NASA TM - 79144

NASA-TM-79144 19790017848

NASA Technical Memorandum 79144

COMPUTER PROGRAMS FOR CALCULATING TWO-DIMENSIONAL POTENTIAL FLOW THROUGH DEFLECTED NOZZLES

J. Dennis Hawk and Norbert O. Stockman
Lewis Research Center
Cleveland, Ohio

May 1979

BEST

AVAILABLE

COPY

1 Report No NASA TM-79144	2 Government Accession No	3 Recipient's Catalog No

4 Title and Subtitle COMPUTER PROGRAMS FOR CALCULATING TWO-DIMENSIONAL POTENTIAL FLOW THROUGH DEFLECTED NOZZLES	5 Report Date May 1979 6 Performing Organization Code
7 Author(s) J. Dennis Hawk and Norbert O. Stockman	8 Performing Organization Report No E-9999 10 Work Unit No
9 Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135	11 Contract or Grant No 13 Type of Report and Period Covered
12 Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546	14 Sponsoring Agency Code

15 Supplementary Notes

16 Abstract

Computer programs to calculate the incompressible potential flow, corrected for compressibility, in two-dimensional nozzles at arbitrary operating conditions are presented. Included are a statement of the problem to be solved, a description of each of the computer programs, and sufficient documentation, including a test case, to enable a user to run the program.

17 Key Words (Suggested by Author(s)) Two dimensional Potential flow Nozzles Computer programs	18 Distribution Statement Unclassified - unlimited STAR Category 02

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

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

CONTENTS

N79-26019

INTRODUCTION

Many proposed V/STOL aircraft require propulsion systems that deflect the thrust during takeoff, landing, and hovering. This deflection can be accomplished by use of flow turning nozzles downstream of the fans. These nozzles must operate efficiently and effectively over wide ranges of internal flow conditions. An important tool in the design of an efficient nozzle is the capability to theoretically analyze the nozzle flow.

Many of the proposed V/STOL nozzles are three-dimensional (fig. 1). Since the capability for a full three-dimensional nozzle analysis does not exist at this time, simpler but approximate methods for analyzing these nozzles were developed. Even though these nozzles are three-dimensional overall, there are sections of the nozzles for which a two-dimensional analysis is an informative approximation. The analytical procedure described herein is a two-dimensional analysis.

E-9999

The procedure developed utilizes three computer programs and is similar to the procedure for calculating axisymmetric inlet flows (ref. 1) and two-dimensional inlet flows (ref. 2). The chief program is the higher order Douglas two-dimensional potential flow program (ref. 3) called 24Y at the Lewis Research Center, which calculates the incompressible potential flow about arbitrary two-dimensional bodies. The other programs, original at Lewis, are called SCIRCL and NOZZLEC. Program SCIRCL generates geometric inputs for 24Y from various specified analytical shapes or sets of coordinate points for the nozzle components. Program NOZZLEC takes the basic solutions output by 24Y and combines them into solutions of interest, and, if desired, applies a compressibility correction. Figure 2 is a schematic representation of these solution steps.

This paper consists of a statement of the problem to be solved, description of each program, sample results, and sufficient documentation, including a test case, to enable the user to run the programs.

SYMBOLS

A, B, C, D	combination coefficients
M	Mach number
V	velocity
$\dot{W}$	mass flow
α	incidence angle

Subscripts:

c	control station
com	compressible
i	incompressible
S1	passage between body 1 and body 2
S2	passage between body 2 and body 3
∞	free stream value

PROBLEM AND SOLUTION DESCRIPTION

Statement of the Problem

Geometric representation. - A three-dimensional nozzle can be approximated two-dimensionally by using the profiles at the plane of symmetry of the three-dimensional nozzle (fig. 3). The basic assumption for two-dimensionality is that all derivatives in the z-direction vanish, that is, $\partial(\ )/\partial z = 0$.

Care should be used when attempting to use a centerbody because in two dimensions the centerbody is actually a flow splitter. In most cases, adequate results can be obtained by omitting the centerbody.

Nozzles. - The basic problem to be solved is to calculate either the compressible or incompressible flow in an arbitrary two-dimensional nozzle at any combination nozzle mass flow rates $\dot{W}_c$, $\dot{W}_{S1}$, or $\dot{W}_{S2}$. Figure 4 shows three possible combinations of nozzle geometries and weight flow specifications to be analyzed. Figure 4(a) shows the single passage nozzle where weight flow rate, $\dot{W}_c$ is specified. Figure 4(b) shows a nozzle with dual passages where the two passage weight flow rates $\dot{W}_{S1}$ and $\dot{W}_{S2}$ are specified. Figure 4(c) shows the same nozzle as figure 4(b). However, in this case the total nozzle and one passage weight flow rate, $\dot{W}_c$ and $\dot{W}_{S1}$ are specified.

Inlets and nozzles. - While most of the nozzles problems are internal flow problems, occasionally external flows about inlets and nozzles are desired (fig. 5). If this is the case the programs can accomplish this by inputting values of free stream velocity V_∞ and nacelle incidence angle, α, along with the required weight flows described previously. A detailed description of inlet calculation methods is given in references 1 and 2.

The two-dimensional flow problem is solved in several steps (programs).

1. Geometric representation (Program SCIRCL)

2. Incompressible potential flow ba- utions (Program 24Y)
3. Combined solutions with compres ility correction (Program NOZZLEC).

Each step and its corresponding progra scribed in the following section.

Description of Solutio and Programs

Geometric representation - Progra CL - The nozzle is assumed to be two dimensional. The profiles are b into segments at convenient tangential points (fig. 6). These profiles are rred to as bodies in the program nomenclature. Each segment may be defi by an analytic expression or by a set of points. The nozzle walls and outer aces (if any) must be extended far downstream (fig. 6) to facilitate obtainin rate potential flow solutions in the nozzle in the region of interest. Th etry program SCIRCL prepares coordinate-point input for efficient use o potential flow program and also prints out information such as curvature, angles, etc., which is useful in preliminary screening of proposed body s

In addition to the surface points, set oints in the vertical plane, called rakes, are needed at axial locations whe locity profiles or streamlines are desired. At least one set of rake points equired, corresponding to the axial location where the weight flow is specified, as discussed with regard to figure 4. Such rakes are called control stations.

Program SCIRCL generates the coordinates of the rake points for 24Y. Program SCIRCL will also produce a Ca plot of the nozzle geometry and rake points.

Incompressible potential flow basic ations - Program 24Y. - Program 24Y is the Douglas incompressible pote al flow computer program for single or multiple two-dimensional bodies. B ly, the program utilizes a distribution of sources and sinks of initially unk strength to represent the body profile. The continuous distribution is app ted by representing the body as a finite number of discrete flat elements onstant source strength and characterized by the mid point of the elemen ed the control point). This approximation results in a set of parabolic alge c equations for source strength at the control points that are solved by matrix me ods. Velocities at the control points and at specified off-body points (rake points re then calculated from the source/sink distribution.

In the two-passage case, the progran used to obtain five basic solutions which are used in linear combination in ord to satisfy the prescribed operating conditions (fig. 4). The first basic solution is axial flow, the second is a 90° cross flow, the third, fourth, and fifth solutions are vorticity solutions about bodies 1, 2, and 3, respectively. In the single passage case, only four solutions are generated. The first two are the same as the two-passage case while the third and fourth are vorticity solutions about the two bodies.

Combined solution - Program NOZZLEC. - This program combines the basic solutions $\overline{V}_j$, j = 1, 2, 3, 4, 5 from 24Y into any number of solutions of interest. A solution of interest is one having specific values of free stream velocity, V_{∞}, angle of attack of the free stream, α, and weight flow, $\dot{W}$, through the control stations described previously. Optional inputs for $\dot{W}$ are average axial velocity at the control stations (V_c, V_{S1}, or V_{S2}) or average Mach numbers (M_c, M_{S1}, or M_{S2}). If either $\dot{W}$ or M_c are specified, they are converted to velocities for use in the combination solutions. Temperature and pressure must also be specified if other than standard conditions are used.

The method of combination of the basic solutions is shown in figure 7. A compressibility correction, if desired, is then applied to the velocities. The compressibility correction used is

$$V_{com} = V_i\left(\frac{\rho_t}{\rho_c}\right)^{V_i/\overline{V}_c}$$

where the terms on the right-hand side are obtained from the incompressible solution. This correction requires no alteration of the inlet geometry and can handle locally sonic or supersonic flows. Flow properties (Mach number, pressure ratio, etc.) are calculated for either the compressible or incompressible cases depending on the version desired. The compressibility correction is actuated by setting ICOMP1 to 0. The incompressible version is generated by setting ICOMP1 to 1. If two passages are specified, two of the rakes mentioned under SCIRCL are used as control stations by NOZZLEC A control station is the rake where the average axial velocity of the combined solution is specified. The possible choices of control stations for two passage cases are shown in figures 4(b) and (c). If only one passage is specified (fig. 4(a)), then any rake can be used as a control station. However, it should be noted that the compressible solution is most accurate in the vicinity of the control station since the compressibility correction does not exactly satisfy continuity.

When the velocity on the surface becomes locally supersonic, the agreement between theory and experiment is generally not as good as when the flow remains subsonic. To improve the agreement, an optional supersonic correction has been incorporated into the program. The supersonic correction is actuated as specifying NX as 1 on input.

Sample Results

To illustrate the results of the present calculation procedure, the graphic output for the Test Case is presented next.

The graphic output of SCIRCL is illustrated in figure 8 and consists of a plot of the nozzle geometry and the computational point spacing, and the location of the rakes and their point spacing. In addition to its reference value the plot is useful for troubleshooting for mistakes in the geometry input and for validating the satisfactory distribution of body points and rake points.

The graphic output of NOZZLEC is illustrated in figures 9 to 11. Figure 9 shows the pressure distribution on the surfaces of body 1. Similar plots can be obtained for bodies 2 and 3.

Figure 10 shows the surface Mach number distribution. The comments made above for the pressure plots apply also to Mach number plots.

Figure 11 shows a plot of the flow field vectors. Whenever rake points are specified (see fig. 8) velocity vectors are calculated and can be plotted.

INPUT AND OUTPUT FORMAT

SCIRCL Input

Card	Format	Column	FORTRAN name	Description
	9A6	1-54	ARE	= title for area plots. "CROSS-SECTIONAL AREA, SQ. IN."
	9A6	1-54	EX	= title for x-axis. "AXIAL POSITION, IN."
	9A6	1-54	CURVO	= title for curvature plots. "CURVATURE ON SHROUD"

Card	Format	Column	FORTRAN name	Description
	9A6	1-54	SURFAC	= title for x-axis with surface distance plots. "DISTANCE ALONG SURFACE FROM DOWNSTREAM END OF SHROUD, IN."

These first four cards above will be unchanged for all runs and can be made a part of the execution setup deck, or replaced with data statements.

Card	Format	Column	FORTRAN name	Description
1	9A6	1-54	TITLE	Description of Case

FOR CALCOMP PLOTTING OF INLET PICTURES

Card	Format	Column	FORTRAN name	Description
2	8F10.2	1-10	XX	Length, in plot-inches, of X-axis required
		11-20	XMIN	Value, in data-inches, of far left X-point.
		21-30	EXEP	Data-inch per plot-inch along X-axis.
		31-40	YY	Length, in plot-inches, of Y-axis required.
		41-50	YMIN	Value, in data-inches, of bottom Y-point
		51-60	ORD	Data-inch per plot-inch along Y-axis (usually equal to EXEP).

Card	Format	Column	FORTRAN name	Description
		61-70	ELREF	The X-values in area output data are nondimensionalized by ELREF. Default value is 1.
		71-80	AREF	The areas in area output data are nondimensionalized by AREF. Default value is 1.
3	4X, F10.2, 1X 3I1	5-14	ALPHER	If only one body is input, ALPHER is the angle of attack (used by 24Y)
		16	IPARA	Element geometry flag used by 24Y = 0 Linear elements = 1 Parabolic elements
		17	IFRST	First-order terms flag = 0 No first-order terms = 1 First derivative terms = 2 Curvature terms = 3 Both first-order terms
		18	ISND	Second-order terms flag = No second-order terms = 1 Second derivative terms = 2 Curvature squared terms = 3 Both second-order terms

Card	Format	Column	FORTRAN name	Description
		19	IFLLL	= 0 No combination solution will be calculated by 24Y = 1 A combination solution will be calculated by 24Y
4	2A6, I4, 2I1, 2I2, 10X, I1, 9X, 3I1, 2I2	1-6	IDENT	Six-character tag for case I D.
		7-12	PROG	24Y (right justified)
		13-16	NO6	0
	(Flag 'A')	17	LPNCHO	1, Do not save output for 24Y on Unit 17.
	Flag B	18	IPLOTA	Plot area against X-position
	Flag C	19-20	IPLOTC	-1, Plot curvature versus X, +1, Plot curvature versus S
	Flag D	21-22	IREAD	0 (Obsolete)
ALL FLAGS are 'on' when =1, unless otherwise noted (Either E or J or neither can be on but not both).				
	Flag J	33	IAB	Redo geometry from point
	Flag E	43	IREDON(1)	Redo entire geometry via direct interpolation

Card	Format	Column	FORTRAN name	Description
	Flag F	44	IREDON(2)	LPNCHO for any redo
	Flag G	45	IREDON(3)	IPLOTA for any redo
	Flag H	46-47	IREDON(4)	IPLOTC for any redo
	Flag I	48-49	IREDON(5)	IREAD for any redo
Skip card 5 if J=0				
5	4F12.5	1-12	XAA	X position of starting point for partial redo
		13-24	YAA	Y position of starting point for partial redo.
		25-36	XBB	X position of ending point for partial redo.
		37-48	YBB	Y position of ending point for partial redo.
6	4F10.2	1-10	ANBDYS	Number of bodies
		11-20	DELS	Spacing between points in region of interest.
		21-30	DELSMX	Maximum spacing far from region of interest
		31-40	XRI	Axial distance at which surface distance equals zero.
7	I4	1-4	NRAKE	Number of axial locations at which data across the passage is desired. (Cannot be greater than 25).
8	3F8.5, I3, 4F8.5	1-8	XRAK	Axial location of rake.
		9-16	YLO	Y value of first point (lowest point) on rake at XRAK.

(Note: There is one card for each rake)

Card	Format	Column	FORTRAN name	Description
		17-24	YHI	Y value of last point (highest point) on rake at XRAK.
		25-27	NY	Number of points in rake at XRAK. Restriction Σ NY $\leq$ 200. Rake points are equally spaced, ΔY, between YHI and YLO where $\Delta Y = \frac{YHI - YLO}{(NY - 1)}$
		28-35	XTRAN	Value of axial translation of rake
		36-43	YTRAN	Value of vertical translation of rake
		44-51	XSCALE	Value of axial scaling of rake
		52-59	YSCALE	Value of vertical scaling of rake
9	4F10.2, 5F8.2	1-10	TYPBDY	Body number. However, if there is symmetry, then any body can be input as a mirror image of any other body [illegible] can be accomplished [illegible] setting TYPBDY [illegible] where M is the n[illegible] the body to be crea[illegible] N is the number of [illegible] body to be copied [illegible] is set to the Y value of [illegible] ure about which body N is to be mirrored No other

Card	Format	Column	FORTRAN name	Description
				input is required for this body except for ANLF
		11-20	ANSEG	= Number of segments for the particular body, except as stated in TYPBDY.
		21-30	DELNEW	= -1., Delta S spacing is set to original value of DELS. = 0., Delta S is set to value of DELS from previous body. = + number, Delta S is set to value of input DELNEW.
		31-40	ANLF	= 0 Body is a lifting body, i.e., in 24Y a vorticity solution about this body will be calculated. = 1, Body is a non-lifting body, i.e., no vorticity solution will be calculated

Note: All lifting bodies must be input prior to any non-lifting bodies.

