUM,3)                                           ,44
      GO TO 5000
C
C     WING POINT W(LWP) LIES ON THE BODY SURFACE.                          ,45
 1500 KOUT=K                                                              ,46
      GO TO 4500
C                                                                         ,47
 1600 CONTINUE                                                            ,48
      CALL TRAV(WGPTS,W(LWPP),6)                                          ,49
      IF(MU .NE. 5) GO TO 1650                                            ,50   ,51   ,52
      CALL TRAV(WGPTS(1,2),PTONWG,3)
C     WGPTS CONTAINS THE COORD. OF 2 CONSECUTIVE POINTS ON THE
C     WING PERCENT LINE WHICH LIE ON OPPOSITE SIDES OF THE BODY SURFACE.
C     FIND THE INTERSECTION OF THE LINE SEGMENT WGPTS WITH THE BODY
C     SURFACE AS FOLLOWS,  CHOOSE A POINT (IN WP) WHICH IS SOME
C     DISTANCE ALONG WGPTS FROM ITS 1ST END POINT.  (A UNIT VECTOR
C     ALONG WGPTS WILL BE UVEC).  USE WBXC TO FIND THE HORIZONTAL
C     DISTANCE (IN Y-DIRECTION) FROM WP TO THE BODY SURFACE.
C     ITERATE ON DISTANCE ALONG WGPTS UNTIL WP LIES ON BODY SURFACE.
C
C     USE FROOTA TO TAKE CARE OF THE BOOK-KEEPING FOR THE ITERATION
C     AND TO PREDICT DISTANCES ALONG WGPTS TO USE FOR NEXT ITERATION.
C     FROOTA WILL SET JR = 0 WHEN INTERSECTION IS FOUND.
C                                                                          ,53
 1650 CONTINUE                                                            ,54
      R(1)=0.                                                             ,55
      R(2)=DP                                                             ,56
      R(3)=UVECN(WGPTS,WGPTS(1,2),UVEC,0,3)
C     R(3) = LENGTH OF THE LINE SEGMENT WGPTS
C     R(4) WAS SET TO D PREVIOUSLY                                         ,57
      R(9)=0.                                                             ,58
      MU=9                                                                ,59
      IF(R(3))9000,9000,1700                                              ,60
 1700 MU=10
C                                                                         ,61
      DO 3000 KR=1,100                                                    ,62
      CALL FROOTA(R,JR)
C     R(1) AND R(3) ARE NEW TRIAL VALUES OF DISTANCE ALONG WGPTS
                                                                          ,63
      DO 1900 I=1,2                                                       ,64
```

```
      II=I+I                                                            ,65
      DO 1800 J=1,3                                                     ,66
 1800 WP(J,I)=WGPTS(J,1)+R(II-1)*UVEC(J)                                ,67   ,68
      CALL WBXC(B,NB,EPC,WP(1,I),KODE,S,R(II),Q(1,I),NU)
C     R(2) AND R(4) ARE CORRESPONDING Y-DISTANCES OF WP FROM BODY SURFACE  ,69
      IF(NU .LT. 0  .OR.  NU .GT. 2) GO TO 9000                         ,70   ,71   ,72
 1900 CONTINUE
C                                                                       ,73   ,74
      IF(JR)2000,3100,3000                                              ,75
 2000 MU=11                                                             ,76
      GO TO 9000
C                                                                       ,77
 3000 CONTINUE
C
C     DID NOT CONVERGE                                                  ,78   ,79
      MU=12                                                             ,80
      GO TO 9000
C                                                                       ,81
 3100 IF(ABS(R(2))-EPS) 4000,4000,3110                                  ,82
 3110 LTAPE=NU(2)                                                       ,83
      NU(3)=NU(3)-1                                                     ,84
      IF(LTAPE .LE. 0  .OR.  NU(3) .LE. 0)GO TO 4000                    ,85   ,86   ,87
      WRITE(LTAPE,3120)IW,EPS,(WP(I,1),I=1,3)                           ,88   ,89   ,90   ,91   ,92   ,93

 3120 FORMAT(/42H0*** INTERSECTION OF WING PERCENT LINE NO.
     1 I3,40H WITH BODY NOT WITHIN GIVEN TOLERANCE OF
     2 F10.6/27H POINT ON WING HAS COORD. ( 3F10.4,1H) )
C
C     THE INTERSECTION (IN Q) HAS BEEN FOUND AND LIES BETWEEN
C     POINTS K-1 AND K.
 4000 KOUT=K-1
C     THE FIRST KOUT POINTS ARE INSIDE THE BODY AND ARE TO
C     BE REMOVED.
C     STORE THE INTERSECTION POINT.                                     ,94
 4500 CALL TRAV(BW(1,IW),Q,3)
C
C     ADJUST THE W ARRAY TO REFLECT ONLY THAT PART OF WING
C     OUTSIDE BODY.                                                     ,95
 5000 CALL WBXA(W,IW,KOUT,BW(1,IW),NU)                                  ,96
      MU=4                                                              ,97
      IF (NU)9000,6000,9000
C                                                                       ,98
 6000 CONTINUE                                                          ,99   ,100
      MU=0                                                              ,101
      GO TO 9900
C
C     WRITE ERROR MESSAGE.                                              ,102
 9000 LT=NU(2)                                                          ,103
      NU(3)=NU(3)-1                                                     ,104
      IF(LT .LE. 0 .OR. NU(3) .LE.0) GO TO 9900                         ,105  ,106  ,107
      WRITE(LT,9100)MU,NU(1),SR,IW,K                                    ,108  ,109  ,110
 9100 FORMAT(6H0ERROR I6,6H, CODE I6,15H IN SUBROUTINE A6/
     , 18H WING PERCENT LINE I5,7H, POINT I5)
 9900 NU=MU                                                             ,111
```

```
EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

RETURN                                                                  ,112
END                                                                     ,113
```

```
      SUBROUTINE WBXA(B,I,K,P,NU)
C
      DIMENSION B(1),P(3)
C
C     USED BY WBX TO ADJUST THE B ARRAY OF WING PERCENT LINES
C     TO REMOVE THAT PART OF THE I-TH PERCENT LINE INSIDE BODY.
C     INPUTS ARE B,I,K,P.  OUTPUTS ARE B,NU.
C
      L=3*I
C     KB IS NO. OF POINTS IN I-TH PERCENT LINE.                          ,1
      KB=B(L-1)
C     KC IS STARTING LOCATION OF I-TH PERCENT LINE.                      ,2
      KC=B(L)                                                            ,3
      KO=KC+3*K-1                                                        ,4
      IF (K .GE. KB .OR. K .LT. 0) GO TO 5000                            ,5    ,6    ,7
      IF (I .EQ. 1) GO TO 1000
C     JB IS NO. OF PTS. IN (I-1)TH PERCENT LINE                          ,8    ,9    ,10
      JB=B(L-4)
C     JC IS STARTING LOCATION.                                           ,11
      JC=B(L-3)
C     KN = STARTING LOCATION OF NEW PERCENT LINE                         ,12
      KN=JC+3*JB                                                         ,13
      B(L)=KN                                                            ,14
      GO TO 2000
C                                                                        ,15
 1000 KN=KC                                                              ,16
 2000 NN=KB-K                                                            ,17
      IF (K)2100,2100,2200                                               ,18
 2100 NN=NN-1
C     NN+1 = NO. OF PTS. IN NEW PERCENT LINE.                            ,19
 2200 B(L-1)=NN+1                                                        ,20
      KN=KN-1
C                                                                        ,21
      IF (K)3000,4000,3000
C     INSERT P AS FIRST PT. IN NEW PERCENT LINE.                         ,22
 3000 DO 3100 J=1,3                                                      ,23
      KN=KN+1                                                            ,24
 3100 B(KN)=P(J)
C
C     MOVE LAST PART OF OLD PERCENT LINE UP TO NEW POSITION.             ,25   ,26
 4000 NN=3*NN                                                            ,27
      DO 4100 J=1,NN                                                     ,28
      KN=KN+1                                                            ,29
      KO=KO+1                                                            ,30
 4100 B(KN)=B(KO)
C                                                                        ,31   ,32
      NU=0                                                               ,33
      GO TO 9000
C                                                                        ,34
 5000 NU=1                                                               ,35
 9000 RETURN                                                             ,36
      END                                                                ,37
```