Card	Format	Column	FORTRAN name	Description
		41-48	XTRAN	Value of axial translation of this body.
		49-56	YTRAN	Value of vertical translation of this body
		57-64	XSCALE	Axial scaling factor

Card	Format	Column	FORTRAN name	Description
		65-72	YSCALE	Vertical scaling factor
		73-80	XTMAX	Maximum value of X for which scaling is to be applied.
10	3F10.2	1-10	ENREED	Code indicating type of curve to be fitted through given points. = 0 , for bisuperellipses. See Table I and figure 12 for available options. Input 1, 2, 3, 4, 5, or 6 (XIN, YIN) points as directed. = 1000. Same as = 0 but with finer point spacing near one end of segment (two such segments required). Usually used to give finer spacing at the highlight. The superellipse going into the highlight and the one coming out should have this flag For bisuperellipses where the '1000.' option is to be used, the rate at which the point spacing, ds, changes near one end dS_i dS_{i-1} - (Rate) (dS_{i-1}) can be specified on input.

Card	Format	Column	FORTRAN name	Description
				The rate (program name = PACE) is entered as the fractional part of ENREED for each segment. For example, if ENREED were input as 1000.06, the spacing for consecutive points would be evaluated as follows: $DS_i = DS_{i-1} - (0.06)DS_{i-1}$ if segment is to go from large-to-small spacing, or: $DS_i = DS_{i-1} + 1.5$ $(0.06)\ DS_{i-1}$ if segment is going from small-to-large spacing.
				If PACE is entered as zero (i.e., ENREED = 1000.), the default value, 0.05, is used.
				(PACE $\leq$ 0.133)
				*The first '1000' superellipse ON A BODY reduces the point spacing as far as possible, down to a limit of 2 percent of the ds value at the beginning of the segment.
				*All subsequent '1000' superellipses input will increase ds as far as possible up to the input value of DELS.

Card	Format	Column	FORTRAN name	Description
				* Any number or types of segments may be input between the first and subsequent '1000' bisuperellipses, with the exception of a normal bisuperellipse (ENREED=0).
				= 1, is a straight line, input 2 coordinates (XIN(1), YIN(1), XIN(2), YIN(2)) (fig. 13(a)).
				The first and last straight lines on bodies 2 and 3 and the last straight line on body 1 will automatically have their spacing increased from approximately DELS near the region of interest to approximately DELSMX away from the region of interest. To get this type of spacing in the first straight line of body 1, ENREED must be specified as 10

Card	Format	Column	FORTRAN name	Description
				= 10., special straight line used for initial straight line on lower shroud. The straight line starts with large spacing (DEMSMX) and ends with small spacing (DELS), (fig. 13(b)).
				= -1., fits a lemniscate between a straight line and a point. Input is three coordinates (fig. 13(c)).
				= -3., fits a cubic between two straight lines. Input 4 coordinates (fig. 13(d)).
				= -4.0, generates a segment which is a mirrored image of all the points from (XIN(1), YIN(1)) to XIN(2), YIN(2)) about the line Y = YIN(3). See cards 11 and 12 for XIN and YIN formats.
				= 99., for direct interpolation option over one segment (see input instructions for card 12)

Card	Format	Column	FORTRAN name	Description
		11-20	REEDEN(1)	(See table I). Input exponent of x-term for bisuperellipse equation Blank for all other segment types.
		21-30	REEDEN(2)	(See table I). Input exponent for y-term of bisuperellipse.
11	6F12.5	1-72	XIN(I) I=1, 2, 3, 6, 4, 5	X-coordinate for specified points.
12	6F12.5	1-72	YIN(I) I=1, 2, 3, 6, 4, 5	Y-coordinate for specified points. Note. If ENREFD = 99. instead of cards 11 and 12, input the following cards.
11a	Namelist/ $BODYIN/		Z(I)	Z is a complex array containing the X value (in the real part) and Y value (imaginary part) of each given point along the segment. The namelist will normally be longer than one card The program will use the input points to generate new points with finer point spacing near regions of high curvature
12a	Namelist/ $AUXIN/		DONE	A logical variable which should be input as .TRUE

Card	Format	Column	FORTRAN name	Description
			BYPASS	= .TRUE. if no refinement of input points is required.
				Note: If ANSEG=0 and TYPBDY $\neq 0$, skip card No. 10, and substitute 11a for 11 and 12a for 12.

<u>Input Deck Structure</u>

Card	
1	
2	
3	
4	
5	(only if flag J > 0)
6	
7	
8 . . . 8	Number of '8' cards = NRAKES
9	Number of '9' cards = ANBDYS
10 11 12 . . . 10 11 12	Number of '10-11-12' groups for each '9' card = ANSEG *If ENREED = 99 on card 10, use 11a and 12a instead of 11 and 12 *If ANSEG = 0 and TYPBDY $\neq 0$ on card 9, skip 10 and substitute 11a and 12a for 11 and 12

Figure 14 shows an SCIRCL input form, reflecting the above instructions.

SCIRCL Output

Printed Output

Input file dump (a list of input cards)

Case number and title

Input card 3

Input card 4 (case I.D. and SCIRCL flags)

Input card 6 (number of bodies, DELS, DELSMX, and XRI)

Total number of points for all bodies should not exceed 500.

Total for any one segment of a body should not exceed 200.

An error message will indicate if these limits have been exceeded.

Total number of off-body points must not exceed 200.

Total number of rakes must not exceed 25.

Body 1 segment data, body 2 segment data, and body 3 segment data. For each segment:

ENREED (as input) and type of segment

Data depending on type of segment.

Straight Line

X	X(1)	X(2)
Y	Y(1)	Y(2)

Last point data*

Bisuperellipse

Exponents

P as read in	X	X(1)	X(2)	X(3)	X(6)	X(4)	X(5)
Q as read in	Y	Y(1)	Y(2)	Y(3)	Y(6)	Y(4)	Y(5)
P as used	A	XO					
P as used	B	YO	OMEGA				

where A and B = Semimajor and minor axes of transformed superellipse

XO and YO = Center of the transformed superellipse

OMEGA = The difference (in radians) between the slopes of the end point slope lines minus $\pi/2$ (i.e., OMEGA is a measure of the non-perpendicularity of the slope lines).

Number of iterations**

Iteration data***

Last data point*

'Magic triangle' messages refer to the triangle formed by extending the superellipses slope lines toward each other and drawing a chordline between input points number (X2, Y2) and (X4, Y4). Input points (X3, Y3) and (X6, Y6) must lie within this triangle, or, for certain special cases, may lie in a similar triangle on the opposite side of the chordline.

If input points (X3, Y3) and/or (X6, Y6) fall outside the magnic triangle, the following message (or messages) is/are printed:

"This point is below the magic triangle..."

"This point is outside the magic triangle..."

<u>Cubic</u>

X	X(1)	X(2)	X(3)	X(4)
Y	Y(1)	Y(2)	Y(3)	Y(4)
A	B	C	D	

where A, B, C, D are the coefficients of the cubic equation.

Number of iterations**

Iteration data***

Last point data*

<u>Lemniscate</u>

X	X(1)	X(2)	X(3)
Y	Y(1)	Y(2)	Y(3)
THETMX		CALC	ACALC

where THETMX CALC = angle between line 1 - 3 and line 1 - 2.

ACALC from equation $R^2 = 2(ACALC)^2 \sin 2\theta$ where

$$\theta = \text{THETMX} \quad \text{and} \quad R^2 = \left[XIN(3) - XIN(2)\right]^2 + \left[YIN(3) - YIN(2)\right]^2$$

Number of iterations**

Iteration data***

Last point data*

*Last point data. This is the coordinate point data for the last point of the segment. It is presented here because it is overwritten by the first point of the following segment and therefore does not appear in the point-by-point array below.

**Number of iterations is the number of iterations required to achieve a satisfactory point spacing for 24Y and have the calculated last point of the segment coincide with the input endpoint (to within a prescribed tolerance of 0.1 DELS). If this satisfactory spacing and end point matching is not achieved within 150 iterations, the following message is printed out above the number of iterations:

"This set of data exceeded 150 iterations. Calculations stopped XBRK YBRK.

X(1), Y(1), X(2), Y(2), X(3), Y(3), X(4), Y(4), X(5), Y(5)."

***Iteration data

DELS IN	Value of ds at end of previous segment
DELS	Value used to start final iteration for this segment
DELS OUT	Value of ds at end of this segment, to be passed on to the next segment
DSTEST	Distance from last calculated point to input segment end-point
FINAL PACE	The value of PACE at the conclusion (or termination) of the point-spacing iteration for bisuperellipses with ENREED 1000

"INPUT FOR THE NOZZLE COMBINATION PROGRAM"

For three body case

NT(1) = , NT(2) = , NT(3) = , NP =

For two body case

NT(1) = , NT(2) = , NP =

Body coordinates (a separate set for each body)

Point number

X-axial distance

Y-radial distance

KAPPA-curvature

DY/DX-slope

ALPHA-slope angle in degrees

S-surface distance measured from first point of each body

S-S(2) - surface distance measured from XRI

DELTAS-distance between points

Rake Information

XRAK - axial location of rake

YLO - Y value of first location on rake

YHI - Y value of last location on rake

NDY - number of Y points on the rake, as input

File output. - The file output (UNIT 17), written by SCIRCL, is used directly as input to 24Y. This file consists of the on-body point coordinates and rake points.

Graphic output. - Standard Calcomp can be produced for each geometry run as follows:

(1) For all cases: an X-Y 'picture' of the nozzle with SCIRCL-generated on-body points denoted by the '+' symbol and connected with straight lines; segment end points have a large circular symbol; off-body rake points are denoted by a small square symbol (see fig. 8).

(2) For cases with IPLOTC $\neq 0$: a graph of upper shroud body curvature against axial position (PILOTC = -1) or against distance along shroud (IPLOTC = +1).

24Y INPUT

Card	Format	Column	FORTRAN name	Description
1	3(I1, 2X), 1X 7 A4, 5X, 9(I1, 2X), 1X, I1	1	ID	Body number.
		4	ISV	Flag to control the saving of geometry data for future use again by the Two-Dimensional Program. = 0 Do not save data. = 1 Save the input geometry data for future use.
		7	ILIFT	Lift control flag. ILIFT = 1-ANLF (from SCIRCL) = 0 This is not a lifting body. = 1 This is a lifting body.
		11-38	TTILE	Body description
		44	IPARA	Element geometry flag = 0 Linear elements = 1 Parabolic elements

Card	Format	Column	FORTRAN name	Description
		47	IFIRST	First-order terms flag. = 0 No first-order terms = 1 First derivative term = 2 Curvature term = 3 Both first-order terms
		50	ISEC	Second-order terms flag. = 0 No second-order terms = 1 Second derivative term = 2 Curvature squared term = 3 Both second-order terms
		53	ITR	Geometry transformation flag. = 0 Transformation card will not be input. = 1 Geometry transformation card will be input. = 2 Ellipse generation. Ellipse generation card will be input. Transformation card will not be input. 3 Ellipse generation. Ellipse generation card will be input. Transformation card will be input.

Card	Format	Column	FORTRAN name	Description
		56	INORM	Geometry normalization flag. = 0 Geometry will not be normalized = 1 All of the geometry data (X and Y) will be divided by the chord length before use by the potential flow program.
		59	IBOD	Body disposition flag. This flag together with the IDOLD parameter controls the sequence of shapes that are to be presented to the potential flow analysis part of the program. With the use of these two flags and the ISV parameter it is possible to perform a variety of multi-element analysis problems with a minimum of input data. For normal useage when all the geometry data are input only the IBOD = 1 and =2 inputs are used. 1 New geometry is being input. The storage of geometry data for the potential flow solution will start with this body.

Card	Format	Column	FORTRAN name	Description
				= 2 New geometry is being input but this is not the first body. This body will be added to the sequence of body data already input.
				= 3 New geometry is being input but it is to be added to an old sequence of data
				= 4 All previously saved geometry will be used.
				= 5 The geometry for this body will be selected from the previously saved data (body IDOLD will be selected). This selected body will be added to the current string of bodies
				= 6 Previously saved geometry data will be used with the body number indicated by the IDOLD parameter removed from the solution
		62	IDOLD	Old body ID number. This parameter is used in conjunction with the IBOD parameter in selecting which previously saved shape is to be retrieved as the present body

Card	Format	Column	FORTRAN name	Description
		65	IPVOR	Vorticity distribution flag. = 0 Use constant vorticity between body elements. = 1 Use variable vorticity distribution
		68	LAST	Last body flag. = 0 This is not the last body. After this body is input the program will return to read another Body Title and Control Card for the next body. = 1 This is the last body.
		72	ITYPE	= 1
2	This card is input only when ITR = 1 or 3.			
	7(F8.0, 1X), F8.0, I1	1 - 8	CHORD	Body chord length. If input as = 0.0 the program will calculate CHORD. This parameter used only when INORM = 1.
		10-17	XMULT	X-coordinate multiplier.
		19-26	YMULT	Y-coordinate multiplier.
		28-35	DX	Delta-X increment applied to the X-coordinates.
		37-44	DY	Delta-Y increment applied to the Y-coordinates.

Card	Format	Column	FORTRAN name	Description
		46-53	THETA	Angle of rotation (positive for a counterclockwise rotation).
		55-62	XTO	X-coordinate of the center of rotation.
		64-71	YTO	Y-coordinate of the center of rotation.
		72	ITYPE	- 2

Note: These transformations are applied in the following order·

1. Rotate by angle THETA.
2. Shift by DX and DY.
3. Apply multiplier factors, XMULT/CHORD, and YMULT/CHORD.

Card	Format	Column	FORTRAN name	Description
3	6F10.0, 4X, I1, 2X, I1, 3X, I1	1-10	X(1)	X-coordinate of the geometry. Up to six points may be input on each card depending upon how the INO flag is set
		11-20	X(2)	
		21-30	X(3)	
		31-40	X(4)	
		41-50	X(5)	
		51-60	X(6)	
		65	INO	Number of data points per card. If there are 6 values per card, INO may be left blank.
		68	ISTAT	Last card flag. 0 This is not the last X-coordinate card. More cards will follow. 1 This is the last X-coordinate card.
		72	ITYPE	3

Card	Format	Column	FORTRAN name	Description
4	6F10.0,4X I1,2X,I1, 3X,I1	1-10	Y(1)	Y-coordinate of the geometry. Up to six points may be input on each card depending upon how the INO flag is set.
		11-20	Y(2)	
		21-30	Y(3)	
		31-40	Y(4)	
		41-50	Y(5)	
		51-60	Y(6)	
		65	INO	Number of data points per card. If there are six values per card, INO may be left blank.
		68	ISTAT	Last card flag. = 0 This is not the last Y-coordinate card. More cards will follow. = 1 This is the last Y-coordinate card.
		72	ITYPE	= 4
5	2X,I3,5X F10.0,51X, I1	This card is input only when ITR = 2 or = 3.		
		3-5	LX	Number of points to be generated.
		11-20	ELPSTH	Ellipse thickness ratio.
		72	ITYPE	= 5

Note: At this point in the input the geometry for one body has been read in and stored. If another body is to be loaded (i.e., a multiple element case) and the LAST parameter on the Body Title and Control Card was = 0, then another Body Titl e and Control Card will be expected next. If LAST = 1 then the geometry load has been completed and the program execution cards will be expected next.