```
      SUBROUTINE WBXC(B,NB,EP,P,KODE,S,D,Q,NU)
C
      DIMENSION B(1),P(3),S(1),Q(3),MU(3)
      DATA MU(2)/0/
C
C     USED BY WBX AND WBXD.  GIVEN A POINT P, CUT THE BODY WITH THE
C     PLANE X = P(1) AND GET BODY SECTION S (POINTS Y,Z).
C     FIND THE HORIZONTAL DISTANCE D FROM P TO S AND THE
C     CORRESP. POINT Q ON S (Q(1)=P(1) AND Q(3)=P(3)).
C     INPUTS-  B,NB,EP,P,KODE.  OUTPUTS-  S,D,Q,NU
C
      X=P(1)                                                            ,1
      Z=P(3)
C     CUT BODY WITH STATION PLANE                                       ,2
      CALL BCUTX(B,NB,X,EP,NS,S)                                        ,3
      NU=-5                                                             ,4
      IF (NS .LT. 2) GO TO 9000
C     INTERPOLATE WITH Z AS INDEPENDENT VARIABLE TO GET
C     CORRESPONDING Y.                                                  ,5    ,6    ,7
      CALL BITURP(S(2),S,2,NS,KODE,Z,Y,MU)                              ,8
      Q(1)=X                                                            ,9
      NU=MU                                                             ,10
      IF(MU)9000,5000,4000                                              ,11
 4000 IF(MU-2)6000,7000,9000                                            ,12
 5000 Q(2)=Y                                                            ,13
      Q(3)=Z                                                            ,14
      GO TO 8000                                                        ,15
 6000 Q(2)=S(2)                                                         ,16
      Q(3)=S(1)                                                         ,17
      GO TO 8000                                                        ,18
 7000 L=NS+NS                                                           ,19
      Q(2)=S(L)                                                         ,20
      Q(3)=S(L-1)
                                                                        ,21
 8000 D=P(2)-Q(2)
C                                                                       ,22
 9000 RETURN                                                            ,23
      END                                                               ,24
```

```
      SUBROUTINE WBXD(R,B,NB,KODE,P,Q,S,W,NU)
      DIMENSION R(4),B(1),P(3),Q(3),S(2,1),W(3,2),NU(3),C(4),LP(2)
C
C     USED BY WBX IF NECESSARY TO PROCESS A WING PERCENT POINT
C     WITH X-VALUE OUTSIDE THE RANGE OF BODY PERCENT LINE X-VALUES
C
C     W(I,1) IS A WING PERCENT POINT WHICH WAS PROCESSED OK BY WBX.
C     W(I,2) IS NEXT POINT, WHICH IS OUTSIDE BODY RANGE.
C     OTHER ARGUMENTS EXPLAINED IN WBX
C     RECALL THAT B IS HEADED BY TABLE OF CONTENTS.
C     S IS SCRATCH ARRAY FOR WBXC.
C
C     INPUTS  ARE B,NB,KODE,W,NU(2),NU(3)
C     OUTPUTS ARE P,Q,S,NU(1),NU(3)
C
      DATA SR,C/4HWBXD,-1.,3*0./
C
      NU=0                                                              ,1
      IF=B(3)                                                           ,2
      ISEC=B(6)                                                         ,3
      BXF=B(IF)                                                         ,4
      BXL=B(ISEC-3)
C     BXF AND BXL ARE EXTREME X-VALUES OF BODY PERCENT LINE 1           ,5
      WXF=W(1,1)                                                        ,6
      WXL=W(1,2)
C     TEST TO SEE IF BXF IN RANGE OF WING POINTS                        ,7
      IF((BXF-WXF)*(BXF-WXL))100,100,200
C     BXF WITHIN RANGE                                                  ,8
  100 R(1)=BXF                                                          ,9
      GO TO 300
C     BXF NOT IN RANGE.  ASSUME BXL IS.                                 ,10
  200 R(1)=BXL
C                                                                       ,11
  300 R(3)=R(1)                                                         ,12
      C(4)=R(1)
C     C IS A STATION PLANE                                              ,13
      CALL POLXN(C,W,3,2,0.,1,P,LP,L)
C     P IS INTERSECTION OF PLANE AND LINE SEGMENT W.                    ,14
      MU=1                                                              ,15
      IF (L .NE. 1) GO TO 8000
C
C     FIND HORIZONTAL DISTANCE FROM P TO BODY AND
C     CORRESPONDING POINT Q ON BODY.                                    ,16   ,17   ,18
      CALL WBXC(B,NB,0.,P,KODE,S,R(2),Q,NU)                             ,19
      MU=2                                                              ,20
      IF(NU)8000,1000,500
C     NU = 1 OR 2 MEANS WING POINT IS ABOVE OR BELOW THE BODY           ,21
  500 IF(NU-2)600,600,8000                                              ,22
  600 IF(R(2)) 1000,8000,1000
C                                                                       ,23
 1000 CONTINUE                                                          ,24
      MU=0                                                              ,25
      R(4)=R(2)                                                         ,26
```

```
      GO TO 9000
C
C     WRITE ERROR MESSAGE.                                                  ,27
 8000 LT=NU(2)                                                              ,28
      NU(3)=NU(3)-1                                                         ,29
      IF(LT .LE. 0  .OR.  NU(3) .LE. 0) GO TO 9000                          ,30   ,31   ,32
      WRITE(LT,8100)MU,NU(1),SR                                             ,33   ,34   ,35
 8100 FORMAT(6H0ERROR I6,6H, CODE I6,15H IN SUBROUTINE A6)
C
 9000 CONTINUE                                                              ,36
      NU=MU                                                                 ,37
      RETURN                                                                ,38
      END                                                                   ,39
```

```
      SUBROUTINE WBXUL(B,LI,LO,DAT,LGDEF,LOK,BTITLE,WTITLE,BAXIS,NU)
C
      DIMENSION B(1),DAT(2),LOK(3,9),NU(3),UPLO(2),PT(3),
     ,BTITLE(12),WTITLE(12),BAXIS(2)
C
      LOGICAL LGDEF(3,6)
      DATA SR,UPLO(1)/5HWBXUL,12H UPPER LOWER/,DEGRAD/0206712273406/
     ,,EP/1.E-5/
C
C      USED BY GECMD TO FIND THE INTERSECTIONS OF THE UPPER AND/OR
C      LOWER WING PERCENT LINES WITH THE BODY SURFACE, AND TO
C      TRUNCATE THE WING PERCENT LINES SO THAT THEY REPRESENT ONLY
C      THAT PART OF THE WING OUTSIDE THE BODY.
C
C      INPUTS ARE B,LI,LO,DAT,LGDEF,LOK,BTITLE,WTITLE,BAXIS,NU
C      OUTPUTS ARE B,LGDEF,LOK,NU
C
C      RECORD THAT THE WING-BODY INTERSECTION HAS BEEN REQUESTED
 100  LGDEF(1,5)=LGDEF(1,1)                                          ,1
      LGDEF(1,6)=LGDEF(1,2)
C                                                                   ,2
      NU=0
C      SEE IF BODY AND AT LEAST 1 SURFACE OF WING ARE DEFINED.      ,3
      IF ((LGDEF(2,1) .OR. LGDEF(2,2)) .AND. LGDEF(2,3)) GO TO 300  ,4    ,5    ,6
      WRITE (LO,210)                                                ,7    ,8
 210  FORMAT(47H1WING-BODY INTERSECTION CANNOT BE FOUND BECAUSE
     1/44H EITHER THE BODY OR THE WING IS NOT DEFINED. )
      NU=1                                                          ,9
      GO TO 9000
C                                                                   ,10
 300  READ(LI,310)WBKODE,EPS                                        ,11   ,12   ,13
 310  FORMAT(6F10.0)
      IF(ABS(EPS)-EP) 320,330,330                                   ,14
 320  EPS=EP                                                        ,15
 330  CONTINUE                                                      ,16
      KODE=WBKODE
C      KODE = 1 IF BODY IS POLYGONAL IN SHAPE, =2 IF SMOOTH.
                                                                    ,17
      WRITE(LO, 340)KODE,EPS                                        ,18   ,19   ,20
 340  FORMAT(/71H0FIND THE INTERSECTION OF EACH WING PERCENT LINE WITH T
     1HE BODY SURFACE. // 1H0 10X 6HCODE = I3/1H0 10X11HTOLERANCE =F7.4/
     2/48H0LET P BE A POINT WHICH LIES ON THE BODY SURFACE /
     356H AND Q A POINT ON A LINE SEGMENT CONNECTING TWO ADJACENT /
     472H WING PERCENT LINE POINTS, SUCH THAT P AND Q HAVE THE SAME X-CO
     5ORDINATE. /
     649H0THE PROGRAM ITERATES UNTIL P AND Q HAVE THE SAME /
     742H (OR NEARLY THE SAME) Y AND Z COORDINATES. /
     834H THEN P IS THE INTERSECTION POINT. /
     952H0IF P AND Q DIFFER BY MORE THAN THE GIVEN TOLERANCE, /
     A30H A WARNING MESSAGE IS WRITTEN. /
     B69H0THE SHAPE OF THE BODY SURFACE BETWEEN MERIDIAN LINES (IN ANY S
     CECTION /
     D59H PARALLEL TO THE YZ PLANE) IS DETERMINED BY THE GIVEN CODE. /
```

```
     E9X65HIF CODE = 1, LINEAR INTERPOLATION IS USED BETWEEN MERIDIAN LI
     FNES.
     G5X47HIF CODE = 2, BIQUADRATIC INTERPOLATION IS USED.
C
      NB=LOK(2,5)
C      NB IS NO. OF BODY PERCENT LINES.  RESERVE STORAGE FOR SCRATCH
C      ARRAY OF LENGTH 2*NB FOR SECTIONS THRU BODY.  STARTS IN B(NAMS).      ,21
      MU=2                                                                   ,22
      IF (IRSERV(NB+NB,B,NAMS))8000,8000,400
C      PACK BUFFER                                                           ,23
 400  IF (IPACK(B))8000,500,500                                              ,24
 500  LB=LOK(1,5)
C      BODY PERCENT LINES START IN B(LB)
C
C      MAIN LOOP ON K IS FOR UPPER WING, THEN LOWER WING.
C                                                                            ,25
      DO 6000 K=1,2
C
C      SEE IF THIS WING SURFACE IS DEFINED                                   ,26
      IF (.NOT. LGDEF(2,K)) GO TO 6000
C      WING DEFINED.  FIND NW = NO. OF WING PERCENT LINES.                   ,27   ,28   ,29
      NW=LOK(2,K)
C
C      RESERVE STORAGE FOR WING-BODY INTERSECTION POINTS                     ,30
      MU=3                                                                   ,31
      NWBE=3*NW                                                              ,32
      K7=K+7                                                                 ,33
      IF (IRSERV(NWBE,B,LOK(1,K7)))8000,8000,700                             ,34
 700  LOK(3,K7)=NWBE                                                         ,35
      LWB=LOK(1,K7)                                                          ,36
      LB=LOK(1,5)                                                            ,37
      NAMS=NAMS
C      FIND LW = LOCATION OF WING PERCENT LINES (START IN B(LW))             ,38
      LW=LOK(1,K)
C
C      FIND INTERSECTIONS AND TRUNCATE WING PERCENT LINES                    ,39
      CALL WBX(B(LB),NB,B(LW),NW,EPS,KODE,B(NAMS),B(LWB),NU)                 ,40
      MU=4                                                                   ,41
      IF (NU)8000,800,8000
C
C      WRITE HEADING                                                         ,42
 800  WRITE (LO,810)BTITLE,WTITLE,DAT,UPLO(K),BAXIS                         ,43   ,44   ,45
 810  FORMAT(1H1 12A6/1H 12A6/54X 2A6/13H INTERSECTIONS OF A6,
     ,43H WING PERCENT CHORD LINES WITH BODY SURFACE /
     ,41H RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( F9.4,1H,F9.4,1H)/
     ,13HONO.  PERCENT 6X 1HX 9X 1HY 9X 1HZ 10X 3HRHO 6X 5HTHETA//)
C
C      WRITE THE INTERSECTION POINTS
      IPCT=LW                                                                ,46
      IWBX=LWB
C                                                                            ,47
      DO 1000 J=1,NW                                                         ,48
      PCT=B(IPCT)                                                            ,49
      IPCT=IPCT+3                                                            ,50
```

```
      CALL TRAV(PT,B(IWBX),3)                                          ,51
      IWBX=IWBX+3                                                      ,52
      IF (PT .GT. 1.E+29) GO TO 900                                    ,53   ,54   ,55
      DY=PT(2)-BAXIS(1)                                                ,56
      DZ=PT(3)-BAXIS(2)                                                ,57
      RHO=SQRT(DY*DY+DZ*DZ)                                            ,58
      IF(RHO)820,820,830                                               ,59
 820  THETA=0.                                                         ,60
      GO TO 850                                                        ,61
 830  THETA=ATN1(ABS(DY),DZ)*DEGRAD                                    ,62
 850  WRITE(LO,860)J,PCT,PT,RHO,THETA                                  ,63   ,64   ,65
 860  FORMAT(1H I2,F10.4,1X 3F10.4,2X 2F10.4)
      GO TO 1000                                                       ,66
 900  WRITE(LO,910)J,PCT                                               ,67   ,68   ,69
 910  FORMAT(1H I2,F10.4,6X 15HNO INTERSECTION)
 1000 CONTINUE
C
C      RECORD THAT THE INTERSECTIONS HAVE BEEN FOUND                   ,70   ,71
 2000 LGDEF(2,K+4)=.TRUE.
C                                                                      ,72
      DO 2100 J=1,2                                                    ,73
 2100 LOK(J,K+2)=LOK(J,K)
C                                                                      ,74   ,75
      NLAST=B(IPCT-2)                                                  ,76
      LLAST=B(IPCT-1)                                                  ,77
      LOK(3,K+2)=LLAST+3*NLAST-1
C
C                                                                      ,78
 6000 CONTINUE
C                                                                      ,79   ,80
      MU=0                                                             ,81
      GO TO 8500
C
C      WRITE ERROR MESSAGE                                             ,82
 8000 KE=MERR(NU,NU+1,MU,SR)
C                                                                      ,83
 8500 NU=MU
C                                                                      ,84
 9000 RETURN                                                           ,85
      END                                                              ,86
```

```
      SUBROUTINE WBXX(B,NB,W,NW,EPS,KODE,S,BW,NU)
C
C     SAME AS WBX, EXCEPT DOES NOT STORE TRUNCATED LINES IN B.

      DIMENSION B(1),W(1),S(2,1),BW(3,1),NU(3),
     1 Q(3,2),R(10),DUM(3)
     2,WGPTS(3,2),UVEC(3),WP(3,2)
     3,PTCNWG(3)
      LOGICAL INTD
C
C     USED BY TFLAT TO FIND THE INTERSECTIONS OF WING PERCENT
C     LINES WITH THE BODY SURFACE.

C     B IS THE ARRAY OF NB BODY PERCENT LINES.
C     W IS THE ARRAY OF NW WING PERCENT LINES.
C
C     A PERCENT LINE OF N POINTS HAS 3*N ELEMENTS
C             X1,Y1,Z1, X2,Y2,Z2, ........., XN,YN,ZN
C     THE FIRST 3*J CELLS OF AN ARRAY B OF J PERCENT LINES FORMS A TABLE OF
C     CONTENTS TO THE REST OF THE ARRAY.  ELEMENTS 1,4,7,...   OF B ARE LABELS
C     FOR PERCENT LINES 1,2,3,...   ELEMENTS 2,5,8,... ARE THE NUMBER OF
C     3-DIMENSIONAL POINTS IN PERCENT LINES 1,2,3,...   ELEMENTS 3,6,9,...   ARE
C     THE STARTING LOCATIONS IN B OF PERCENT LINES 1,2,3,...
C     FOR EXAMPLE, IF B(7) = 30., B(8) = 12. AND B(9) = 49., THEN THE PERCENT
C     LINE WITH THE LABEL 30. CONTAINS 12 POINTS AND STARTS IN B(49).
C
C     EPS IS THE TOLERANCE IN LOCATING INTERSECTION POINTS.
C       A WARNING MESSAGE IS WRITTEN IF THE ERROR IS GREATER THAN EPS.
C     KODE =1 IF THE BODY IS POLYGONAL IN CROSS SECTON, = 2 IF SMOOTH.
C     S IS A SCRATCH ARRAY AT LEAST 2*NB CELLS LONG.  USED FOR 2-DIMENSIONAL
C       CROSS SECTIONS THRU THE BODY.
C     BW IS AN OUTPUT ARRAY OF BODY-WING INTERSECTION POINTS.  IF THE JTH
C       WING PERCENT LINE DOES NOT INTERSECT THE BODY, THEN
C       BW(1,J) = BW(2,J) = BW(3,J) = 1.E+30.
C
C     NU IS AN ERROR ARRAY WHICH IS PASSED DOWN TO LOWER LEVEL SUBROUTINES.
C     NU(1), AN OUTPUT, IS ZERO FOR SUCCESS, NON-ZERO FOR FAILURE.
C     NU(2), AN INPUT, MAY BE INTERPRETED AS A  TAPE NUMBER ON WHICH TO
C       WRITE AN ERROR MESSAGE, IF GREATER THAN ZERO.
C     NU(3), BOTH AN INPUT AND AN OUTPUT, IS AN ERROR COUNTER AND ERROR MESSAGE
C       LIMITER.  SOME LOWER LEVEL SUBROUTINES MAY SUBTRACT 1 FROM NU(3) IF AN
C       ERROR OCCURS AND THEN WRITE AN ERROR MESSAGE IF NU(3) AND NU(2) ARE BOTH
C       GREATER THAN ZERO.
C
C     INPUTS ARE B,NB,W,NW,EPS,KODE,NU(2),NU(3). OUTPUTS BW,S,BW,NU(1),NU(3)
C
      DATA SR,DUM/3HWBX,3*1.E+30/
C
      R(10)=1.E-7
C
C     CUT THE BODY WITH A STATION PLANE THRU EACH WING PERCENT POINT IN TURN
C     UNTIL THE INTERSECTION POINT IS ISOLATED BETWEEN 2 ADJACENT WING POINTS.
C     THEN ITERATE UNTIL THE X-VALUE OF THE INTERSECTION IS FOUND (WITHIN EPS).
```