Two-Dimensional Program Execution Cards

All of the following cards are associated with the potential flow solution and are input after the geometry data has been loaded. For most normal problems only the Flow Title Card and the Flow Control Card are required to complete the analysis. All of the other flow execution cards are associated with other options normally not used such as non-uniform flow.

Card	Format	Column	FORTRAN name	Description
6	15A4, 11X, I1	1-60	FTITLE	Title or description of case
		72	ITYPE	8
7	I1, 4X, F10.5, 2X, I1, 2X, F10 5, 5(4X, I1), 9X, I1, 4X, I1, I2, I2	1	INCLT	C_L - α flag 0 Angle of attack, α, is input. = 1 Total lift coefficient, C_L is input.
		6-15	CLT	Value of angle of attack or lift coefficient depending upon how the INCLT flag was set.
		18	ICHORD	Reference length flag. = 0 The reference length used in calculating the C_L is set - 1 0 1 The reference length used in calculating the C_L will be input as the CCL parameter
		21-30	CCL	The input value for the reference length (chord) used in calculating the C_L

Card	Format	Column	FORTRAN name	Description
		35	IND	Print control flag. = 0 Minimum print of output solutions = 1 Pring the individual solutions (0^o, 90^o, Γ, etc.).
		40	ISOL	Matrix solution method control flag. = 0 Use routine SOLVIT for the matrix solution (used when a very large number of geometry points have been input). = 1 Use routine QUASI for the matrix solution. = 2 Use routine MIS1 for the matrix solution. Maximum number of geometry points is = 101. If number of points is greater than 101 the program will automatically shift to use SOLVIT.
		45	IOFF	Off-body calculation flag. = 0 Off-body points will not be calculated. = 1 Off-body points will be calculated.
		50	NONU	Non-uniform flow flag. = 0 Non-uniform flow is not input.

Card	Format	Column	FORTRAN name	Description
				≠ 0 Non-uniform flow will be input. The number of flows input is = NONU (maximum of 6 permitted). When this option is used the program automatically sets the parameter ISOL = 1.
		55	NBNU	The number of bodies for which the non-uniform flows are input.
		65	IPRINT	Print/punch flag.
				= 0 Normal output.
				= 2 Print the individual matrices.
				7 Punch the output on cards.
		70	MORE	Last case flag.
				= 0 This is the last solution case.
				= 1 This is not the last solution case Another set of Flow Title and Flow Control Cards (and any non-uniform or off-body cards) will be expected next after this case is completed
		71	IFLLL	0 No combination solution calculated

Card	Format	Column	FORTRAN name	Description
				= 1 Combination solution calculated
		72	ITYPE	= 9
8	6F10.0, 10X, I2	1-10	CNU(1)	Combination constant for the first non-uniform flow.
		11-20 etc.	CNU(2)	Combination constant for the second non-uniform flow. etc.
		51-60	CNU(6)	Combination constant for the sixth non-uniform flow.
		71-72	ITYPE	= 10
9	I1, 1X, 3I1 10(2X, I3) F10.5, 5X, I2	This card is input only when NONU $\neq$ 0.		
		1	IBOD	Body ID number for which non-uniform flow is input
		3	IN	Normal velocity input flag. = 0 Normal velocities are not input. = 1 Normal velocities are input.
		4	IT	Tangential velocity input flag = 0 Tangential velocities are not input

Card	Format	Column	FORTRAN name	Description
		5	NN	= 1 Tangential velocities are input.
		8-10	I1(1)	I1 and IE specify the element range for which the non-uniform flow is input. I1 is the first element of the range and IE is the last element of the range. Up to five ranges are permitted per body.
		13-15	IE(1)	
		18-20	I1(2)	
		23-25	IE(2)	
		28-30	I1(3)	
		33-35	IE(3)	
		38-40	I1(4)	
		43-45	IE(4)	
		48-50	I1(5)	
		53-55	IE(5)	
		56-65	CB	Flow scaling coefficient for this body.
		71-72	ITYPE	= 11

10 This card is input only when NONU ≠ 0 and IN = 1. One set of these cards must be present for each body for which non-uniform flow is to be input (when NONU ≠ 0 and IN = 1). The number of data points in each set is determined by the number of element ranges input and by the number of elements in each element range.

Card	Format	Column	FORTRAN name	Description
	6F10.0, 10X, I2	1-10	VNUF(I)	Input values of non-uniform normal velocity. Input six values per card.
		11-20	VNUF(I+1)	
		21-30 etc.	VNUF(I+2)	
		51-60		
		71-72	ITYPE	12

11 This card is input only when NONU ≠ 0 and IT 1. One set of these cards must be present for each body for which non-uniform flow is to be input

(when NONU ≠ 0 and IT = 1). The number of data points in each set is determined by the number of element ranges input and by the number of elements in each element range.

Card	Format	Column	FORTRAN name	Description
	6F10.0,10X, I2	1-10	VTUF(I)	Input values of non-uniform tangential velocity. Input six values per card.
		11-20	VTUF(I+1)	
		21-30 etc.	VTUF(I+2)	
		51-60		
		71-72	ITYPE	= 13

Note: The cycle of input for the cards that follow the Flow Control Card is controlled by the NONU and NBNU parameters. The order of cards is as follows:

Flow Control Card (NONU and NBNU are input)

Non-Uniform Flow Coefficients

Non-Uniform Flow Body Card

Non-Uniform Normal Velocities

Non-Uniform Tangential Velocities

these cards are cycles for NBNU bodies

these cards are cycled for NONU flows

Card	Format	Column	FORTRAN name	Description
12	The following cards related to off-body calculations are input only if IOFF = 1.			
	I1, 9X, 7A4, 12X, 2(2X, I1), 2(5X, I1), 2X, I2	1	ID	Identification number for this group of off-body points. Off-body points are read in groups of up to 100 maximum at a time. There is no limit on the number of groups.
		11-38	TITLE	Title or description for this group of off-body points.

Card	Format	Column	FORTRAN name	Description
		53	ITR	Coordinate transformation flag. See the ITR parameter on card 1.
		56	INORM	Coordinate normalization flag. = 0 Off-body coordinates will not be normalized = 1 Normalize the coordinates by the input chord or by the chord for body with ID = IDOLD.
		62	IDOLD	Body selection flag for normalizing off-body points. = 0 Use input chord (Type 22 Card) to normalize off-body points. $\neq$ 0 Use chord for body with ID = IDOLD to normalize the off-body points.
		68	LAST	Off-body group termination flag. = 0 Additional group(s) of off-body points will be read in after this group is completed. = 1 This is the last group of off-body points.
		71-72	ITYPE	21

Card	Format	Column	FORTRAN name	Description
13	This card is input if ITR = 1 or = 3 on the Off-body Title and Control Card.			
	7(F8.0, 1X), F8.0, I1			
		1-8	CHORD	Body chord length This parameter used only when INORM = 1 and IDØLD = 0
		10-17	XMULT	X-coordinate multiplier.
		19-26	YMULT	Y-coordinate multiplier.
		28-35	DX	Delta-X increment applied to the X-coordinates.
		37-44	DY	Delta-Y increment applied to the Y-coordinates.
		46-53	THETA	Angle of rotation (positive for a counterclockwise rotation).
		55-62	XTO	X-coordinate of the center of rotation.
		64-71	YTO	Y-coordinate of the center of rotation.
		72	ITYPE	= 2
14	6F10.0, 4X, I1, 2X, I1, 3X, I1	1-10	X(1)	X-coordinate of the off-body point. Up to six points may be input on each card, depending upon how the INO flag is set.
		11-20	X(2)	
		21-30	X(3)	
		31-40	X(4)	
		41-50	X(5)	
		51-60	X(6)	

Card	Format	Column	FORTRAN name	Description
		65	INO	Number of data points per card. If there are six values per card, INO may be left blank.
		68	ISTAT	Last card flag = 0 This is not the last X-coordinate card. More cards will follow. = 1 This is the last X-coordinate Card.
		72	ITYPE	= 3
15	6F10.0, 4X, I1, 2X, I1, 3X, I1	1-10	Y(1)	Y-coordinate of the off-body point. Up to six points may be input on each card depending upon how the INO flag is set.
		11-20	Y(2)	
		21-30	Y(3)	
		31-40	Y(4)	
		41-50	Y(5)	
		51-60	Y(6)	
		65	INO	Number of data points per card. If there are six values per card, INO may be left blank.
		68	ISTAT	Last card flag. = 0 This is not the last Y-coordinate card. More cards will follow. = 1 This is the last Y-coordinate Card.
		72	ITYPE	= 4

Card	Format	Column	FORTRAN name	Description
16	This card is input only when ITR = 2 or = 3 on the Off-Body Title and Control Card.			
	2X, I3, 5X, F10.0, 51X, I1	3-5	LX	Number of points to be generated.
		11-20	ELPSTH	Ellipse thickness ratio.
		72	ITYPE	5

24Y OUTPUT

Printed Output

For each body the following data are output: UNTRANSFORMED COORDINATE DATA FOR BODY xx and ELEMENT COORDINATE DATA FOR BODY xx. The untransformed data are the geometric points which were input from SCIRCL. For the element data, the midpoint of each segment is calculated. Also printed are the linear distance and surface distance between each segment. For each element, the sine, cosine, and local curvature are also output. Following the element data is the Antal surface distance for that body.

After the input and transformed coordinates are output a BODY GEOMETRY SUMMARY table is output. The following is output in the summary:

BODY DESCRIPTION	Title of test case
BODY ID	Body number
LIFT TYPE	= Yes, vorticity solution about this body is calculated
	= No, no vorticity solution calculated
N/O	New or old body data
SID	= S, body geometry data saved
TFORM	= Yes, body coordinates have been transformed (see ITR in input)
	= No, body coordinates have not been transformed

NORM	= Yes, body coordinates have been normalized (see INORM in input)
	= No, body coordinates have not been normalized
CHORD	If INORM = 1, the CHORD is the body chord length. If INORM = 0, CHORD 000
TYPE	Type of solution to be generated.
	= L, only linear terms
	= P, parabolic terms
SIGMA	If TYPE = P, SIGMA describes which parabolic terms considered
F	= 1, First derivative term
C	= 1, Curvature term
S	= 1, Second derivative term
C	= 1, Curvature squared term
ELEMENT STORAGE	
N/O	See N/O above
FIRST	Number of first element for given body
NO	Number of elements for given body.
Total Number of Bodies Total Number of Elements	Self explanatory

Following the GEOMETRY SUMMARY table is a table of the vorticity for each element. If IPVOR does not equal zero then a variable vorticity for each element is calculated If IPVOR equals zero then the vorticity is a constant for each element.

Following the vorticity functions is a printout of the combination coefficients used internally by 24Y for its version of a combined solution based upon input alpha or total lift if INCLT = 1.

After the combination coefficients of INDIVIDUAL FLOW SOLUTIONS for each body are printed. The first solution is axial flow and the second individual solution is for crossflow. There is also an individual solution based upon each body which is a lifting body. Each individual solution contains the following:

POINT NUMBER	Element number
VN	Normal velocity at the point
VT	Tangential velocity at the point
SIGMA	Source strength density

The individual solutions are also printed on a disc file to be used as input to NOZZLEC as basic solutions.

The next section of output is the COMBINED SOLUTION if IFILL = 1. This solution is the result of the combination of individual solutions into a solution which satisfies the input conditions of either alpha or total lift. Also included are the integrated values of lift for each body.

If there are off-body points specified in the input, the next section is the individual solutions at each off-body point. These solutions correspond to those for the on-body points. Also included is a combined solution for the off-body points (if IFILL = 1) using the same combination coefficients as those for the on-body points.

NOZZLEC Input

English engineering units are used throughout the program.

Length, in.
Velocities, ft/sec
Angles, deg
Pressure, lb/ft^2
Temperatures, ^{o}R
Densities, slug/ft^3
Force, lb
Weight flow, lb/sec

Card	Format	Column	FORTRAN name	Description
1	3A6	1-18	TITLE	Title card
2	9I4	1-4	NT	Total number of on-body points. [NT(3) from

Card	Format	Column	FORTRAN name	Description
				SCIRCL (3 body case), or NT(2) from SCIRCL (2 body case)] .
		5-8	NS1	Number of on-body points on body 1. (NT(1) from SCIRCL).
		9-12	NH	Number of on-body points on bodies 1 and 2 (NT(2) from SCIRCL) If there are only 2 bod- ies, NH = number of on- body points on body 1 only. (NT(1) from SCIRCL).
		13-16	NP	Total number of off-body points. (NP from SCIRCL).
		17-20	IW	Flag for type of input through the control sta- tion: 0, weight flow 1, Mach number 2, velocity
		21-24	NX	1, apply supersonic velocity correction to data
		25-28	KND	Flag for scaling variables. All input lengths are divi- ded by ELND If KND -1, ELND YCU 0, ELND 1 1, ELND YCU - YCL

Card	Format	Column	FORTRAN name	Description
				= 2, ELND = the read-in value from card 4.
		29-32	ICOMP1	Type of solution to be computed: = 0, compressible = 1, incompressible
		33-36	IHUB	IHUB = 0, 2 body case ≠ 0, 3 body case
3	10F8.0	1-8	VC	Average axial velocity at the control station between bodies 1 and 3 upstream of the body 2.
		9-16	VS1	Average axial velocity at the control station between bodies 1 and 2.
		17-24	VS2	Average axial velocity at the control station between bodies 2 and 3. Note if IW=2, then two of the three velocities must be input for the two passage case
		25-32	VINF	Free stream velocity
		33-40	ALFA	Angle between free stream velocity and X-axis of the nozzle
		41-48	MC	Average Mach number at the control station between bodies 1 and 3.
		49-56	MC1	Average Mach number at the control station between bodies 1 and 2.

Card	Format	Column	FORTRAN name	Description
		57-64	MC2	Average Mach number at the control section between bodies 2 and 3. If IW=1 then two of the three Mach numbers must be input for the two passage case
		65-72	TTOTAL	Total temperature. If TTOTAL = TSTAT=0, then TTOTAL = 518.67 will be used.
		73-80	PT	Total pressure. If PT=0.0 and PSTAT≠0.0, the program will calculate PT. If PT = 0.0 and PSTAT=0.0, PT is set to 2116.
4	10F8.0	1-8	ELND	ELND is the arbitrary length used for scaling or normalizing. Refer to KND input.
		9-16	WDOTC	Weight flow at the control station between bodies 1 and 3.
		17-24	WDOTC1	Weight flow at the control station between bodies 1 and 2.
		25-32	WDOTC2	Weight flow at the control station between bodies 2 and 3. If IW=0, then two of the three weight flows must be input for the two passage case.

Card	Format	Column	FORTRAN name	Description
		33-40	PSTAT	Static pressure
		41-48	TSTAT	Static temperature. If PSTAT and TSTAT are not 0.0, total temperature and total pressure will be calculated using PSTAT and TSTAT.
		49-56	CUTOF1	If CUTOF1 $\neq 0$, then the pressure ratio P_S/P_T on body 1 will be plotted against a dimensionless surface distance S/CUTOF1 starting at X = XR1 for a distance of S = CUTOF1.
		57-64	CUTOF3	Same as CUTOF1 except for body 3.
		65-72	CUTOF2	Same as CUTOF1 except for body 2.
		73-80	VPERIN	If VPERIN is greater than zero then a CALCOMP plot of the nozzle showing rake point flow field data will be produced. VPERIN is the value of a unit vector in ft/sec/in. If VPERIN is not equal to zero, card "4a" must be input.
4A	6F10.0	1-10	XX	The length in plot inches of the abscissa of the velocity plot.
		11-20	XMIN	Value, in data inches, of far left X-point.

Card	Format	Column	FORTRAN name	Description
		21-30	EXEP	Data inch per plot inch along X-axis.
		31-40	YY	Length, in plot inches, of the ordinate.
		41-50	YMIN	Value, in data-inches, of bottom Y point.
		51-60	ORD	Data inch per plot inch along Y-axis.
5	3F10.0	1-10	XTEST	Axial location of the control station upstream of body 2 between bodies 1 and 3.
		11-20	YCL	Y on body 1 at XTEST
		21-30	YCU	Y on body 3 at XTEST
6	3F10.0	1-10	XTEST1	Axial location of the control station between bodies 1 and 2.
		11-20	YCL1	Y on body 1 at XTEST1.
		21-30	YCU1	Y on body 2 at XTEST1.
7	3F10.0	1-10	XTEST2	Axial location of the control station between bodies 2 and 3.
		11-20	XCL2	Y on body 2 at XTEST2
		21-30	YCU2	Y on body 3 at XTEST2

Note: Cards "6" and "7" are not used for a single-passage case.

8	3F10.0	1-10	XR1	Axial location on body 1 where S 0.
		11-20	XR3	Axial location on body 3 where S 0.

Card	Format	Column	FORTRAN name	Description
		21-30	XR2	Axial location on body 2 where S = 0. Leave blank for two body case.
9	3F10.0	1-10	YR1	Y on body 1 at XR1.
		11-20	YR3	Y on body 3 at XR2.
		21-30	YR2	Y on body 2 at XRH.

Figure 15 is the input form for NOZZLEC.

NOZZLEC Output

Printed Output

Input file dump
TITLE - followed by 2-D COMBINATION SOLUTION
Version of run (i.e., compressible or incompressible)
A list of the basic flow solutions obtained from 24Y
In the table that follows, several functions of four different velocities are given. The velocities are:

Control: VC, average axial velocity at upstream control station

Lower passage: VS1, average axial velocity at control station between bodies 1 and 2

Upper passage: VS2, average axial velocity at control station between bodies 2 and 3

Free stream: V_∞, free stream velocity

The rest of the table is self-explanatory perhaps with the exception of the terms INC and COMP. INC means calculated from the incompressible equations and COMP means calculated from the compressible equations. The rest of the output will be defined by its name.

ALPHA	Angle of attack of nozzle
VINF/VC	V_∞/V_c
VSONIC	Critical velocity uncorrected for compressibility
VSONICC	Critical velocity

WDOTCR	Corrected weight flow at upstream control station $\frac{\text{WDOT} \times \sqrt{\text{THET}}}{\text{DEL}}$
WDOTLCR	Corrected weight flow at control station between bodies 1 and 2
WDOTUCR	Corrected weight flow at control station between bodies 2 and 3
TSTAT	Free stream static temperature
PSTAT	Free stream static incompressible pressure
PSTATC	Free stream static compressible pressure
ASTAT	Free stream static speed of sound
RHOSTAT	Free stream static density
WDOTC	Input mass flow at upstream control station
WDOTL	Input mass flow at downstream lower control station
WDOTU	Input mass flow at downstream upper control station
VIC	Incompressible average velocity at upstream control station
VICL	Incompressible average velocity at downstream lower control station
VICU	Incompressible average velocity at downstream upper control station
TTOT	Free stream total temperature
PTOT	Free stream total pressure incompressible
PTOTC	Free stream total pressure compressible
ATOT	Free stream stagnation speed of sound
RHOTOT	Free stream stagnation density
THET	TTOT/518.67
DEL	PTOTC/2116.22
XRI1	Input
YRI1	Input

XRI2	Input
YRI2	Input
XTEST	Input
YCL	Input
YCU	Input
LND	Length used for scaling
XTEST1	Input
YCL1	Input
YCU1	Input
XTEST2	Input
YCL2	Input
YCU2	Input
P - S CUTOFF1	Input
P - S CUTOFF2	Input
P - S CUTOFF3	Input
NT	Input
NP	Input
NS1	Input
NH	Input
KND	Input
IW	Input
NX	Input
ICOMP1	Input
IHUB	Input
V1	Average axial velocity at specified control station for basic solution 1 from 24Y
V2	Same as V1 except for basic 24Y solution 2
V_3	Same as V1 except for basic 24Y solution 3

V4	Same as V1 except for basic 24Y solution 4
V5	Same as V1 except for basic 24Y solution 5

24Y basic solutions 1 and 2 are axial and crossflow solutions. For a three-body case, solutions 3, 4, and 5 are vorticity solutions about bodies 1, 2, and 3, respectively. For a two-body case, solutions 3 and 4 are the vorticity solutions about bodies 1 and 3. Note there is no 5 solution for this case.

A, B, C, D	Coefficients of combination
VINFP	Incompressible free stream velocity "uncorrected" for compressibility if the input value was compressible

OTHER MESSAGES: "VRESON = ____________ IS GREATER THAN VMAX. VCONC = ____________." The velocity at a certain on-body point exceeds the allowable value for the local expansion condition so that the isentropic ratio term: 1. - VCONC is less than zero. Where,

$$VCONC = \frac{\gamma - 1}{2}\left(\frac{VRESON}{a_{tot}}\right)^2.$$

"I EXCEEDS 20 ITERATIONS FOR RHOBAR. VBAR = ____________, VCOMP = ____________, RHOBAR = ____________. VBAR HAS BEEN REDUCED TO VCOMP * RHOBAR/RHOTOT". Subroutine VBARIT attempts to find the average density at each axial location using the isentropic density ratio, the stagnation density (RHOTOT) and the average incompressible VBAR (based on weight flow and the cross section). It has failed. VCOMP is the 20th attempt at finding the compressible velocity and has been used to compute the RHOBAR that will be returned. The normally unchanged VBAR is adjusted to agree with these abbreviated results.

For NX = +1, supersonic velocity correction is operating and a message to that effect will appear each time a region of local supersonic flow is encountered on the body, and also when it ends. The body point number where these transitions occur will also be printed.

ON-BODY POINTS

Body 1, 2, or 3

For both compressible and incompressible versions:

I	The index number of the point
X	Axial distance
Y	Height
S	Surface distance from XRI

Compressible Version

VCOM	Resultant velocity with compressibility correction applied
VBAR	Average incompressible velocity at a given axial location
MACH	Mach number
CP	Compressible pressure coefficient $\left(\frac{p_i - p_o}{q_o}\right)$
RB/RT	$\bar{\rho}_c/\rho_t$
PS/PT	Static to total pressure ratio, $\left[1 - 0.2\left(\frac{V}{a_t}\right)^2\right]^{3.5}$

Incompressible Version

VINC	Resultant incompressible velocity
MACH	Mach number (based on incompressible velocity and compressible flow equations)
CP	Incompressible pressure coefficient $\left[1 - \left(\frac{V_i}{V_o}\right)^2\right]$
PS/PT	Static to total pressure ratio, $1.0 - \frac{1/2\,\rho V^2}{P_t}$

OFF-BODY POINTS

For both incompressible and compressible versions:

Rake number

I	Number of the point (points without numbers are interpolated values at a body)
X	Axial location
V	Vertical location
THETA	Flow angle, $\tan^{-1}\left(\frac{VY}{VX}\right)$
MACH	Mach number
WFRACT	Local cumulative weight flow at a given point on a rake divided by total weight flow at the rake

Compressible Version

VX	Axial velocity corrected for compressibility
VY	Vertical velocity corrected for compressibility
VRE	Resultant velocity = $\sqrt{VX^2 + VY^2}$
VBL	Average incompressible velocity at given axial location
RB/RT	$\bar{\rho}/\rho_t$
PS/PT	Same as ON-BODY POINTS

Incompressible Version

VX	Incompressible axial velocity
VY	Incompressible vertical velocity
VRE	Resultant velocity = $\sqrt{VX^2 + VY^2}$
PS/PT	Same as ON-BODY POINTS

RAKE WEIGHT FLOW DATA

For each rake the following data are given:

I	Number of rake
X	Axial location of rake
IRAK	Parameter to describe location of rake on the inlet = 1; upstream of nozzle = 3; upstream of body 2 between bodies 1 and 3 = 4; downstream of body 2 highlight between bodies 1 and 2 = 5; downstream of body 2 highlight between bodies 2 and 3
WDOT	Integrated weight flow for rake
WDOTCA	Specific corrected weight flow at each rake
MACH	One-dimensional Mach number at each rake based on WDOTCA

Graphic Output

Standard CALCOMP plots of PS/PT (fig. 9) and Mach number (fig. 10) distribution against S/CUTOF1, S/CUTOF2, or S/CUTOF3 are made for any body whose value of cutoff is not zero.

If VPERIN not equal to zero, then a plot will be made of the velocity flow field at the off-body points (see fig. 11). At each off-body point a vector will be drawn showing flow angularity and magnitude. Magnitude will be shown by relative size of the vectors (a unit vector will have a magnitude of VPERIN).

DESCRIPTION OF SUBROUTINES

Figure 16 illustrates the calling relations between the main program and their subroutines.

Program SCIRCL

(A) MAIN SCIRCL — Read all input, call required subroutines for each segment as requested, plot each segment after points are generated by subroutine; list points; test for reworking of geometry if required

Straight Lines

(B) STRAIT	Generate points on a general straight line segment
(C) FNSTRH	Generate points on final straight segment of a body
(D) FRSTSH	Generate points on first straight segment of a body

Bisuper Ellipses

(E) TEST	Test superellipse input to see if mirroring about y-axis is required
(F) PRELPS	Mirror superellipse input data about y-axis so that slope (1, 2) is greater than the slope (1, 4) (fig. 12 (a))
(G) SUPERC	Generated points on a general bisuperellipse (table I)
(H) FONISØ	Iterate on input conditions to find bisuperellipse exponents

Other Curves

(I) CUBIC	Fit a cubic polynomial between two nonvertical parallel lines
(J) SIMQ	Simultaneous solution of equations to obtain coefficients of the cubic polynomial
(K) LEM	Generates points on a general Lemniscate
(L) MIRROR	Mirror the hub points to obtain the shroud

Direct Interpolation

(M) XYCALC	Executive routine for the following modules purpose is to generate points "correctly" spaced along the curve defined by a list of input points. Inputs are used to develop double 3-point interpolating polynomial in successive regions along the curve. Polynomials are then used to suggest points, derivatives, etc. which can be tested for spacing as defined by standard criterion (see comments in SPGEN listing)

(N) SGEN	Refer to listing for comment card description
(O) DSTRP	
(P) SPGEN	
(Q) DNTRPC	
(R) FNTRP	
(S) FSTRP	
(T) FNTRPC	
(U) FNTRPA	
(V) TLU	
(W) LIMIT	

Special Calculations; Output to 24Y

(X) WPUNCH	Generate rake points at requested positions; plot rakes
(Y) WRTXY	Write all X, Y coordinates
(Z) AREAA	Compute area

Picture Plotting

(Z1) DRAW	Plot X-Y meridional plane picture of each inlet segment
(Z2) PLOXIS	Plot frames for inlet picture and label axis

Utility

(Z3) SINTP	Lagrange three-point interpolation
(Z4) SORTXY	Rearrange the values in an array, x, to increase with increasing index (ascending order); sort y accordingly

System Library

ERTRAN	Routine which gives FORTRAN access to several UNIVAC 1110 operating system commands. Can be eliminated by defining unit 25 before execution

SYMBOL SCALE LINE NUMBER PLOT	Standard CALCOMP routines needed for all plotting

Program 24Y

A) MAIN 24Y — Executive calls to MAIN1, SOLVE, and MAIN3 also, this program rewinds all disc files used.

B) ABFORM — This subroutine calculates induced velocities array elements in arrays A and B.

C) ASSEMB — This subroutine reads in normal and tangential onset flows in row order and assembles them in column order. Also it reads in non-uniform onset flows if any.

D) COMBO — This subroutine calculates combination coefficients if required.

E) ELFORM — This subroutine reads in body coordinates and associate data and calculates element data.

F) FILES — This subroutine assigns temporary file numbers.

G) FLOWS — This subrouting calculates individual and combined flow for on-body points. It also writes the individual flows on a tape file for input to NOZZLEC.

H) GEOMCF — This subroutine calculates required geometry combination coefficients associated with parabolic distribution.

I) MAFORM — This subroutine forms and stores matrices A and B and calculates alpha and circulatory onset velocities.

J) MAIN1 — This subroutine calls subroutine ELFORM and MAFORM and reads in the flow control parameters.

K) MAIN3 — This subroutine calls COMBO (if required), FLOWS, and OFFBOD.

L) MIS1 — This subroutine inverts a matrix and solves simultaneous equations. This is used when ISOL = 2 and number of points less than 102. If more than 101 points are input program shifts to SOLVIT.

M) OFFBOD — This subroutine calls OFFPTS, VXYOFF, and VPROFF.

N) OFFPTS — This subroutine reads in the off-body points and prints them back out.

O) PRINTG — This subroutine writes out the body coordinate data.

P) PRNTEL — This subroutine writes out the body summary data.

Q) QUASI — This subroutine performs direct matrix solution. Program uses this to solve matrix when ISOL = 1.

R) RMAX — This subroutine calculates maximum radius or chord.

S) SOLVE — This subroutine obtains the sigma solutions from QUASI.

T) SOLVIT — This subroutine performs direct matrix solution (used when a very large number of points have been input).

U) TYPE — This subroutine checks input card type to verify proper data card is being input.

V) VXYOFF — This subroutine calculates the individual and combined solutions for each off-body point.

W) VPROFF — This subroutine prints out the individual and combined solutions (in blocks of 100) for each off-body point.

X) WEIGHT — This subroutine calculates the vorticity weighting functions for each element.

Y) XYFORM — This subroutine calculates the indiced velocity array elements A and B.

Program NOZZLEC

(A) MAIN NOZZLEC — Executive calls to INPTR, SEARCH, ANGLEF, SOLVF, OFBDY, and if compressibility correction desired COMCOR. Also, this program calls PLTER if output plots are desired

(B) INPTR	Reads input parameters from unit 5. Also reads 24Y output coordinates and basic flow solution velocities.
(C) CONST	Calculated most constants and intermediate parameters and prints results.
(D) SEARCH	Finds the highlight on each body.
(E) ANGLEF	Finds the body surface angle for each point on the body.
(F) SURF	Calculates surface distance along body as a function of X.
(G) SOLVE	Computes linear combination coefficients A, B, C, and D to satisfy input flow conditions. Also calculates the incompressible velocities and average velocity $\overline{V}_i$ (used in compressibility correction) for each point.
(H) COMCOR	Applies compressibility correction.
(I) ONBODY	Uses velocities and densities to calculate: pressure ratios, Mach number, flow angles and list all results for points on the body.
(J) OFBDY	Same as ONBODY for rakes. Also calculates local fractional weight flows for each rake point.
(J) VBARIT	Calculates average density ratio for compressibility correction.
(L) INTER	See SCIRCL routine SINTP (duplicate)
(M) SORTXY	See SCIRCL routine SORTXY (duplicate)
(N) INTER2	Calls INTER
(O) INTER3	Calls INTER
(P) INTEG	Performs trapezoidal integration
(Q) CALTIT	Titles plots of pressure and Mach number versus surface distance.
(R) PLTER	Plots pressure and Mach number versus surface distance.