```
C                                                                       ,1
 100  DO 6000 IW=1,NW
C     FOR EACH WING PERCENT LINE IN W.
C                                                                       ,2
      INTD=.FALSE.                                                      ,3
      L=IW*3
C     NWP IS NO. OF PTS. IN THE IW PERCENT LINE AND LWP IS THE LOCATION OF THE
C     FIRST POINT IN W.  LWP IS UPDATED IN THE LOOP BELOW.              ,4
      NWP=W(L-1)                                                        ,5
      LWP=W(L)
C                                                                       ,6
 200  DO 1200 K=1,NWP
C     FOR EACH POINT W(LWP) IN THE PERCENT LINE.
C
C     FIND THE HORIZONTAL DISTANCE D FROM POINT W(LWP) TO THE BODY AND THE
C     CORRESPONDING  POINT Q ON THE BODY.                               ,7
      EPC=AMAX1(ABS(W(LWP)),1.)*1.E-6                                   ,8
      CALL WBXC(B,NB,EPC,W(LWP),KODE,S,D,Q,NU)                          ,9
      IF(NU)250,500,300                                                 ,10
 250  MU=6
C     NU = -5 MEANS BODY IS NOT DEFINED AT STATION X                    ,11
      IF(NU+5)9000,400,9000                                             ,12
 300  MU=7
C     NU = 1 OR 2 MEANS WING POINT IS ABOVE OR BELOW THE BODY           ,13
      IF(NU-2)310,310,9000                                              ,14
 310  MU=8                                                              ,15
      IF(D)510,9000,510
C     SEE IF PREVIOUS POINT WAS CK.                                     ,16
  400 IF(INTD) GO TO 1250                                               ,17   ,18   ,19
      GO TO 1000                                                        ,20
 500  IF(D .EQ. 0.) GO TO 1500                                          ,21   ,22   ,23
 510  IF(INTD) GO TO 600                                                ,24   ,25   ,26
      INTD=.TRUE.                                                       ,27
      GO TO 700
C
C     SEE IF THIS POINT AND PREVIOUS POINT ARE ON OPPOSITE SIDES OF BODY SURFACE   ,28
  600 IF(D*DP)650,700,700
C     OPPOSITE SIDES
C     THE INTERSECTION PT. IS BETWEEN W(LWPP) AND W(LWP)                ,29
 650  R(4)=D                                                            ,30
      GO TO 1600
C
C     SAME SIDE OR ELSE FIRST TIME WBXC SUCCEEDED.                      ,31
 700  DP=D
C                                                                       ,32
 1000 LWPP=LWP                                                          ,33
      LWP=LWP+3
C                                                                       ,34
 1200 CONTINUE
C                                                                       ,35   ,36
      GO TO 1300
C
C     THE X-VALUE (=W(LWP)) OF THIS WING PERCENT POINT IS OUTSIDE
C     THE RANGE OF BODY X-VALUES.  HOWEVER, THE PREVIOUS WING
```

```
C     PERCENT POINT WAS OK.  ONLY 1 OF THE BODY END POINTS CAN
C     LIE BETWEEN W(LWPP) AND W(LWP).  FIND THIS POINT, INTERSECT
C     THE WING PERCENT LINE SEGMENT WITH THE CORRESPONDING
C     BODY STATION PLANE TO GET A POINT ON THE WING P.C. LINE.
C     CALL THIS POINT  PTONWG.
C     THE DISTANCE FROM THIS PT. TO THE BODY SURFACE WILL BE IN
C     R(2)=R(4).  THE X-VALUE WILL BE IN R(1)=R(3).
C                                                                       ,37
 1250 CALL WBXD(R,B,NB,KODE,PTONWG,Q,S,W(LWPP),NU)                      ,38
      MU=5                                                              ,39
      IF(NU)9000,1260,9000                                              ,40
 1260 D=R(2)                                                            ,41
      IF(D*DP)1600,1600,1300
C
C     NO INTERSECTION.  LEAVE PERCENT LINE UNCHANGED.  STORE X,Y,Z = 1.E+30
C     AS INTERSECTION POINT.                                            ,42
 1300 KOUT=0                                                            ,43
      CALL TRAV(BW(1,IW),DUM,3)                                         ,44
      GO TO 5000
C
C     WING POINT W(LWP) LIES ON THE BODY SURFACE.                       ,45
 1500 KOUT=K                                                            ,46
      GO TO 4500
C                                                                       ,47
 1600 CONTINUE                                                          ,48
      CALL TRAV(WGPTS,W(LWPP),6)                                        ,49
      IF(MU .NE. 5) GO TO 1650                                          ,50   ,51   ,52
      CALL TRAV(WGPTS(1,2),PTONWG,3)
C     WGPTS CONTAINS THE COORD. OF 2 CONSECUTIVE POINTS ON THE
C     WING PERCENT LINE WHICH LIE ON OPPOSITE SIDES OF THE BODY SURFACE.
C     FIND THE INTERSECTION OF THE LINE SEGMENT WGPTS WITH THE BODY
C     SURFACE AS FOLLOWS,  CHOOSE A POINT (IN WP) WHICH IS SOME
C     DISTANCE ALONG WGPTS FROM ITS 1ST END POINT.  (A UNIT VECTOR
C     ALONG WGPTS WILL BE UVEC).  USE WBXC TO FIND THE HORIZONTAL
C     DISTANCE (IN Y-DIRECTION) FROM WP TO THE BODY SURFACE.
C     ITERATE ON DISTANCE ALONG WGPTS UNTIL WP LIES ON BODY SURFACE.
C
C     USE FROOTA TO TAKE CARE OF THE BOOK-KEEPING FOR THE ITERATION
C     AND TO PREDICT DISTANCES ALONG WGPTS TO USE FOR NEXT ITERATION.
C     FROOTA WILL SET JR = 0 WHEN INTERSECTION IS FOUND.
C                                                                       ,53
 1650 CONTINUE                                                          ,54
      R(1)=0.                                                           ,55
      R(2)=DP                                                           ,56
      R(3)=UVECN(WGPTS,WGPTS(1,2),UVEC,0,3)
C     R(3) = LENGTH OF THE LINE SEGMENT WGPTS
C     R(4) WAS SET TO D PREVIOUSLY                                      ,57
      R(9)=0.                                                           ,58
      MU=9                                                              ,59
      IF(R(3))9000,9000,1700                                            ,60
 1700 MU=10
C                                                                       ,61
      DO 3000 KR=1,100                                                  ,62
      CALL FROOTA(R,JR)
```

```
C     R(1) AND R(3) ARE NEW TRIAL VALUES OF DISTANCE ALONG WGPTS                ,63
                                                                                ,64
      DO 1900 I=1,2                                                             ,65
      II=I+I                                                                    ,66
      DO 1800 J=1,3                                                             ,67   ,68
 1800 WP(J,I)=WGPTS(J,1)+R(II-1)*UVEC(J)
      CALL WBXC(B,NB,EPC,WP(1,I),KODE,S,R(II),Q(1,I),NU)                        ,69
C     R(2) AND R(4) ARE CORRESPONDING Y-DISTANCES OF WP FROM BODY SURFACE       ,70   ,71   ,72
      IF(NU .LT. 0  .OR.  NU .GT. 2) GO TO 9000
 1900 CONTINUE                                                                  ,73   ,74
C                                                                               ,75
      IF(JR)2000,3100,3000                                                      ,76
 2000 MU=11
      GO TO 9000                                                                ,77
C
 3000 CONTINUE
C                                                                               ,78   ,79
C     DID NOT CONVERGE                                                          ,80
      MU=12
      GO TO 9000                                                                ,81
C                                                                               ,82
 3100 IF(ABS(R(2))-EPS) 4000,4000,3110                                          ,83
 3110 LTAPE=NU(2)                                                               ,84
      NU(3)=NU(3)-1                                                             ,85   ,86   ,87
      IF(LTAPE .LE. 0  .OR.  NU(3) .LE. 0)GO TO 4000                            ,88   ,89   ,90   ,91   ,92   ,93
      WRITE(LTAPE,3120)IW,EPS,(WP(I,1),I=1,3)

 3120 FORMAT(/42H0*** INTERSECTION OF WING PERCENT LINE NO.
     1 I3,40H WITH BODY NOT WITHIN GIVEN TOLERANCE OF
     2 F10.6/27H POINT ON WING HAS COORD. ( 3F10.4,1H) )
C
C     THE INTERSECTION (IN Q) HAS BEEN FOUND AND LIES BETWEEN
C     POINTS K-1 AND K.
 4000 KOUT=K-1
C     THE FIRST KOUT POINTS ARE INSIDE THE BODY AND ARE TO
C     BE REMOVED.
C     STORE THE INTERSECTION POINT.                                             ,94
 4500 CALL TRAV(BW(1,IW),Q,3)
C                                                                               ,95
 5000 CONTINUE
C                                                                               ,96
 6000 CONTINUE                                                                  ,97   ,98
      MU=0                                                                      ,99
      GO TO 9900
C
C     WRITE ERROR MESSAGE.                                                      ,100
 9000 LT=NU(2)                                                                  ,101
      NU(3)=NU(3)-1                                                             ,102
      IF(LT .LE. 0 .OR. NU(3) .LE.0) GO TO 9900                                 ,103  ,104  ,105
      WRITE(LT,9100)MU,NU(1),SR,IW,K                                            ,106  ,107  ,108
 9100 FORMAT(6H0ERROR I6,6H, CODE I6,15H IN SUBROUTINE A6/
     , 18H WING PERCENT LINE I5,7H, POINT I5)
 9900 NU=MU                                                                     ,109
      RETURN                                                                    ,110
```

END ,111

```
      SUBROUTINE WING
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KASE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/IJ,NPTS(16),X(90,16),Y(90,16),Z(90,16),KPNT,VALUE(5),
     1YCEPT(16),SLOPE,KPANEL(2,15),XCOR(16,16),YCOR(16,16),ZCOR(16,16),Y
     2INT(2,15),YINT(2,15),ZINT(2,15),XCEN(15,15),YCEN(15,15),ZCEN(15,15
     3),XCON(15,15),YCON(15,15),ZCON(15,15),AREA(15,15),ARAT(15,15),THET
     4A(15,15),ALPHA(15,15),CHORD(15,15),XFOIL(16,25,2),ZFOIL(16,25,2),X
     5NUM(16,2),XTAB1(25),ZTAB1(25),XTAB2(25),ZTAB2(25)
      DIMENSION ND3(9455)
      EQUIVALENCE (ND3(1),NPTS(1))

C     CONTROL PROGRAM FOR WING PANELING SUBROUTINES
      DATA NAME/6HDEFINE/

C     DEFINE CONSTANTS
C     NWING IS NUMBER OF WING PANELS      KODEW,
C     KODEWU ARE INPUT ERROR CODES
C     IF KODEW SET = 1 IN INPUTW THEN TAPE ERROR
C     ENCOUNTERED ON SCRATCH TAPE      IF KODEWU
C     SET = 1, ERROR ENCOUNTERED ON INPUT CARDS
C     FOR THICK WING (OR CAMBERED SURFACE)
 10   NWING=0                                                          ,1
      KODEW=0                                                          ,2
      KODEWU=0                                                         ,3
      IJ=0

C     INITIALIZE COMMON ARRAYS                                         ,4
      DO 20 I=1,9455                                                   ,5
 20   ND3(I)=0

C     CALL SUBROUTINES                                                 ,6     ,7
 30   CALL INPUTW                                                      ,8
      IF (KODEW) 200,40,110                                            ,9
 40   CALL CRNRW                                                       ,10
      IF (KODEW) 200,45,100                                            ,11
 45   CALL AREAP(XCOR,YCOR,ZCOR,AREA,ARAT,NPLN1,NPER1)                 ,12
      CALL CENTRD (XCOR,YCOR,ZCOR,ARAT,XCEN,YCEN,ZCEN,
     1NPLN1,NPER1)                                                     ,13
      CALL CNTRLW                                                      ,14
      CALL CHORDW                                                      ,15
      IF (KODEW) 200,50,100                                            ,16
 50   IJ=IJ+1                                                          ,17
      CALL INPUTW                                                      ,18
      IF (KOPTF) 200,105,60                                             19
 60   IF (KODEWU) 200,80,120                                           ,20
   80 CALL SLOPEW                                                      ,21
      IF (KODEW) 200,105,100                                           ,22
 100  WRITE (NWRITE,920)                                               ,23    ,24
 105  CALL OUTPTW                                                      ,25
```

```
      IF (KODEW) 150,150,200                                          ,26
110   WRITE (NWRITE,900)                                              ,27    ,28
      GO TO 200                                                        29
120   WRITE (NWRITE,910)                                              ,30    ,31
      GO TO 200                                                       ,32
150   RETURN                                                          ,33
 200  READ (NREAD,890) KARD                                            34    ,35    ,36
      IF (KARD-NAME) 200,210,200                                      ,37
 210  BACKSPACE NREAD                                                 ,38
      KODEW=-1                                                         39
      GO TO 150                                                       ,40
 890  FORMAT (A6)
900   FORMAT (1H1,9X,31HERROR MESSAGE - SUBROUTINE WING/10X,
     131HIMPROPER PLANAR WING DEFINITION)
910   FORMAT (1H1,9X,31HERROR MESSAGE - SUBROUTINE WING/10X,
     152HIMPROPER THICK WING (OR CAMBERED SURFACE) DEFINITION)
 920  FORMAT (1H1,9X,13HERROR MESSAGE/10X,
     149HPANELLING DATA NOT COMPLETED DUE TO PROGRAM ERROR/10X,
     238HINCOMPLETE PANELLING DATA TO BE OUTPUT)
      END                                                             ,41
```

```
      SUBROUTINE WING1(B,LI,LO,DAT,LLETE,LOGUL,NAF,NPL,LAFN,LPCT,KOD,
     1 CHD,EPS,TITLE,NU)
      DIMENSION B(1),DAT(2),KOD(2),NU(3),NLTEP(4),LPLF(2),HLETE(2,2),
     1 TITLE(12)
      LOGICAL LOGUL(2)
C     WING1 IS USED BY GEOMD.  IT READS DATA CARDS WHICH DEFINE
C     A WING, CHECKS THE DATA, PUTS IT IN STANDARD FORM FOR
C     LATER USE BY WING2 AND GEOMD.
C
C     B IS BUFFER IN WHICH DATA IS STORED
C     LOGUL(1) = TRUE IF UPPER WING IS DEFINED
C     LOGUL(2) = TRUE IF LOWER WING IS DEFINED
C     NAF = NO. OF AIRFOILS
C     NPL = NO. OF PERCENT LINES DESIRED
C     LAFN.   A TABLE IS STORED IN B(LAFN) WHICH HAS THE
C          LOCATIONS IN B WHERE AIRFOILS ARE STORED, AND
C          NO. OF POINTS IN EACH AIRFOIL.
C     LLETE.   THE LEADING AND TRAILING EDGE POINTS OF EACH
C           AIRFOIL ARE STORED IN B(LLETE).
C     LPCT.    THE PERCENTS AT WHICH PERCENT LINE ARE TO BE
C          GENERATED ARE STORED IN B(LPCT).
C     NU IS ERROR INDICATOR.
C
      DATA HLETE(1,1),HLETE(1,2),WING/7HLEADING,8HTRAILING,5HWING1/,
     .HGIVEN,HUSED/6HGIVEN),6HUSED) /,MAX/150/
C
 10   LOGUL(1)=.TRUE.                                                  ,1
      LOGUL(2)=.TRUE.                                                  ,2
      NU=0
C
C     READ AND WRITE TITLE CARD (CARD 1)                               ,3
      READ (LI,310)TITLE                                               ,4    ,5    ,6
 310  FORMAT(13A6,A2)
      WRITE (LO,320)DAT,TITLE                                          ,7    ,8    ,9
 320  FORMAT(16H1WING DEFINITION 38X,2A6/1H014A6)
C
C     READ CARDS 2,3
      READ (LI,400)PNLE,PNTE,AFN,PLN,WUL,CHD,PCODE,ACODE,EPS           ,10   ,11   ,12
 400  FORMAT(6F10.0)
      NLTEP(1)=PNLE                                                    ,13
      NLTEP(2)=PNTE                                                    ,14
      NAF=AFN                                                          ,15
      NPL=PLN                                                          ,16
      NLTEP(3)=NAF                                                     ,17
      NLTEP(4)=NPL                                                     ,18
      WRITE (LO,410)NLTEP                                              ,19   ,20   ,21
 410  FORMAT(  /17H0THE PLANFORM HAS I3,28H POINTS ON THE LEADING EDGE,
     ./I20,29H POINTS ON THE TRAILING EDGE.
     ./I4,20H AIRFOILS ARE GIVEN.
     .  I4,35H PERCENT LINES ARE TO BE GENERATED.)
C
      KOD(1)=ACODE                                                     ,22
      KOD(2)=PCODE                                                     ,23
```

```
      IF(EPS)411,411,412                                                ,24
 411  EPS=.01                                                           ,25
 412  WRITE(LO,413)PCODE,ACODE,EPS,CHD                                  ,26   ,27   ,28
 413  FORMAT( 8HOPCODE = F3.0, 9H, ACODE = F3.0, 7H, EPS = F10.6,
     X 7H, CHD = F7.4)

 420  IF (WUL*(WUL-2.))440,450,430
C     WUL NOT 0.,1., OR 2.                                              ,29
 430  MU=1                                                              ,30
      GO TO 9000
C     UPPER WING                                                        ,31
 440  LOGUL(2)=.FALSE.                                                  ,32
      WRITE(LO,441)                                                     ,33   ,34
 441  FORMAT(30H UPPER WING SURFACE IS DEFINED)
      GO TO 480
C     WUL = 0. OR 2.                                                    ,35
 450  IF (WUL)470,470,460
C     LOWER WING                                                        ,36
 460  LOGUL(1)=.FALSE.                                                  ,37
      WRITE(LO,461)                                                     ,38   ,39
 461  FORMAT(30H LOWER WING SURFACE IS DEFINED)
      GO TO 480
C     UPPER AND LOWER WING                                              ,40
 470  WRITE(LO,471)                                                     ,41   ,42
 471  FORMAT(42H UPPER AND LOWER WING SURFACES ARE DEFINED)
C
 480  CONTINUE
C
C     CHECK FOR REASONABLE VALUES                                       ,43
 500  DO 520 I=1,4                                                      ,44
      I1=NLTEP(I)-1                                                     ,45
      IF (I1*(I1-MAX))520,510,510
C     WRITE ERROR MESSAGE  NO. 2 AND GO ON                              ,46
 510  IERR=MERR(0,1,2,WING)                                             ,47
 520  CONTINUE
C
C     PACK BUFFER (B) SO STORAGE FOR AIRFOIL NAMES WILL
C     NOT LATER BE DISTURBED                                            ,48   ,49
      LEFT=IPACK(B)
C     RESERVE STORAGE FOR NAMES OF AIRFOILS                             ,50
      LAFN=IRSERV(4*NAF,B,LAFN)                                         ,51
      IF (LAFN)550,550,560                                              ,52
 550  MU=3                                                              ,53
      GO TO 9000
C
C     RESERVE STORAGE FOR INTERSECTION OF LE AND TE WITH AIRFOILS       ,54
 560  LLETE=IRSERV(4*NAF,B,LLETE)                                       ,55
      IF (LLETE)570,570,600                                             ,56
 570  MU=4                                                              ,57
      GO TO 9000
C
C     RESERVE STORAGE FOR PLANFORM POINTS, READ THEM.                   ,58
 600  DO 640 I2=1,2                                                     ,59
      I5=2*NLTEP(I2)                                                    ,60
```