INPUT/OUTPUT UNITS

SCIRCL	5 - Standard card input
	6 - Standard output list
	4 - Temporary storage; if flag J or E is 1, input to direct interpolation routines is written here
	17 - Saved. If flag A is 0, input for 24Y consisting of X, Y points is written here
24Y	5 - Input (=7 from SCIRCL)
	6 - Standard output list
	7 - Saved. Input for NOZZLEC is written here (6E13.8) X, Y, V1, V2, V3, etc.
	2-4, 8-18 - Temporary storage
NOZZLEC	5 - Standard card input, flow conditions, etc.
	6 - Standard output list
	7 - Input from 24Y
	12 - Saved. Input data (for body 1) for boundary layer program
	13 - Saved. Input data (for body 3) for boundary layer program
	14 - Temporary storage

Program SCIRCL

```
HAWK*SCIRSO(1).SCIRCLE
      C     PREPARE INPUT DATA FOR DOUGLAS POTENTIAL FLOW PROGRAMS EOD AND 23YA   0000
      C                                                                         A   0010
      C                                                                         A   0030
            DIMENSION REEDEN(2),ARE(9),EX(9),CURVO(9),SURFAC(9),NBIT5T          A   0040
            DIMENSION CAPPER(200),DIST(200),KAY(9),TYP(9),IREDON(4)
            DIMENSION            S(500), NY(25)
            COMMON /SPREP/ KPREP,NIN                                            A   0070
            COMMON /MNSD/ NNSD,NSDBOY(10)                                       A   0080
            COMMON /NHIGH/ NSPHG,NLAST,XLAST(500),YLAST(500)                    A   0090
            COMMON /MWRTE/ IFLAG,NDY4,PROG,TITLE(9),BODIES(4),IDENT,YLO(25),YHA   0100
           1I(25),NDY(25),XRAK(25),NBDPTS(9),NO6,NRAKES
            COMMON /FOR3SS/IO,DELS,XBK(20),YBK(20),XON(500),YON(500),DYDXO(500A   0120
           1),ALPHA(500),CAPPA(500),SON(500),PIO180                             A   0130
            COMMON /FOREOD/ IGEOMF,ISIGF,ICURVN,NONEWF,IVORT,ALPHER             A   0140
            COMMON /SUPF/ IFLD                                                  A   0150
            COMMON/SEGNO/NSEG,J,ENREED                                          A   0160
            COMMON /MAIN/ XIN(10),YIN(10),DELSMX,PIO2,DELS1,IHUB,DELNEW         A   0170
            COMMON /SS/ NBDY1,NBODY2,TYPBDY,NBDYS,D2TEST
      C(((((( THE GENERAL PLOTTING VARIABLES                                    A   0190
            COMMON/ LOT/XMIN,YMIN,ORD,EXEP,XX,YY,LPNCHO,IPLOTA,MM               A   0200
            COMMON/TITL/ TTITL(9,6)                                             A   0210
            COMMON/TOL/BAGS(15),BAGX(15),ZAP(15),NZAP(15)                       A   0220
            COMMON/SENSE/  X(2),Y(2),A,B                                        A   0230
            COMMON/PAC/ PACE,DELSHL                                             A   0240
            COMMON/BURLEY/ ELREF,AREF
            COMMON/OHVORT/ NLF(10)
            LOGICAL TRANSX,TRANSY,SCALEX,SCALEY,CHANGE
            COMMON/TRANSF/ XTRAN,YTRAN,XSCALE,YSCALE,
           1  TRANSX,TRANSY,SCALEX,SCALEY,CHANGE,XTMAX
            COMMON/Y24Y/ IPAR,IFST,ISND,IFLLL
            DATA D2TEST/6H   23Y/
            DATA BLANK/6H      /                                                A   0250
            DATA EODFF/6HENDOFF/                                                A   0260
            DATA REDONE/6HREDONE/                                               A   0270
            PI=3.14159265                                                       A   0280
            PIO180=PI/180.                                                      A   0290
            PIO2=PI/2.                                                          A   0300
      C                                                                         A   0310
      C     WHEN NO6 = 1, A FLAG IN CARD COLUMN 6 IS PUNCHED FOR EOD            A   03
      C      ONLY BASIC DATA WILL BE GIVEN IN EOD PROGRAM
      C                                                                         A   0340
            CALL PLOTID                                                         A   0350
      C(((((( READ AXIS LABELS FOR THE MOST POPULAR PLOTTED VARIABLES           A   0360
            READ (5,500)ARE                                                     A   0370
            READ (5,500)EX                                                      A   0380
            READ (5,500)CURVO                                                   A   0390
            READ (5,500)SURFAC                                                  A   0400
            CALL ECHO                                                           A   0410
         10 NIN=25                                                              A   0420
            READ (25,500,END=630)TITLE                                          A   0430
            IF (FLD(0,36,TITLE(1)).EQ.FLD(0,36,EODFF)) GO TO 630                A   0440
            REWIND 4                                                            A   0450
            LOWER=0                                                             A   0460
      C(((((( READ GENERAL PLOTTING VARIABLES                                   A   0470
            READ(25,505) XX,XMIN,EXEP,YY,YMIN,ORD,ELREF,AREF                    A   0480
            IF(ELREF .EQ. 0.0) ELREF = 1.
```

```
      IF(AREF .EQ. 0.0) AREF = 1.0
      READ(25,15)ICEOMF,ISIGF,ICURVN,NONEWF,ALPHER,IVORT,IPAR,IFST,ISND
     1  ,IFLLL
   15 FORMAT(4I1,F10.2,5I1)
   20 READ (NIN,490)IDENT,PROG,NO6,LPNCHO,IPLOTA,IPLOTC,IPEAD,IAB,(IREDOA    0510
     2N(I),I=1,5)
      PROG1=BLANK                                                       A    0520
      IF(PROG.EQ.'   23Y') PROG1='   23Y'
      IF(PROG.EQ.'   24Y') PROG1='   24Y'
      IF(PROG1.NE.'   23Y'.AND.PROG1.NE.'   24Y') PROG1=PROG
      IF(PROG.EQ.'   24Y') PROG='   23Y'
      LPDUM=LPNCHO
      IF(LPDUM.EQ.0)LPNCHO=1                                            A    0530
      IF(LPDUM.EQ.1)LPNCHO=0                                            A    0540
   25 JSTART=0                                                          A    0550
      JSTOP=0                                                           A    0560
      PACE=0.                                                           A    0570
      IF (IAB.LE.0) GO TO 30                                            A    0580
      READ (NIN,625)XAA,YAA,XBB,YBB                                     A    0590
      IBUMB=0                                                           A    0600
      WRITE (4,495)IDENT,PROG,NO6,(IREDON(I),I=2,4),IBUMB,(IREDON(I),I=2A    0610
     2,4)                                                               A    0620
C(((((( LEEHI WILL CONTAIN THE VALUE OF N AT THE HIGHLIGHT. NEEDED TO SPA    0630
C(((((( THE CURVATURE VS. X PLOTS INTO INTERNAL AND EXTERNAL PORTIONS   A    0640
C(((((( MM COUNTS THE NO. OF SEGMENTS ON SHROUD AS PLOTTING PROCEEDS    A    0650
   30 MM=0                                                              A    0660
      LEEHI=9000                                                        A    0670
C(((((( LOAD AXIS LABELS INTO COMMON                                    A    0680
      IF (NIN.EQ.4) GO TO 40                                            A    0690
      DO 35 I=1,9                                                       A    0700
      TTITL(I,1)=TITLE(I)                                               A    0710
      TTITL(I,2)=BLANK                                                  A    0720
      TTITL(I,3)=ARE(I)                                                 A    0730
      TTITL(I,4)=EX(I)                                                  A    0740
      TTITL(I,5)=CURVO(I)                                               A    0750
   35 TTITL(I,6)=SURFAC(I)                                              A    0760
   40 WRITE (6,525)IDENT,TITLE                                          A    0770
      IF (NIN.EQ.4) WRITE (6,470)                                       A    0780
      IF (NIN.EQ.4) GO TO 55                                            A    0790
      IF (IVORT.EQ.0) GO TO 50                                          A    0800
      IVORT=0                                                           A    0840
      GO TO 55                                                          A    0850
   50 IVORT=1                                                           A    0860
55    CONTINUE                                                          A    0870
      IF(LPNCHO.GT.0)WRITE(5,631)
      IF(LPNCHO.EQ.0)WRITE(6,632)
      WRITE(6,633) PROGI
      IF(IGEOM+ISIGF+ICURVN+NONEWF.EQ.0)WRITE(6,634)
      IF(IGEOM+ISIGF+ICURVN+NONEWF.NE.0)WRITE(6,635)
      IF(IVORT.EQ.1) WRITE(6,636)
      IF(IVORT.NE.1) WRITE(6,637)
      WRITE(6,638)
      IF(IPLOTA.NE.0) WRITE(6,639)
      IF(IPLOTA.EQ.0) WRITE(6,640)
      IF(IPLOTC.EQ.0) WRITE(6,641)
      IF(IPLOTC.NE.0) WRITE(6,642)
      IF(IAB+IREDON(I).NE.0)WRITE(6,643)
```

```
      IF(IAH+IRFLON(I).FL. )WRITE(6,644)
      IF(ABS(XX).GT.100.) WRITE(6,64) XX
 64   FORMAT(17H,YLEN (XX) = ,F10.2,30H.GT. 100.. CHECK FORMATS,INPUT)
      IF(ABS(XX).GT.100.) STOP
C((((( PLOT THE AXES NEEDED FOR THE INLET PICTURE, AND LABEL THE CASE
   65 CALL PLOXIS(XX,YY,IXEP,ORD,XMIN,YMIN,.25,.25,0,0,1,2,1,1)
      CALL SYMBOL(-.3,YY-.5 ,.25,IDENT,0.,6)
      CALL SYMBOL(XX-1.5,-.5,.25,IDENT,0.,6)
      IF(NIN.EQ.4)CALL SYMBOL(XX/2.,YY-.5,.25,RFDONE,0.,6)
      IF (NIN.EQ.4) GO TO 75
C
C     READ INPUT CARDS FOR SUPERCIRCLE
C         1  -- CASE HEADER CARD -- NO. OF BODIES,CASE NO.,DELS,DELSMX
C      OFF-BODY
C         2 -- NRAKES = NUMBER OF RAKES (TOTAL NUMBER CANNOT EXCEED 25)
C         3 -- X,YLO,YHI, NY (DATA FOR EACH RAKE)
C              X = X OF THE RAKE,
C     YLO = Y OF THE FIRST PT. ON RAKE CLOSEST TO THE HUB - SHOULD BE
C                ABOUT DS GREATER THAN Y ON HUB
C     YHI = Y OF THE LAST PT ON RAKE CLOSEST TO THE SHROUD - SHOULD BE
C                ABOUT DS LESS THAN Y ON SHROUD
C              NY  = NO. OF PTS TO GENERATE FOR THAT RAKE
C     ON-BODY
C         FOR EACH SEGMENT A DESCRIPTION CARD IS NEEDED,
C           THIS CARD DENOTES THE TYPE OF LINE, AND THE
C           COORDINATES OF THE LINE (UP TO 6 SETS)
C
      READ (NIN,555,END=630)ANBDYS,DELS,DELSMX,XRI,ANNSD
      HOLYDS=DELS
      READ (NIN,485)NRAKES
C
C     RAKE POINTS ARE TRANSFORMED INDEPENDENT OF ONE ANOTHER AND
C     INDEPENDENT OF BODY TRANSFORMATIONS. ZERO VALUES FOR YLO AND
C     YHI ARE TRANSFORMED AUTOMATICALLY (BY CALCULATION FROM TRANSFORMED
C     BODY POINTS) IN SUBROUTINE WPUNCH.
C
      DO 67 I=1,NRAKES
      READ(NIN,550) XRAK(I),YLO(I),YHI(I),NY(I),
     1              XTRAN,YTRAN,XSCALE,YSCALE
      TRANSX = XTRAN .NE. 0.0
      TRANSY = YTRAN .NE. 0.0
      SCALEX = XSCALE .NE. 0.0
      SCALEY = YSCALE .NE. 0.0
      CHANGE = TRANSX .OR. TRANSY .OR. SCALEX .OR. SCALEY
      IF(.NOT. CHANGE) GO TO 67
      IF(.NOT. SCALEX) XSCALE = 1.0
      IF(.NOT. SCALEY) YSCALE = 1.0
C
C     TRANSFORM THE RAKE POINTS
      XRAK(I) = XRAK(I) * XSCALE + XTRAN
      YLO(I) = YLO(I) * YSCALE + YTRAN
      YHI(I) = YHI(I) * YSCALE + YTRAN
   67 CONTINUE
      WRITE (6,480)ANBDYS,DELS,DELSMX,XRI
      DO 7( I=1,NRAKES

      N[illegible](I)=NY(I)
```

```
   75 NLAST=0
      NSPHG=0                                                  A    1210
      DSAVE=DELS                                               A    1220
      DELS2=DELS                                               A    1230
      DELS1=DELS                                               A    1240
      NBDYS=ANBDYS                                             A    1250
      NNSD=ANNSD                                               A    1260
C                                                              A    1270
C     K = COUNTER FOR THE NUMBER OF ONBODY POINTS GENERATED    A    1280
C     IFLAG = 0, IF THERE IS MORE THAN ONE BODY                A    1290
C                                                              A    1320
      K=0                                                      A    1330
      IHUB=0                                                   A    1340
      IFLAG=1                                                  A    1350
C
C     NZ LOOP IS FOR THE NUMBER OF BODIES                      A    1370
C                                                              A    1380
      DO 225 NZ=1,NBDYS                                        A    1390
      IFLD=0                                                   A    1400
      IF (NZ.GE.2) IFLAG=0                                     A    1410
      IF (NZ.GE.2) IHUB=1
C                                                              A    1420
C                                                              A    1430
      READ(NIN,558) TYPBDY,ANSEG,DELNEW,ANLF,XTRAN,YTRAN,XSCALE,YSCALE   A    1500
     1               ,XTMAX
      NSEG=ANSEG                                               A    1530
      NLF(NZ)=ANLF
      NUMBOD=IABS(IFIX(TYPBDY))
      WRITE(6,626) NUMBOD
C
C     SET LOGICALS RELATED TO TRANSLATING OR SCALING INPUT BODY POINTS
C
      CHANGE = .FALSE.
      TRANSX = XTRAN .NE. 0.0
      TRANSY = YTRAN .NE. 0.0
      SCALEX = XSCALE .NE. 0.0
      SCALEY = YSCALE .NE. 0.0
      IF(TRANSX .OR. TRANSY .OR. SCALEX .OR. SCALEY) CHANGE = .TRUE.
      IF (IAB.GT.0) WRITE (4,555)TYPBDY,ANSEG,DELNEW,ANLF
      IF(DELNEW.LT.0.)DELS1=HOLYDS
      IF(DELNEW.GT.0.)DELS1=DELNEW
      K=K+1                                                    A    1600
      IF (TYPBDY.LE.0.0) GO TO 215
      SON(K)=0.0                                               A    1610
C                                                              A    1620
      IF(NZ.EQ.NBD)NSEG=NSEG-LOWER                             A    1630
C     SEGMENT LOOP                                             A    1640
C                                                              A    1650
      DO 200 J=1,NSEG                                          A    1660
C((((((  KK= FIRST POINT ON CURRENT SEGMENT TO BE PLOTTED      A    1840
      KK=K                                                     A    1850
C((((((  DIRECT INTERPOLATION FLAG--                           A    1670
   80 IF (NSEG.EQ.0) GO TO 155                                 A    1680
      READ (NIN,555)ENRFED,(REEDEN(I),I=1,2)                   A    1690
      IF (IAB.GT.0) WRITE (4,555)ENREED,(REEDEN(I),I=1,2)      A    1720
      IF (ENRFED.EQ.99.) GO TO 155                             A    1730
      READ (NIN,725)(XINIT(I),I=1,2),YIN(I),(XINIT(I),I=4,5)   A    1740
```

```
      READ (NIN,625)(YIN(I),I=1,3),YIN(6),(YIN(I),I=4,5)                A 1750
C
C     TRANSFORM POINTS IF CALLED FOR ( CHANGE = .TRUE. )
C
      IF(CHANGE) CALL XYTRAN(J,NSEG,ENREED,NS)
      IF (IAB.GT.0) WRITE (4,625)(XIN(I),I=1,3),XIN(6),(XIN(I),I=4,5)   A 1760
      IF (IAB.GT.0) WRITE (4,625)(YIN(I),I=1,3),YIN(6),(YIN(I),I=4,5)   A 1770
   90 KSV=K                                                             A 1780
      CAPPA(K)=0.0                                                      A 1790
      IF (ENREED.NE.1.0.AND.ENREED.NE.10.) GO TO 105                    A 1800
      XON(K)=XIN(1)                                                     A 1810
      YON(K)=YIN(1)                                                     A 1820
      WRITE (6,570)ENREEL,(XIN(I),I=1,2),(YIN(I),I=1,2)                 A 1830