```
      IF (IRSERV(I5,B,LPLF(I2)))610,610,620                                  ,61
 610  MU=5                                                                   ,62
      GO TO 9000                                                             ,63
 620  I3=LPLF(I2)                                                            ,64
      I4=I3+I5-1                                                             ,65
      READ (LI,400)(B(I),I=I3,I4)
C     LE,TE POINTS START IN B(LPLF(1)),B(LPLF(2))                            ,66   ,67   ,68   ,69   ,70
 640  WRITE (LO,650)HLETE(1,I2),HLETE(2,I2),HGIVEN,(B(I),I=I3,I4)            ,71   ,72   ,73   ,74   ,75   ,76
                                                                             ,77
 650  FORMAT(  /10HOPLANFORM A6,A3,13HEDGE POINTS ( A6,12H ARE (X,Y) =/
     .  (1H 3(2F10.4,4X)))
C
C     READ AIRFOILS ANC THEIR LOCATION ON PLANFORM.
C     FIND INTERSECTION OF AIRFOILS WITH LE AND TE.
C     SCALE AIRFOILS TC FIT ON PLANFORM
 700  CALL WING1A(B,B(LAFN),LPLF,NLTEP,LOGUL,B(LLETE),LI,LO,
     .TITLE,DAT,NU)                                                          ,78
      IF(NU)710,800,71C                                                      ,79
 710  MU=7                                                                   ,80
      GO TO 9000
C
C     WRITE PLANFCRM DETERMINED BY AF INTERSECTIONS WITH LE, TE              ,81
 800  LL=LLETE                                                               ,82
      WRITE (LO,320) DAT,TITLE                                               ,83   ,84   ,85
      DO 810 I2=1,2                                                          ,86
      L1=LL                                                                  ,87
      L2=L1+4*NAF-4                                                          ,88
      WRITE (LO,650)HLETE(1,I2),HLETE(2,I2),HUSED,
     . (B(I),B(I+1),I=L1,L2,4)                                               ,89   ,90   ,91   ,92   ,93   ,94

 810  LL=LL+2
C
C     RELEASE STORAGE                                                        ,95   ,96
      DO 910 I=1,2                                                           ,97
      I3=LPLF(I)                                                             ,98
      IF (IDLETE(B(I3)))920,910,920                                          ,99
 910  CONTINUE                                                               ,100  ,101
      GO TO 1000                                                             ,102
 920  MU=8                                                                   ,103
      GO TO 9000
C
C     READ PERCENTS FOR WHICH PERCENT LINES ARE TO BE GENERATED.
C     CHECK BETWEEN 0, 100.  CONVERT TO PER-UNIT VALUES.
C     STORE IN B(LPCT).                                                      ,104
 1000 LPCT=IRSERV(NPL,B,LPCT)                                                ,105
      IF (LPCT)1010,1010,1050                                                ,106
 1010 MU=9                                                                   ,107
      GO TO 9000
C
C     READ CARC 5                                                            ,108
 1050 I3=LPCT+NPL-1                                                          ,109
      READ (LI,400)(B(I),I=LPCT,I3)                                          ,110  ,111  ,112  ,113  ,114
      WRITE (LO,1060)(B(I),I=LPCT,I3)                                        ,115  ,116  ,117  ,118  ,119
 1060 FORMAT(/56HOTHE FOLLOWING PERCENT CHORD LINES ARE TO BE CONSTRUCTE
```

```
     .D/(1H 6F10.4))
C
      DO 1200 I=LPCT,I3                                                  ,120
      P=B(I)                                                             ,121
      IF (P*(P-100.))1200,1200,1150                                      ,122
 1150 MU=10                                                              ,123
      GO TO 9000                                                         ,124
 1200 CONTINUE                                                           ,125  ,126
      GO TO 9100
C
C     ERROR
 9000 IERR=MERR(NU,NU+1,MU,WING)                                         ,127
      NU=MU                                                              ,128
C
 9100 RETURN                                                             ,129
      END                                                                ,130
                                                                         ,131
```

```
      SUBROUTINE WING1A(B,NAME,LPLF,NLTEP,LOGUL,PT,LI,LO,TITLE,DAT,MU)
C
C     USED BY WING1 TO READ AIRFOILS AND SCALE THEM TO FIT ON PLANFORM
      DIMENSION B(1),NAME(2,2,1),NLTEP(3),PT(2,2,1),RUB(2,2),LQ(2),
     1YLTE(2),NAFUL(2),CLINE(3),MX(2),HUL(2),LPLF(2),
     ,TITLE(12),DAT(2)
      LOGICAL  LOGUL(2)
C
C     THE POINTS FOR THE JTH AIRFOIL WILL BE STORED IN B(NAME(1,I,J))
C       WHERE I=1 FOR UPPER, I=2 FOR LOWER AIRFOIL.
C       THE CORRESPONDING NO. OF ELEMENTS (TWICE NO. OF POINTS) IS IN
C         NAME(2,I,J).
C     LE POINTS STORED IN B(LPLF(1)), TE IN B(LPLF(2)).
C     NLTEP(1) = NO PTS IN LE, NLTEP(2) = NO PT IN TE.
C     NLTEP(3) = NO AIRFOILS.
C     LOGUL(1) = .TRUE. IF UPPER AIRFOIL DEFINED.
C     LOGUL(2) = .TRUE. IF LOWER AIRFOIL DEFINED.
C
      DATA HUL(1)/12H UPPER LOWER/
C
 10   MU=0                                                              ,1
      NAF=NLTEP(3)
C     CLEAR NAME ARRAY                                                  ,2
      DO 100 I=1,NAF                                                    ,3    ,4
      DO 100 J=1,2                                                      ,5    ,6
      DO 100 K=1,2                                                      ,7    ,8
 100  NAME(K,J,I)=0
C                                                                       ,9    ,10   ,11   ,12
      DO 8000 IA=1,NAF
C                                                                       ,13   ,14
      WRITE(LO,110) DAT,TITLE                                           ,15   ,16   ,17
  110 FORMAT(16H1WING DEFINITION 38X 2A6/1H0 12A6//)
      READ (LI,200)AFK,BETA,YLTE,AFNU,AFNL                              ,18   ,19   ,20
 200  FORMAT(6F10.0)
      WRITE (LO,210)IA,AFK,BETA,YLTE,AFNU,AFNL                          ,21   ,22   ,23
 210  FORMAT(  /12H0AIRFOIL NO. I3/6H AFK =  F4.0,12H,    AFANG = F9.4,
     X9H,    YL =F10.4,9H,    YT =F10.4/7H AFNU =F3.0,12H,     AFNL =
     X F3.0)
      NAFUL(1)=AFNU                                                     ,24
      NAFUL(2)=AFNL                                                     ,25
      IC=AFK                                                            ,26
      IF (IC*(IC-4))320,310,310                                         ,27
 310  MU=5                                                              ,28
      GO TO 9000
C
C     FIND LE AND TE X VALUES CORRESPONDING TO YLTE.
C     (ONLY 1 NEEDED IF IC=1 OR 2)                                      ,29
 320  CLINE(1)=0.                                                       ,30
      CLINE(2)=-1.                                                      ,31
      DO 350 I=1,2                                                      ,32   ,33
      CLINE(3)=YLTE(I)                                                  ,34
      ID=LPLF(I)                                                        ,35
 350  CALL POLXN(CLINE,B(ID),2,NLTEP(I),1.E-4,1,RUB(1,I),LQ,MX(I))      ,36   ,37
```

```
      IF (IC-2)400,400,500
C
 400  BETA=BETA* .01745329252                                          ,38
      SN=SIN(BETA)                                                     ,39
      CS=COS(BETA)                                                     ,40
      CLINE(1)=SN                                                      ,41
      CLINE(2)=CS                                                      ,42
      IF (MX(IC))420,420,430                                           ,43
 420  MU=6                                                             ,44
      GO TO 9000                                                       ,45
 430  CLINE(3)=-SN*RUB(1,IC)-CS*RUB(2,IC)                              ,46
      GO TO 700                                                        ,47
C
 500  DO 530 I=1,2                                                     ,48
      IF (MX(I))520,520,530                                            ,49  ,50
 520  MU=7                                                             ,51
      GO TO 9000                                                       ,52
 530  CONTINUE                                                         ,53
C
      CLINE(1)=RUB(2,1)-RUB(2,2)                                       ,54  ,55
      CLINE(2)=RUB(1,2)-RUB                                            ,56
      CLINE(3)=RUB*RUB(2,2)-RUB(2,1)*RUB(1,2)                          ,57
      IF (CLINE(1) .NE. 0. .OR. CLINE(2) .NE. 0.) GO TO 700            ,58
C     AIRFOIL IS A POINT
      DO 610 J=1,2                                                     ,59  ,60  ,61
      DO 610 I=1,2                                                     ,62  ,63
 610  PT(I,J,IA)=RUB(I,1)                                              ,64  ,65
      CHDL=0.                                                          ,66  ,67  ,68
      GO TO 820                                                        ,69
C
C     CLINE CONTAINS COEFF. OF LINE (IN XY PLANE) WHICH
C     IS THE TRACE OF THE AIRFOIL.
C     FIND INTERSECTION WITH LE,TE AND CHORD LENGTH.
 700  DO 800 I=1,2                                                     ,70
      ID=LPLF(I)                                                       ,71  ,72
      CALL POLXN(CLINE,B(ID),2,NLTEP(I),1.E-4,1,PT(1,I,IA),LQ,MX(I))   ,73
      IF (MX(I)-1) 750,800,750                                         ,74
 750  MU=8                                                             ,75
      GO TO 9000                                                       ,76
 800  CONTINUE                                                         ,77
C
      CHDL=DISPTA(PT(1,1,IA),PT(1,2,IA),2)                             ,78  ,79
C
 820  WRITE (LO,850)((PT(I,J,IA),I=1,2),J=1,2),CHDL                    ,80
                                                                       ,81  ,82  ,83  ,84  ,85  ,86
 850  FORMAT(43H LEADING AND TRAILING EDGE POINTS (X,Y) ARE/           ,87  ,88  ,89
     .2(10X,2F10.4)/15H CHORD LENGTH =F10.4)
C
C     READY TO READ, SCALE, AND STORE AIRFOILS
 1000 DO 2000 IB=1,2
C
      IF (.NOT. LOGUL(IB)) GO TO 2000                                  ,90  ,91
 1020 N=NAFUL(IB)                                                      ,92  ,93  ,94
      IF (N)1040,1040,1100                                             ,95
                                                                       ,96
```

```
 1040 IF (IA-1)1060,1060,1080                                          ,87
 1060 MU=9                                                             ,88
      GO TO 9000                                                       ,89
 1080 NB=NAME(2,IB,IA-1)                                               ,90
      GO TO 1120                                                       ,91
 1100 NB=N+N
C                                                                      ,92
 1120 NAME(2,IB,IA)=NB
C     RESERVE STORAGE FOR AIRFOIL POINTS                               ,93
      IF (IRSERV(NB,8,NAME(1,IB,IA)))1140,1140,1200                    ,94
 1140 MU=10                                                            ,95
      GO TO 9000
C                                                                      ,96
 1200 LA=NAME(1,IB,IA)                                                 ,97
      LB=NAME(1,IB,IA-1)                                               ,98
      NA=N+N
C                                                                      ,99
      CALL WING1B(LI,LO,HUL(IB),CHDL,NB,B(LB),NA,B(LA),IU)             ,100
 1250 IF (IU)1280,2000,1260                                            ,101
 1260 MU=11                                                            ,102
      GO TO 9000
C     AIRFOIL IS A POINT                                               ,103
 1280 NAME(2,IB,IA)=2                                                  ,104
      IF (CHDL)1300,2000,1300
C     SET TE POINT TO LE POINT                                         ,105
 1300 DO 1310 I=1,2                                                    ,106
 1310 PT(I,2,IA)=PT(I,1,IA)
C                                                                      ,107  ,108
 2000 CONTINUE
C                                                                      ,109  ,110
 8000 CONTINUE                                                         ,111  ,112
 9000 RETURN                                                           ,113
      END                                                              ,114
```

```
      SUBROUTINE WING1P(LI,LO,UL,CHORD,NB,B,NA,A,NU)
      DIMENSION B(1),A(1)
C     USED BY WING1A TO READ,WRITE,SCALE AIRFOILS
C
      IF (NA-1)400,400,200                                          ,1
 200  READ (LI,210)(A(I),I=1,NA)                                    ,2    ,3    ,4    ,5    ,6
 210  FORMAT(6F10.0)
      N=NA/2                                                        ,7
      WRITE (LO,220)N,UL                                            ,8    ,9    ,10
 220  FORMAT(  10HOTHE GIVEN I4,A6,19H AIRFOIL POINTS ARE/
     ,5X 5HXWING 17X 5HYWING)
      WRITE (LO,230)(A(I),I=1,NA)                                   ,11   ,12   ,13   ,14   ,15
 230  FORMAT(1H F11.4,F22.4)
      GO TO 600
C                                                                   ,16
 400  IF (NB-3)410,410,420                                          ,17
 410  NU=1                                                          ,18
      GO TO 9000
C     USE PREVIOUS AIRFOIL                                          ,19
 420  NA=NB                                                         ,20
      DO 430 I=1,NA                                                 ,21
 430  A(I)=B(I)                                                     ,22   ,23
      WRITE (LO,440) UL                                             ,24   ,25   ,26
 440  FORMAT(10HOTHE GIVEN A6,37H AIRFOIL IS THE SAME AS PRECEDING ONE)
C
 600  IF (NA-3)1000,1000,700
C     FIND CHORD LENGTH OF AIRFOIL                                  ,27
 700  CHDA=ABS(A(NA-1)-A)                                           ,28
      IF (CHDA)900,900,800                                          ,29
 800  IF (CHORD)900,900,1100                                        ,30
 900  NA=2
C     AIRFOIL IS A SINGLE POINT                                     ,31
 1000 NU=-1                                                         ,32
      WRITE (LO,1010)                                               ,33   ,34
 1010 FORMAT(23HOTHE AIRFOIL IS A POINT)
      GO TO 9000
C                                                                   ,35
 1100 T=CHORD/CHDA
C                                                                   ,36
      DO 1130 I=3,NA,2                                              ,37
      IF (A(I)-A(I-2))1110,1110,1130                                ,38
 1110 WRITE (LO,1120)                                               ,39   ,40
 1120 FORMAT(//49H0POINTS NOT INCREASING IN X.  JOB MAY BLOW LATER. 60X.
     1 7HWARNING///)
      GO TO 1200                                                    ,41
 1130 CONTINUE
C
C
C     SCALE A                                                       ,42   ,43
 1200 DO 1210 I=1,NA                                                ,44
 1210 A(I)=T*A(I)                                                   ,45   ,46
      NU=0
C                                                                   ,47
```

```
EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

 2000 WRITE (LO,2010)                                                    ,48   ,49
 2010 FORMAT(52H0AIRFOIL POINTS AFTER SCALING TO FIT ON PLANFORM ARE/
     ,5X 14HXWING (SCALED) 10X 1H7)
      WRITE (LO,230)(A(I),I=1,NA)
C                                                                        ,50   ,51   ,52   ,53   ,54
 9000 RETURN                                                             ,55
      END                                                                ,56
```

```
      SUBROUTINE WING2(B,LC,LGDEF,LOK,LCGUL,NAF,NPL,LAFN,KOD,PLT,
     1 NAME,PCT,CHD,EP,TITLE,DAYT,NU)
      DIMENSION NAME(2,2,1),PLT(2,2,1),PCT(1),LOK(3,2),B(1),KOD(2),NU(3)
     1,MPL(2),UL(2),TITLE(12),DAYT(2)
      LOGICAL LGDEF(3,6),LOGUL(2)
      DATA UL(1)/12H UPPER LOWER/
      DATA WING/5HWING2/
C
C     USED BY WINGA TO COMPUTE A 3D POINT ON EACH PERCENT LINE
C     AT EACH AIRFOIL AND THEN ENRICH EACH PERCENT LINE.
C
 10   NAP=3*NAF*NPL                                                    ,1
      NU=0
C                                                                     ,2
 50   DO 5000 KA=1,2
C                                                                     ,3
      IF (.NOT. LOGUL(KA)) GO TO 5000                                  ,4    ,5    ,6
      LEFT=IPACK(B)
C     RESERVE SPACE FOR PERCENT LINE ARRAY                             ,7
      IF(IRSERV(NAP,B,LOK(1,KA)))100,100,200                           ,8
 100  MU=1                                                             ,9
      GO TO 9000
C                                                                     ,10
 200  IA=LOK(1,KA)-3
C                                                                     ,11
 300  DO 4000 KB=1,NPL
C                                                                     ,12
      P=PCT(KB)/100.                                                   ,13
      Q=1.-P
C                                                                     ,14
 500  DO 3000 KD=1,NAF
C                                                                     ,15
      IA=IA+3                                                          ,16
      LAFP=NAME(1,KA,KD)                                               ,17
      NAFP=NAME(2,KA,KD)/2                                             ,18
      MAFP=LAFP+2*NAFP-2
C                                                                     ,19
      IF (NAFP-1)600,700,800
C                                                                     ,20
 600  MU=2                                                             ,21
      GO TO 9000
C                                                                     ,22
 700  B(IA)=PLT(1,1,KD)                                                ,23
      B(IA+1)=PLT(2,1,KD)                                              ,24
      B(IA+2)=B(LAFP+1)                                                ,25
      GO TO 3000
C                                                                     ,26
 800  B(IA)=Q*PLT(1,1,KD)+P*PLT(1,2,KD)                                ,27
      B(IA+1)=Q*PLT(2,1,KD)+P*PLT(2,2,KD)                              ,28
      U=Q*B(LAFP)+P*B(MAFP)                                            ,29
      CALL BITURP(B(LAFP),B(LAFP+1),2,NAFP,KOD,U,B(IA+2),NU)           ,30
      MU=3
C                                                                     ,31
```