                                                                        A 1890
      IF (ENREED.EQ.10..OR.(NUMBOD.GE.2.AND.J.EQ.1))CALL FRSTSH(K)
      IF (ENREED.EQ.1.0.AND.J.EQ.NSEG) CALL FNSTRH (K)                  A 1870
      IF(K.EQ.KK)                     CALL STRAIT (K,0.0)
   95 DYDXO(KSV)=DYDXO(KSV+1)                                           A 1920
      ALPHA(KSV)=ALPHA(KSV+1)                                           A 1930
      GO TO 160                                                         A 2020
                                                                        A 1900
  105 IF(ENREED.LT.-3.0)GOTO216
      IF (ENREED.LT.-2.0) GO TO 150
      IF (ENREED.LT.-1.0) GO TO 140                                     A 2080
                                                                        A 2090
      IF (ENREED.LT.0.0) GO TO 135                                      A 2100
C                                                                       A 2110
                                                                        A 2120
C     SET-UP SUPER ELLIPSE                                              A 2130
C                                                                       A 2140
                                    WRITE (6,85)ENREED
      KPREP=0                                                           A 2150
      ENRD=ENREED-1000.                                                 A 2160
      IF (ENRD.LT.0.0) GO TO 110                                        A 2170
      PACE=ENREED-1000.                                                 A 2180
      IF(PACE.LE.0.)PACE=.05                                            A 2190
      ENREED=0.                                                         A 2200
      IFLD=IFLD+1                                                       A 2210
      GO TO 115                                                         A 2220
  110 IFLD=0                                                            A 2230
  115 WRITE (6,575)REEDEN(1),(XIN(I),I=1,3),XIN(6),(XIN(I),I=4,5),REEDENA 2240
     1(2),(YIN(I),I=1,3),YIN(6),(YIN(I),I=4,5)                          A 2250
      CALL TEST (5)                                                     A 2260
      ISTART=K                                                          A 2270
      KI=K                                                              A 2280
      CALL SUPERC(XIN,YIN,REEDEN,DELSI,ISTART)                          A 2300
      K=IO-1                                                            A 2310
      IF (N2.EQ.1) GO TO 125                                            A 2320
      KKKK=K-1                                                          A 2330
      DO 120 JE=ISTART,KKKK                                             A 2340
      IF (XON(JE).EQ.XON(JE+1)) GO TO 120                               A 2350
      IF((DYDXO(JE)-DYDXO(JE+1))/(XON(JE)-XON(JE+1))*CAPPA(JE).LT.0.)CAPA A 2360
     1PA(JE)=-CAPPA(JE)                                                 A 2370
  120 CONTINUE                                                          A 2380
  125 K2=K                                                              A 2390
      IDUM=0                                                            A 2400
      IF (KPREP.EQ.0) GO TO 170
```

```
      CALL PRELPS (IDUM,...,K1,K2)                                      A 2420
      GO TO 160                                                         A 2480
C                                                                       A 2490
C     SET-UP LEMNISCATE                                                 A 2500
C                                                                       A 2510
  135 WRITE (6,560)ENREEL,(YIN(I),I=1,3),(YIN(I),I=1,3)                 A 2520
      CALL LEM (K)                                                      A 2540
      K=K+1                                                             A 2550
      GO TO 160                                                         A 2610
C                                                                       A 2620
C     SET-UP ELLIPSE                                                    A 2630
C                                                                       A 2640
  140 WRITE (6,585)ENREEL,(XIN(I),I=1,4),(YIN(I),I=1,4)                 A 2650
      KPREP=0                                                           A 2660
      CALL TEST (4)                                                     A 2670
      K1=K                                                              A 2680
      CALL ELIPSE (K)                                                   A 2700
      K=K+1                                                             A 2710
      K2=K                                                              A 2720
      IDUM=0                                                            A 2730
      IF (KPREP.EQ.0) GO TO 160
      CALL PRELPS (IDUM,1,4,K1,K2)                                      A 2750
      GO TO 160                                                         A 2810
C                                                                       A 2820
C     SET-UP CUBIC                                                      A 2830
C                                                                       A 2840
  150 WRITE (6,605)ENREEL,(XIN(I),I=1,4),(YIN(I),I=1,4)                 A 2850
      CALL CUBIC (K)                                                    A 2870
      K=K+1                                                             A 2880
      GO TO 160                                                         A 2940
C(((((( NEW LINEAR INTEGRATION OPTION, BODY OR FULL INLET               A 2950
 155  KSV=K                                                             A 2970
      CAPPA(K)=0.0                                                      A 2980
      IF(NSEG.EQ.0)DELS1=DELSMX                                         A 2990
      CALL XYCALC(K,K2,MIN)                                             A 3000
      K=K2                                                              A 3010
      GOTO160
 216  KTOT=0
      KBEGIN=0
      KSTOP=0
      DO 219 JBOP=1,K
      IF(XIN(1).EQ.XON(JBOP).AND.YIN(1).EQ.YON(JBOP))KBEGIN=JBOP
      IF(XIN(2).EQ.XON(JBOP).AND.YIN(2).EQ.YON(JBOP))KSTOP=JBOP
      IF(KSTOP*KBEGIN.NE.0)GOTO221
 219  CONTINUE
      WRITE(6,222)KBEGIN,KSTOP
 221  KTOT=KSTOP-KBEGIN+1
      YCL=YIN(3)
      WRITE(6,223)YCL,ENREEL,(XIN(L2),L2=1,2),(YIN(L2),L2=1,2)
      CALL MIRROR(K,KTOT,KBEGIN,YCL)
CC((((((( KR= TOTAL NO. OF POINTS TO BE PLOTTED FOR THIS SEGMENT        A 1940
 160  KR=K-KK                                                           A 1950
C(((((( PLOT CURRENT SEGMENT                                            A 1960
      CALL DRAW(KR,KK)                                                  A 1970
      IF (IAB.LE.0) GO TO 200
      IF(ABS(XON(KK)-XAA+YON(KK)-YAA).LE.1.E-7)JSTART=KK                A 2910
      IF(ABS(XON(K) -XBB+YON(K) -YBB).LE.1.E-7)JSTOP=K                  A 2920
```

```
      IF(JSTART.EQ.KK)MEL=N7
      IF ((JSTART+JSTOP).EQ.0.OR.(JSTOP.GT.0.AND.K.GT.JSTOP)) GO TO 200 A    2930
      DO 165 K9=1,7                                                     A    3050
  165 BACKSPACE 4                                                       A    3060
      IF (JSTART.NE.KK) GO TO 180                                       A    3070
      WRITE (4,170)                                                     A    3080
  170 FORMAT(3X,3H99.,74X)                                              A    3090
      WRITE (4,445)XON(KK),YON(KK)                                      A    3100
      KK1=KK+1                                                          A    3110
      DO 175 K9=KK1,K                                                   A    3120
  175 WRITE (4,450)XON(K9),YON(K9)                                      A    3130
      BACKSPACE 4                                                       A    3140
      IF (JSTOP.NE.K) GO TO 180                                         A    3150
      WRITE (4,455)XON(K),YON(K)                                        A    3160
      GO TO 200                                                         A    3170
  180 IF (JSTOP.NE.K) GO TO 190                                         A    3180
      LOWER=LOWER+1                                                     A    3190
      KM1=K-1                                                           A    3200
      DO 185 K9=KK,KM1                                                  A    3210
  185 WRITE (4,450)XON(K9),YON(K9)                                      A    3220
      WRITE (4,455)XON(K),YON(K)                                        A    3230
      GO TO 200                                                         A    3240
  190 IF (JSTART.EQ.KK.AND.JSTOP.EQ.0) GO TO 200                        A    3250
      LOWER=LOWER+1                                                     A    3260
      KM1=K-1                                                           A    3270
      DO 195 K9=KK,KM1                                                  A    3280
  195 WRITE (4,450)XON(K9),YON(K9)                                      A    3290
  200 WRITE (6,205)K,XON(K),YON(K),CAPPA(K),DYDXO(K),ALPHA(K)           A    3300
  205 FORMAT(1H0,18X,                                                   A    3310
     1                  11HLAST POINT ,2HK=,I5,4H, X=,E12.5,4H, Y=,E12.5,7H,A    332
     1KAPPA=,E12.5,7H,DY/DX=,E12.5,7H,ALPHA=, E12.5)
C                                                                       A    3330
C     END OF SEGMENT LOOP                                               A    3340
C                                                                       A    3350
210   GOTO220                                                           A    3360
 215  MIRBOD=IFIX(( ABS(TYPBDY)-FLOAT(NUMBOD))*10.1)
      YCL=ANSEG
      KTOT=NBDPTS(MIRBOD)-NBDPTS(MIRBOD-1)
      KBEGIN=NBDPTS(MIRBOD-1)+1
      IF(MIRBOD.EQ.1)KBEGIN=1
      IF(MIRBOD.EQ.1)KTOT=NBDPTS(1)
      WRITE(6,216)NUMBOD,MIRBOD,YCL
      KK=K
      CALL MIRROR(K,KTOT,KBEGIN,YCL)
      KR=K-KK
      CALL DRAW(KR,KK)
  220 NBDPTS(NZ)=K
      NBODY1=NBDPTS(1)                                                  A    3390
      NBODY2=NBDPTS(2)                                                  A    3400
      TYP(NZ)=TYPBDY                                                    A    3410
      KAY(NZ)=K                                                         A    3420
  225 CONTINUE                                                          A    3430
C                                                                       A    3440
C     END OF BODY LOOP                                                  A    3450
C                                                                       A    3460
      ITOP12=K                                                          A    3470
      IF(IFLAG.EQ.1)NBODY2=ITOP12                                       A    3480
                                                                        A    3490
```

```
      DELSND =DELS                                                      A 3500
      IF (IFLAG.EQ.1) NBDY1=0                                           A 3510
C9                                                                      A 3520
      ITOPT4=K+1
      ITOPT5=K+1
      ITOPT6=K+2
      ITOPT7=K+2
C                                                                       A 3530
C                                                                       A 3580
C     CO-ORDINATES OF POINTS ON DOWNSTREAM CLOSURE                      A 3540
C       (GENERATED FOR NON-VORTICITY CASES ONLY)
      IF(IVORT.EQ.1) GOTO 255
C                                                                       A 3560
C                                                                       A 3590
C     STRAIGHT SECTION BETWEEN HUB AND SHROUD OR SPLITTER               A 3570
      IF (NBDY1.EQ.0) GO TO 230                                         A 3600
      YNBDY1=YON(NBDY1)                                                 A 3610
      Y4SAVE=YON(NBDY1)                                                 A 3620
      Y5SAVE=YON(NBDY1+1)                                               A 3630
      Y6SAVE=YON(NBODY2)                                                A 3640
      Y7SAVE=YON(NBODY2+1)                                              A 3650
      GO TO 235                                                         A 3660
  230 YNBDY1=0.0                                                        A 3670
      Y4SAVE=0.0                                                        A 3680
  235 NDY4=(YON(NBDY1+1)-YNBDY1)*1.5/DELSMX                             A 3690
      ENDY4=NDY4                                                        A 3700
      NPTS=NDY4+1                                                       A 3710
      NBDPTS(NBDYS+1)=NPTS+NBDPTS(NBDYS)                                A 3720
      DY4=(YON(NBDY1+1)-YNBDY1)/ENDY4                                   A 3730
      DO 240 I=1,NPTS                                                   A 3740
      AYEM=I-1                                                          A 3750
      IPN=I+K                                                           A 3760
      XON(IPN)=XON(NBDY1+1)                                             A 3770
      YON(IPN)=YNBDY1+AYEM*DY4                                          A 3780
  240 CONTINUE                                                          A 3790
  245 ITOPT4=K+1                                                        A 3800
      ITOPT5=ITOPT4+NDY4                                                A 3810
      IF (NBDYS.LE.2) GO TO 255                                         A 3820
C                                                                       A 3830
C     STRAIGHT SECTION BETWEEN FLOW SPLITTER AND SHROUD                 A 3850
C                                                                       A 3870
      YNBDY2=Y6SAVE                                                     A 3880
      NDY5=(Y7SAVE-Y6SAVE)*1.5/DELSMX                                   A 3890
      ENDY5=NDY5                                                        A 3900
      NPTS=NDY5+1                                                       A 3910
      NBDPTS(NBDYS+2)=NPTS+NBDPTS(NBDYS+1)                              A 3920
      DY5=(Y7SAVE-Y6SAVE)/ENDY5                                         A 3930
      DO 250 I=1,NPTS                                                   A 3940
      AYEM=I-1                                                          A 3950
      IPN=I+ITOPT5                                                      A 3960
      XON(IPN)=XON(NBODY2+1)                                            A 3970
      YON(IPN)=YNBDY2+AYEM*DY5                                          A 3980
  250 CONTINUE                                                          A 3990
      ITOPT6=ITOPT5+1                                                   A 4000
      ITOPT7=ITOPT6+NDY5                                                A 4010
C                                                                       A 3860
C                                                                       A 4020
```

```
C                                                                A   3840
C           CALL SUBROUTINE TO WRITE AND PUNCH CARDS             A   4030
C                                                                A   4040
  255 CALL WPUNCH                                                A   4050
      IF (NBDYS.GT.2) GO TO 260                                  A   4060
      NT1=ITOPT5-3                                               A   4070
      NT2=M-4                                                    A   4080
      GO TO 265                                                  A   4090
 260  NT1=ITOPT7-NBDYS-(1-IVORT)*(NBDYS-1)-2*IVORT
      NT2=ITOPT5-NBDYS-(1-IVORT)*(NBDYS-2)-IVORT
  265 NHUPMX=NBDY1-1                                             A   4140
      NP=0                                                       A   4150
      DO 270 I=1,NRAKES                                          A   4160
      NP=NP+NDY(I)+1                                             A   4170
  270 CONTINUE                                                   A   4180
      NHBMXO=NHUBMX                                              A   4190
      IF(PROG.NE.O2TEST) GO TO 280
      DO 281 I=1,NBDYS
      NB(I)=NBDPTS(I)-I
  281 CONTINUE
      IF(NBDYS.EQ.2) WRITE(6,541) NB(1),NB(2),NP
      IF(NBDYS.EQ.3) WRITE(6,542) NB(1),NB(2),NB(3),NP
      GO TO 285
  280 WRITE (6,540)NT1,NT2,NHUBMX,NP                             A   4270
C                                                                A   4280
C                                                                A   4300
C     CALCULATION OF SURFACE DISTANCE ALONG EACH BODY FROM X=XRI
C                                                                A   4400
 285  DO 339 IBOD=1,NBDYS
      WRITE(6,505)   IBOD
      NBP1=NBDPTS(IBOD-1) +1
      IF(IBOD.EQ.1)NBP1=1
      NBP2=NBDPTS(IBOD)
      DO 325 I=NBP1,NBP2
      JJ=I                                                       A   4770
      IF (XON(I).LT.XON(I+1)) GO TO 330                          A   4780
  325 CONTINUE                                                   A   4790
 330  NSTRT=NBP1
      XSEOO=XRI
      NTOTT=JJ-NBP1+1
      IF(IBOD.EQ.1.AND.NBDYS.NE.1)NTOTT=NBP2-JJ+1
      IF(IBOD.EQ.1.AND.NBDYS.NE.1)NSTRT=JJ
      IF(XRI.LT.XON(JJ))XSEOO=XON(JJ)
      SDEL=0.0                                                   A   4820
C
C....  THE FOLLOWING LOOP IS A REDUNDANT CALCULATION OF THE SON ARRAY.
C... PURPOSE IS TO ELIMINATE A :JUMP: IN THE OUTPUT VALUES OF S, SON, AND SDEL.
C... THE :JUMP' HAS BEEN NOTICED AT THE FIRST POINT OF A BI-ELLIPSE WHEN
C... PRECEDED BY A STRAIGHT LINE. THE :JUMP' IN ARC LENGTH QUANTITIES WAS NOT
C... ACCOMPANIED BY A 'JUMP' IN XON,YON VALUES. THE 'JUMP  WAS APPROXIMATELY
C... EQUAL TO THE LENGTH OF THE BI-ELLIPSE.  (8/4/78)
      DO 334 I=NBP1,NBP2
      IF(I .GT. NBP1) GO TO 333
      SON(I) = 0.0
      GO TO 334
  333 ADELX2 = (ABS(XON(I) - XON(I-1)))**2
      ADELY2 = (ABS(YON(I) - YON(I-1)))**2
```

```
      SON(I) = SON(I-1) + SQRT(ADELX2 + ADELY2)
  334 CONTINUE
      CALL SINTP(XON(NSTRT),          SON(NSTRT),NTOTT,XSEQ0,S33)
      DO 335 I=NBP1,NBP2
      IF (I.NE.NBP1) SDEL=SON(I)-SON(I-1)                              A    4840
      S(I)=S33-SON(I)                                                  A    4850
  335 WRITE (6,520)I,XON(I),YON(I),CAPPA(I),DYDXO(I),ALPHA(I),SON(I),S(IA    4860
     1),SDEL                                                           A    4870
C                                                                      A    4880
C                                                                      A    4600
339   NBDSV=NBDYS
C                                                                      A    5010
C     WRITE OUT CLOSURE COORDINATES                                    A    5020
C                                                                      A    5030
C                                                                      A    4620
  350 IBD=NBDYS+1                                                      A    5040
      IF(IVORT.EQ.1) GOTO 355
      WRITE (6,515)IBD,(I,XON(I),YON(I),I=ITOPT4,ITOPT5)               A    5050
      IF (NBDSV.NE.3) GO TO 355                                        A    5060
      IBD=IBD+1                                                        A    5070
      WRITE (6,515)IBD,(I,XON(I),YON(I),I=ITOPT6,ITOPT7)               A    5080
  355 WRITE (6,545)(XRAK(I),YLO(I),YHI(I),NY(I),I=1,NRAKES)            A    5090
      IF(XON(1).GE.XON(3).OR.NBDYS.NE.1)
     1CALL AREAA
C                                                                      A    5110
C(((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((A    5120
C                                                                      A    5130
C((((((         CURVATURE PLOTS      ((((((((((((((((((((((((((((((((((A    5140
C                                                                      A    5150
      LEL=6                                                            A    5160
C(((((( IF IPLOTC IS NOT ZERO,  PLOT THE CURVATURE VS. S(IF IPLOTC .GT.0A    5170
C((((((                                        OR VS.X (IF IPLOTC .LT.0A    5180
      IF (IPLOTC.EQ.0) GO TO 405                                       A    5190
      IF(IPLOTC.LT.0) LEL=4                                            A    5200
      LE=0                                                             A    5210
C(((((( LEA = LAST PT. ON SHROUD                                       A    5220
      LEA=NBDPTS(NBDYS)
      LL=1                                                             A    5240
C(((((( TEST EACH SHROUD PT. FOR    1) IN RANGE OF X-AXIS?             A    5250
C((((((                             2) BEFORE OR AFTER HIGHLIGHT?      A    5260
C((((((                             3) FINITE CURVATURE?               A    5270
      NBDYS1=NBDPTS(NBDYS-1)
      IF(NBDYS.EQ.1)NBDYS1=0
360   LEE=LL+NBDYS1
      IF (XON(LE).GT.(XX*EXEP+XMIN)) GO TO 375                         A    5290
      IF (XON(LEE).LT.XON(LEE+1).AND.LEE.LT.LEEHI) GO TO 365           A    5300
      IF (CAPPA(LEE).EQ.99999.) GO TO 375                              A    5310
      LE=LE+1                                                          A    5320
      DIST(LE)= SON(LEE)                                               A    5330
      CAPPER(LE)= CAPPA(LEE)                                           A    5340
C(((((( FLAG THE HIGHLIGHT                                             A    5350
      IF (XON(LEE).GT.XON(LEE+1)) GO TO 370                            A    5360
  365 IF(LEE.LT.LEEHI) LEEHI=LEE                                       A    5370
C(((((( USE ABSOLUTE VALUE OF CURVATURE ON EXTERIOR OF SHROUD          A    5380
      IF(CAPPER(LE).LT.0.) CAPPER(LE)=-CAPPER(LE)                      A    5390
  370 IF(IPLOTC.LT.0) DIST(LE)=XON(LEE)                                A    5400
C(((((( IF CURVAT.-VS.-X PLOT IS NEEDED AND HIGHLIGHT HAS BEEN REACHED, A    5410
```

```
C(((((( PLOT THE FIRST FRAME (INTERNAL SHROUD PTS.)
        IF (IPLOTC.LT.0.AND.LEE.EQ.LEEHI) GO TO 380
  375 LL=LL+1
        IF (LEE.LE.LEA                        ) GO TO 360
  380 CALL PLOT(XX,0.,-2)
C(((((( IF THE SECOND X-CURVE (EXTERNAL PTS.) IS BEING PLOTTED, DO NOT
C(((((( GENERATE NEW SCALE FACTORS. USE THOSE OF THE INTERNAL PLOT.
        IF (LEE.GT.LEEHI.AND.IPLOTC.LT.0) GO TO 385
        CALL CSCALE(CAPPER,YY,LE,1,10,EXMIN,DEEX)
  385 CAPPER(LE+1)=EXMIN
        CAPPER(LE+2)=DEEX
        IF (LEE.GT.LEEHI.AND.IPLOTC.LT.0) GO TO 390
        CALL CSCALE(DIST ,YY,LE,1,10,EXMIND,DEEXD)
  390 DIST(LE+1)=EXMIND
        DIST(LE+2)=DEEXD
        IF(IPLOTC.LT.0) DIST(LE+1)= XMIN
        IF(IPLOTC.LT.0) DIST(LE+2)= EXEP
C((((((      DRAW AXES FOR CURVATURE PLOT
        CALL PLOXIS(XX,YY,DIST(LE+2),CAPPER(LE+2),DIST(LE+1),CAPPER(LE+1),
       1.25,.25,0,0,LEL,5,1,2)
        CALL LINE(DIST,CAPPER,LE,1,1,3,DIST(LE+1),DIST(LE+2),CAPPER(LE+1),
       1CAPPER(LE+2))
C((((((      DRAW SEGMENT MARKERS
        DO 400 MEM=1,MM
        IF(IPLOTC.LT.0) BAGS(MEM)=BAGX(MEM)
        IF(ZAP(MEM).LT.0..AND.NZAP(MEM).GT.LEEHI) ZAP(MEM)=-ZAP(MEM)
        IF(ZAP(MEM).EQ.99999.) ZAP(MEM)=YY*CAPPER(LE+2) + CAPPER(LE+1)
        IF (LEE.GT.LEEHI.AND.IPLOTC.LT.0) GO TO 395
        BAGS(MEM)=(BAGS(MEM)-DIST(LE+1))/DIST(LE+2)
        ZAP(MEM)=(ZAP(MEM)-CAPPER(LE+1)) /CAPPER(LE+2)
        IF(IPLOTC.GT.0.AND. NZAP(MEM).LT.LEEHI) CALL SYMBOL(BAGS(MEM),ZAP(
       1MEM),.2,1,0.,-1)
  395 IF (IPLOTC.LT.0.AND.BAGS(MEM).GT.(XX+EXEP+XMIN)) GO TO 400
        IF((NZAP(MEM).LE.LEEHI.AND.LEE.EQ.LEEHI).OR.(NZAP(MEM).GE.LEEHI.AN
       1D.LEE.NE.LEEHI)) CALL SYMBOL(BAGS(MEM),ZAP(MEM),.2,1,0.,-1)
  400 CONTINUE
        IF (LEE.GT.LEA                        .OR.LEE.GT.LEEHI.OR.XON(LEE).EQ.XON(
       1LEA)) GO TO 405
        LE=0
        CALL SYMBOL(XX-.5,YY-.5,.3,52,0.,-1)
        GO TO 375
C
C((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C
C(((((( ADVANCE THE PLOT ORIGIN FOR THE NEXT CASE
  405 CALL PLOT(XX,0.,-3)
C
C(((((( IF THE CASE IS NOT TO BE REWORKED VIA FESSLER, BEGIN NEXT JOB
        IF (IAB.LE.0) GO TO 410
        REWIND 4
        NIN=4
        DELS =DSAVE
        GO TO 20
  410 IF (IREDON(1).EQ.1) GO TO 10
        REWIND 4
        LPNCHO= IREDON(2)
        IPLOTA= IREDON(3)
```

```
      IPLOTC= IREDON(4)                                                 A    5980
      IREAD = IREDON(5)                                                 A    5990
      IREDON(I)=  IREDON(I)- 1                                          A    6000
C(((((( WRITE THE ORIGINAL CASE OUTPUT ON UNIT 4 FOR ACCEPTANCE AS INPUTA    6010
C   BY DIRECT INTERPOLATION OPTION(XYCALC,FESSLER)                      A    6020
      WRITE (4,475)IDENT,PROG,NO6,LPNCHO,IPLOTA,IPLOTC,IAR,(IREDON(I),I=A    6030
     71,4)                                                              A    6040
      DO 425 I=1,NBDYS                                                  A    6050
      IF(NBDYS.EQ.2.AND.I.EQ.2) TYP(I)=2.                               A    6060
      IF (I.EQ.1) GO TO 415                                             A    6070
      KA= KAY(I-1)+2                                                    A    6080
      GO TO 420                                                         A    6090
  415 KA=2                                                              A    6100
  420 WRITE (4,440)TYP(I)                                               A    6110
      KB= KAY(I)-1                                                      A    6120
      KAB=KA-1                                                          A    6130
      WRITE (4,445)XON(KAB),YON(KAB)                                    A    6140
      WRITE (4,450)(XON(J),YON(J),J=KA,KB)                              A    6150
      KC=KAY(I)                                                         A    6160
      WRITE (4,455)XON(KC),YON(KC)                                      A    6170
  425 CONTINUE                                                          A    6180
  435 NIN=4                                                             A    6320
      REWIND 4                                                          A    6330
      GO TO 20                                                          A    6340
  440 FORMAT(F10.2,10H    0.       )                                    A    6350
  445 FORMAT(1X,11H$BODYIN Z=(,F10.6,1H,,F10.6,4H),   )                 A    6360
  450 FORMAT(12X,1H(,F10.6,1H,,F10.6,3H), )                             A    6370
  455 FORMAT(12X,1H(,F10.6,1H,,F10.6,3H) $/1X,20H$AUXIN DONE=.TRUE. $)  A    6380
  460 FORMAT(8F10.2)                                                    A    6390
  465 FORMAT(3X,7H-1      /1X,11H$BODYIN Z=(,F10.6,1H,,F10.6,4H),   )   A    6400
  470 FORMAT(1H+,13X,6HREDONE)                                          A    6410
  475 FORMAT(          2A6,I4,2I1, I2,12X,I1,9X,3I1, I2)                A    6420
C                                                                       A    6430
C     FORMATS                                                           A    6440
C                                                                       A    6450
C                                                                       A    6460
  480 FORMAT (1H0,10X,16HNO. OF BODIES = ,F2.0,5X,7HDELS = ,F6.3,5X,9HDEA    6470
     1LSMX = ,F6.3,5X,6HXRI = ,F10.6)                                   A    6480
  485 FORMAT (2014)                                                     A    6490
  490 FORMAT(          2A6,I4,2I1,2I2,10X,I1,9X,3I1,2I2)                A    6500
  495 FORMAT(          2A6,I4,2I1, I2,12X,1H0,9X,3I1, I2)               A    6510
  500 FORMAT (9A6)                                                      A    6520
  218  FORMAT('0',3X,
     1           'BODY NO. ',I4,' WILL BE GENERATED AS A MIRROR IMAGE OF ',
     1'BODY NO.',I4,' ABOUT AN AXIS AT Y= ',E15.4)
  222  FORMAT('USEARCH FOR SEGMENT TO BE MIRRORED HAS FAILED',2I4)
  223  FORMAT(1H0,7X,18H---- MIRROR IMAGE /
     1           25X,'ENREED   THIS SEGMENT IS A MIRROR IMAGE ABOUT THE'
     1,' Y= ',1P1E15.4,'LINE'/35X,' THE ORIGINAL SEGMENT LIES BETWEEN',
     2' THE FOLLOWING POINTS'/26X,F6.3,5X, 1HX,1P2E15.4/37X,1HY,1P2E15.4)
   505 FORMAT(1X/' BODY ',I2,' CO-ORDINATES- X',I2X1HY10X,5HKAPPA10X5HDY/
      10X10X
      2  5HALPHA5X1HS8X6HS-S(2),8X,6HDELTAS/1X)
  626  FORMAT(8H0  BODY ,I2,' ',19('*******'))
  631  FORMAT('0GEOMETRY OUTPUT IS SAVED ON UNIT 17 FOR')
  632  FORMAT('0GEOMETRY OUTPUT IS NOT SAVED FOR')
  633  FORMAT(' PROGRAM ',A6,' WHICH WILL GENERATE')
```

```
  634 FORMAT(' HIGHER ORDER SOLUTION')
  635 FORMAT('  LOWER ORDER SOLUTION')
  636 FORMAT(' WITH STRIP VORTEX FLOW.')
  637 FORMAT(' WITH AXISYMMETRIC AND CROSSFLOWS ONLY.')
  638 FORMAT('OSCIRCLE WILL PRODUCE')
  639 FORMAT(' CROSSECTIONAL AREA PLOTS,')
  640 FORMAT(' NO CROSSECTIONAL AREA PLOTS.')
  641 FORMAT(' NO CURVATURE PLOTS,')
  642 FORMAT(    ' CURVATURE PLOTS,')
  643 FORMAT(' A REWORKED INPUT AND GEOMETRY (TWO PASSES).')
  644 FORMAT(' NO REWORKING OF GEOMETRY.')
   85 FORMAT(1HO,7X,18H---- SUPERELLIPSE,
     1         9H ENREED= ,F9.2)
  515 FORMAT (1X/1X5HBODY I1,17H CO-ORDINATES - X12X1HY/1X/(9XI4,3XE12.5A   6570
     1,E13.5))                                                          A   6580
  520 FORMAT (9XI4,3XE12.5,7E13.5)                                      A   6590
  525 FORMAT (6HICASE A6,10X,9A6/1X)                                    A   6600
  530 FORMAT (1HO)                                                      A   6610
  540 FORMAT (/10X,30H INPUT FOR THE COMBINE PROGRAM,7H NT(1)=,I4,7H NT(A   6640
     12)=,I4,8H NHUBMX=,I4,4H NP=,I4/)                                  A   6650
  541 FORMAT (/10X,30H INPUT FOR THE COMBINE PROGRAM,7H NT(1)=,I4,7H NT(A   6640
     12)=,I4,4H NP=,I4/)                                                A   6650
  542 FORMAT (/10X,30H INPUT FOR THE COMBINE PROGRAM,7H NT(1)=,I4,7H NT(A   6640
     12)=,I4,' NT(3)=',I4,4H NP=,I4/)                                   A   6650
  545 FORMAT (1X,4HXRAK,10X,3HYLO,11X,3HYHI,16X,3HNDY//,(3E14.5,5X,I3)) A   6660
  550 FORMAT(3E8.5,I3,5F8.0)                                            A   6670
  555 FORMAT(8F10.2)
  558 FORMAT(4F10.0,5F8.L)
  560 FORMAT( F10.2,10F7.4/F7.5,F7.2)                                   A   6690
  565 FORMAT (/5X,30H INPUT FOR THE COMBINE PROGRAM,7H NT(1)=,I4,7H NT(2A   67C0