```
      IF(NU)9000,3000,9000                                              ,32
C
 3000 CONTINUE                                                          ,33    ,34
C
 4000 CONTINUE                                                          ,35    ,36
C
      LEFT=IPACK(B)                                                     ,37
      KDEL=LOK(1,KA)                                                    ,38
      IF(IDLETE(B(KDEL)))4910,4920,4910                                 ,39
 4910 MU=5                                                              ,40
      GO TO 9000
C                                                                       ,41
 4920 IF(IRSERV(LEFT+NAP,B,LOK(1,KA)))4930,4930,4940                    ,42
 4930 MU=6                                                              ,43
      GO TO 9000
C                                                                       ,44
 4940 IA=LOK(1,KA)                                                      ,45
      CALL WING2P(LO,B(IA),NPL,NAF,TITLE,UL(KA),DAYT,PCT)               ,46
      CALL RICHNA(B(IA),LEFT/3,NPL,NAF,PCT,EP,CHD,KOD(2),IPL,NU)        ,47
      MPL(KA)=3*(IPL+NPL)
C                                                                       ,48
      IF(NU)4950,4960,4950                                              ,49
 4950 MU=7                                                              ,50
      GO TO 9000                                                        ,51
 4960 KDEL=LOK(1,KA)                                                    ,52
      IF(IRLEAS(B(KDEL),MPL(KA)))4970,4980,4970                         ,53
 4970 MU=8                                                              ,54
      GO TO 9000                                                        ,55
 4980 LOK(2,KA)=NPL                                                     ,56
      LOK(3,KA)=MPL(KA)                                                 ,57
      LGDEF(2,KA)= .TRUE.
C                                                                       ,58
 5000 CONTINUE
C
C     RELEASE STORAGE IN B WHICH CONTAINS THE AIRFOIL POINTS AND NAMES  ,59    ,60
      DO 8100 JA=1,NAF                                                  ,61
      DO 8100 JC=1,2                                                    ,62
      JB=NAME(1,JC,JA)                                                  ,63
 8100 JD=IDLETE(B(JB))                                                  ,64    ,65    ,66
      JD=IDLETE(B(LAFN))                                                ,67
      GO TO 9900
C                                                                       ,68
 9000 I=MERR(NU,NU+1,MU,WING)                                           ,69
      NU=MU                                                             ,70
 9900 RETURN                                                            ,71
      END                                                               ,72
```

```
      SUBROUTINE WING2P(LO,W,NPCT,NPT,TITLE,UPLO,DAYT,PCT)
      DIMENSION W(3,NPT,NPCT),TITLE(12),DAYT(2),PCT(NPCT)
C
C     USED BY WING2 TO WRITE PERCENT LINES (BEFORE ENRICHING)
C     W = ARRAY OF NPCT PERCENT LINES.  EACH LINE CONTAINS
C         NPT POINTS (X,Y,Z)
C     UPLO =6H UPPER OR 6H LOWER
C     PCT = PERCENT VALUES ASSOC. WITH EACH PERCENT LINE
C
      DATA LINPAG/45/
C
      LINREM=0
C
      DO 2000 K=1,NPCT                                                  ,1
      NAF=0                                                             ,2
      NP=NPT                                                            ,3
      IF(LINREM .GT. 4) GO TO 200                                       ,4
C
  100 WRITE(LO,110) TITLE,UPLO,DAYT                                     ,5    ,6    ,7
  110 FORMAT(1H1 12A6/10H POINTS ON A6,                                 ,8    ,9    ,10
     ,33H WING SURFACE PERCENT CHORD LINES 5X 2A6)
      LINREM=LINPAG-6
C
  200 WRITE(LO,210) K,PCT(K)                                            ,11
  210 FORMAT(/24HOWING PERCENT CHORD LINE I3,1H, F9.4,8H PERCENT        ,12   ,13   ,14
     ,/7H AF NO. 7X 1HX 10X 1HY 10X 1HZ)
      N=MINO(NP,LINREM)                                                 ,15
      LINREM=LINREM-N-4
C
      DO 300 J=1,N                                                      ,16
      NAF=NAF+1                                                         ,17
  300 WRITE(LO,310)NAF,(W(I,NAF,K),I=1,3)                               ,18
                                                                        ,19   ,20   ,21   ,22   ,23   ,24
  310 FORMAT(1H I3,4X 3F11.4)                                           ,25
C
      NP=NP-N                                                           ,26
      IF(NP)2000,2000,100                                               ,27
 2000 CONTINUE                                                          ,28   ,29
 9000 RETURN                                                            ,30
      END                                                               ,31
```

```
      SUBROUTINE WINGA(B,LI,LO,DAT,LGDEF,LOK,LWPF,TITLE,NU)
      DIMENSION B(1),DAT(2),LOK(3,9),KOD(2),NU(3),TITLE(12)
      LOGICAL LGDEF(3,6),LOGUL(2)
C
C     USED BY GEOMD
C
 100  CALL WING1(B,LI,LO,DAT,LWPF,LOGUL,NAF,NPL,LAFN,LPCT,
     1 KOD,CHD,EP,TITLE,NU)                                          ,1
      LGDEF(1,1)=LOGUL(1)                                            ,2
      LGDEF(1,2)=LOGUL(2)                                            ,3
      LGDEF(1,4)= .TRUE.
C                                                                    ,4
      IF(NU)1000,200,1000                                            ,5
 200  LGDEF(2,4)= .TRUE.                                             ,6
      LOK(2,7)=NAF                                                   ,7
      LOK(3,7)=4*NAF                                                 ,8
      LEFT=IPACK(B)
C
C     WE MUST RE-DEFINE LWPF,LAFN,LPCT.  IF THIS IS NOT DONE,THE
C     COMPILER WILL STUFF ADDRESSES AT STATEMENT 300 BASED ON
C     LOCATIONS AT STATEMENT 100.  BUT THE PACKING,
C     LEFT=IPACK(B),MAY HAVE CHANGED LOCATIONS.
C                                                                    ,9
      LWPF=LWPF                                                      ,10
      LAFN=LAFN                                                      ,11
      LPCT=LPCT
C                                                                    ,12
  300 CALL WING2(B,LO,LGDEF,LOK,LOGUL,NAF,NPL,LAFN,
     1 KOD,B(LWPF),B(LAFN),B(LPCT),CHD,EP,TITLE,DAT,NU)
C                                                                    ,13
 1000 RETURN                                                         ,14
      END                                                            ,15
```

```
      SUBROUTINE WLDM (NM,NROW,XP,ZP,REAREA,AREA,XBAR,ZBAR,ALPHAM
     1,THETAM,CPM,SCL,SCD,CL,CD,CM)

C     COMPUTES COEFFICIENT OF LIFT, DRAG, AND MOMENT ON WING
C     SPANWISE DISTRIBUTION OF LIFT AND DRAG COEFFICIENTS

      DIMENSION AREA(1),XBAR(1),ZBAR(1),ALPHAM(1),THETAM(1),CPM(1)
     1,SCL(1),SCD(1)


      NCOL=NM/NROW                                                      ,1
      WDRAG=0.                                                          ,2
      WLIFT=0.                                                          ,3
      WMOM=0.                                                           ,4
      J=0
                                                                        ,5
      DO 200 K=1,NCOL                                                   ,6
      DRAG=0.                                                           ,7
      XLIFT=0.
                                                                        ,8
      DO 100 I=1,NROW                                                   ,9
      J=J+1                                                             ,10
      F=CPM(J)*AREA(J)                                                  ,11
      XL=-F*COS(THETAM(J))                                              ,12
      XD=F*ALPHAM(J)                                                    ,13
      WMOM=WMOM-XL*(XBAR(J)-XP)+XD*(ZBAR(J)-ZP)                         ,14
      XLIFT=XLIFT+XL                                                    ,15
      DRAG=DRAG+XD                                                      ,16
  100 CONTINUE
                                                                        ,17   ,18
      SCD(K)=DRAG/REAREA                                                ,19
      SCL(K)=XLIFT/REAREA                                               ,20
      WDRAG=WDRAG+DRAG                                                  ,21
      WLIFT=WLIFT+XLIFT                                                 ,22
  200 CONTINUE
                                                                        ,23   ,24
      CL=WLIFT/REAREA                                                   ,25
      CD=WDRAG/REAREA                                                   ,26
      CM=WMOM/REAREA
                                                                        ,27
      RETURN                                                            ,28
      END                                                               ,29
```

PRECEDING PAGE BLANK NOT FILMED.

6. REFERENCES

1. Aerodynamics Research Staff: Summary Description of a Method of Optimizing Camber Surfaces for Wing-Body Combinations at Supersonic Speeds. Boeing Document D6-10740, September 1965.
2. Woodward, F. A.; and Larsen, J. W.: A Method of Optimizing Camber Surfaces for Wing-Body Combinations at Supersonic Speeds—Theory and Application. Boeing Document D6-10741—Part I, September 1965.
3. IBM 7090-7094 Programming Systems—IBJOB Processor—Overlay Feature of IBLDR. IBM Form C28—5331, 1963.

PRECEDING PAGE BLANK NOT FILMED.