     1)=,I4,7H NT(3)=,I4,8H NHUBMX=,I4,8H NSPLMX=,I4,4H NP=,I4/)         A   6710
  570 FORMAT (1HO,7X,18H---- STRAIGHT LINE/25X,6HENREED/
     2                                              26X,F6.3,5X,1HX,1P2EA   6720
     115.4/37X,1HY,1P2E15.4)
  575 FORMAT(1H ,19X,9HEXPONENTS/
     1                                              19X,4HP = ,F6.3,5X,1HA   6740
     1X,1P6E15.4/19X,
     1                 4HQ = ,0PF6.3,5X,1HY,1P6E15.4)
  580 FORMAT (1HO,7X,18H---- LEMNISCATE    /25X,6HENREED/
     1                                              26X,F6.3,5X,1HX,1P3E15.A   6760
     14/37X,1HY,1P3E15.4)
  585 FORMAT (1HO,7X,18H---- ELLIPSE      /25X,6HENREED/
     1                                     26X,F6.3,5X,1HX,1P4E15.4/37A   67
     1X,1HY,1P4E15.4)                                                  A   6790
  605 FORMAT (1HO,7X,18H---- CUBIC        /25X,6HENREED/
     1                                     26X,F6.3,5X,1HX,1P4E15.4/37X,A   6830
     11HY,1P4E15.4)                                                    A   6840
  615 FORMAT (9X,I4,3X,F12.5,5E13.5)                                   A   6870
  620 FORMAT(2F10.7)                                                   A   6880
  625 FORMAT(6F12.5)                                                   A   6890
  630 CALL PLOT(3.,C.,-3)                                              A   6900
      CALL PLOTID                                                      A   6910
      STOP                                                             A   6920
      END                                                              A   6930
```

```
C     ..................................................................
C
C        SUBROUTINE SIMQ
C
C        PURPOSE
C           OBTAIN SOLUTION OF A SET OF SIMULTANEOUS LINEAR EQUATIONS,
C           AX=B
C        USAGE
C           CALL SIMQ(A,B,N,KS)
C
C        DESCRIPTION OF PARAMETERS
C           A - MATRIX OF COEFFICIENTS STORED COLUMNWISE.  THESE ARE
C               DESTROYED IN THE COMPUTATION.  THE SIZE OF MATRIX A IS
C               N BY N.
C           B - VECTOR OF ORIGINAL CONSTANTS (LENGTH N). THESE ARE
C               REPLACED BY FINAL SOLUTION VALUES, VECTOR X.
C           N - NUMBER OF EQUATIONS AND VARIABLES. N MUST BE .GT. ONE.
C           KS - OUTPUT DIGIT
C                0 FOR A NORMAL SOLUTION
C                1 FOR A SINGULAR SET OF EQUATIONS
C
C        REMARKS
C           MATRIX A MUST BE GENERAL.
C           IF MATRIX IS SINGULAR , SOLUTION VALUES ARE MEANINGLESS.
C           AN ALTERNATIVE SOLUTION MAY BE OBTAINED BY USING MATRIX
C           INVERSION (MINV) AND MATRIX PRODUCT (GMPRD).
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           METHOD OF SOLUTION IS BY ELIMINATION USING LARGEST PIVOTAL
C           DIVISOR. EACH STAGE OF ELIMINATION CONSISTS OF INTERCHANGING
C           ROWS WHEN NECESSARY TO AVOID DIVISION BY ZERO OR SMALL
C           ELEMENTS.
C           THE FORWARD SOLUTION TO OBTAIN VARIABLE N IS DONE IN
C           N STAGES. THE BACK SOLUTION FOR THE OTHER VARIABLES IS
C           CALCULATED BY SUCCESSIVE SUBSTITUTIONS. FINAL SOLUTION
C           VALUES ARE DEVELOPED IN VECTOR B, WITH VARIABLE 1 IN B(1),
C           VARIABLE 2 IN B(2),........, VARIABLE N IN B(N).
C           IF NO PIVOT CAN BE FOUND EXCEEDING A TOLERANCE OF 0.0,
C           THE MATRIX IS CONSIDERED SINGULAR AND KS IS SET TO 1. THIS
C           TOLERANCE CAN BE MODIFIED BY REPLACING THE FIRST STATEMENT.
C
C     ..................................................................
C
      SUBROUTINE SIMQ (A,B,N,KS)
      DIMENSION A(1), B(1)
C
C        FORWARD SOLUTION
C
      TOL=0.0
      KS=0
      JJ=-N
      DO 65 J=1,N
      JY=J+1
```

```
      JJ=JJ+N+1
      BIGA=0                                                            K 0560
      IT=JJ-J                                                           K 0570
      DO 15 I=J,N                                                       K 0580
C                                                                       K 0590
C        SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN                       K 0600
C                                                                       K 0610
      IJ=IT+I                                                           K 0620
      IF (ABS(BIGA)-ABS(A(IJ))) 10,15,15                                K 0630
   10 BIGA=A(IJ)                                                        K 0640
      IMAX=I                                                            K 0650
   15 CONTINUE                                                          K 0660
C                                                                       K 0670
C        TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)           K 0680
C                                                                       K 0690
      IF (ABS(BIGA)-TOL) 20,20,25                                       K 0700
   20 KS=1                                                              K 0710
      RETURN                                                            K 0720
C                                                                       K 0730
C        INTERCHANGE ROWS IF NECESSARY                                  K 0740
C                                                                       K 0750
   25 I1=J+N*(J-2)                                                      K 0760
      IT=IMAX-J                                                         K 0770
      DO 30 K=J,N                                                       K 0780
      I1=I1+N                                                           K 0790
      I2=I1+IT                                                          K 0800
      SAVE=A(I1)                                                        K 0810
      A(I1)=A(I2)                                                       K 0820
      A(I2)=SAVE                                                        K 0830
C                                                                       K 0840
C        DIVIDE EQUATION BY LEADING COEFFICIENT                         K 0850
C                                                                       K 0860
   30 A(I1)=A(I1)/BIGA                                                  K 0870
      SAVE=B(IMAX)                                                      K 0880
      B(IMAX)=B(J)                                                      K 0890
      B(J)=SAVE/BIGA                                                    K 0900
C                                                                       K 0910
C        ELIMINATE NEXT VARIABLE                                        K 0920
C                                                                       K 0930
      IF (J-N) 35,50,35                                                 K 0940
   35 IQS=N*(J-1)                                                       K 0950
      DO 45 IX=JY,N                                                     K 0960
      IXJ=IQS+IX                                                        K 0970
      IT=J-IX                                                           K 0980
      DO 40 JX=JY,N                                                     K 0990
      IXJX=N*(JX-1)+IX                                                  K 1000
      JJX=IXJX+IT                                                       K 1010
   40 A(IXJX)=A(IXJX)-(A(IXJ)*A(JJX))                                   K 1020
   45 B(IX)=B(IX)-(B(J)*A(IXJ))                                         K 1030
C                                                                       K 1040
C        BACK SOLUTION                                                  K 1050
C                                                                       K 1060
   50 NY=N-1                                                            K 1070
      IT=N*N                                                            K 1080
      DO 55 J=1,NY                                                      K 1090
      IA=IT-J                                                           K 1100
      IB=N-J                                                            K 1110
      IC=N                                                              K 1120
      DO 55 K=1,J                                                       K 1130
      B(IB)=B(IB)-A(IA)*B(IC)                                           K 1140
      IA=IA-N                                                           K 1150
   55 IC=IC-1                                                           K 1160
      RETURN                                                            K 1170
      END                                                               K 1180
                                                                        K 1190
```

```
      SUBROUTINE STRAIT (K,ISHR)                                        B    0000
C                                                                       B    0010
C      A REGULAR STRAIGHT SEGMENT                                       B    0020
C                                                                       B    0030
      COMMON /MAIN/ XIN(10),YIN(10),DELSMX,PIO2,DELS1,IHUB              B    0040
      COMMON /FOR3SS/IO,DELS,XBK(20),YBK(20),XON(500),YON(500),DYDXO(500B    0050
     1),ALPHA(500),CAPPA(500),SON(500),PIO180                           B    0060
      COMMON /SS/ NBDY1,NBODY2,TYPBDY,NPDYS                             B    0070
      COMMON /FNST/ NFIRST                                              B    0080
      COMMON /NHIGH/ NSPHG,NLAST,XLAST(500)                             B    0090
      KFIRST=K                                                          B    0100
      XTEST=XIN(2)-XIN(1)                                               B    0110
      YTEST=YIN(2)-YIN(1)                                               B    0120
      IF (XTEST.EQ.0.0) GO TO 10                                        B    0130
      IF (XTEST.EQ.0.0) GO TO 10                                        B    0130
      DYDXC=YTEST/XTEST                                                 B    0140
      ALPHAC=ATAN(YTEST/XTEST)                                          B    0150
      GO TO 15                                                          B    0160
   10 DYDXC=99999.                                                      B    0170
      ALPHAC=PIO2                                                       B    0180
C                                                                       B    0190
C     CALCULATE DELSNW                                                  B    0200
C                                                                       B    0210
   15 STOT=SQRT(XTEST**2+YTEST**2)                                      B    0220
      ANDS=STOT/DELS1                                                   B    0230
      AINDS=AINT(ANDS)                                                  B    0240
      TEST=ANDS-AINDS                                                   B    0250
      IF (TEST.GE..5) AINDS=AINDS+1.0                                   B    0260
      DELSNW=STOT/AINDS                                                 B    0270
      DELSNW=ABS(DELSNW)                                                B    0280
      DELS1=DELSNW                                                      B    0290
      DELS=DELS1                                                        B    0300
      IF (YTEST) 20,35,20                                               B    0310
   20 IF (XTEST.EQ.0.0) GO TO 50                                        B    0320
      DYDXO(K+1)=DYDXC                                                  B    0330
      ALPHA(K+1)=ALPHAC                                                 B    0340
      SIGN=1.0                                                          B    0350
      IF (XTEST.LT.0.0) SIGN=-1.0                                       B    0360
      YON(K+1)=YON(K)+SIGN*DELSNW*SIN(ALPHA(K+1))                       B    0370
      IF (NSPHG.EQ.0) GO TO 25                                          B    0380
      XON(K+1)=XON(NLAST-1)                                             B    0390
      NLAST=NLAST-1                                                     B    0400
      GO TO 30                                                          B    0410
   25 XON(K+1)=XON(K)+SIGN*DELSNW*COS(ALPHA(K+1))                       B    0420
   30 SON(K+1)=SON(K)+SQRT((XON(K+1)-XON(K))**2+(YON(K+1)-YON(K))**2)   B    0430
      CAPPA(K+1)=0.0                                                    B    0440
      IF (XTEST.LT.0.0.AND.XON(K+1).LE.XIN(2).OR.XTEST.GT.0.0.AND.XON(K+B    0450
     1).GE.XIN(2)) GO TO 55                                             B    0460
      IF (ABS(XON(K+1)-XIN(2)).LE.1.0E-4*DELS.AND.ABS(YON(K+1)-YIN(2)).LB    0470
     1E.1.0E-4*DELS) GO TO 60                                           B    0480
      K=K+1                                                             B    0490
      GO TO 20                                                          B    0500
   35 DYDXO(K+1)=0.0                                                    B    0510
      ALPHA(K+1)=0.0                                                    B    0520
      SIGN=1.0                                                          B    0530
      IF (XTEST.LT.0.0) SIGN=-1.0                                       B    0540
```

```
      IF (NSPH6.EQ.0) GO TO 40                                          B   0550
      XON(K+1)=XON(NLAST-1)                                             B   0560
      NLAST=NLAST-1                                                     B   0570
      GO TO 45                                                          B   0580
   40 XON(K+1)=XON(K)+SIGN*DELSNW                                       B   0590
   45 YON(K+1)=YON(K)                                                   B   0600
      SON(K+1)=SON(K)+SQRT((XON(K+1)-XON(K))**2+(YON(K+1)-YON(K))**2)   B   0610
      CAPPA(K+1)=0.0                                                    B   0620
      IF (XTEST.LT.0.0.AND.XON(K+1).LE.XIN(2).OR.XTEST.GT.0.0.AND.XON(K+B   0630
     11).GE.XIN(2)) GO TO 55                                            B   0640
      IF (ABS(XON(K+1)-XIN(2)).LE.1.0E-4*DELS.AND.ABS(YON(K+1)-YIN(2)).LB   0650
     1E.1.0E-4*DELS) GO TO 60                                           B   0660
      K=K+1                                                             B   0670
      GO TO 35                                                          B   0680
   50 DYDXO(K+1)=99999.                                                 B   0690
      ALPHA(K+1)=PIO2                                                   B   0700
      SIGN=1.0                                                          B   0710
      IF (YTEST.LT.0.0) SIGN=-1.0                                       B   0720
      XON(K+1)=XON(K)                                                   B   0730
      YON(K+1)=YON(K)+SIGN*DELSNW                                       B   0740
      SON(K+1)=SON(K)+SQRT((XON(K+1)-XON(K))**2+(YON(K+1)-YON(K))**2)   B   0750
      CAPPA(K+1)=0.0                                                    B   0760
      IF (YTEST.LT.0.0.AND.YON(K+1).LE.YIN(2).OR.YTEST.GT.0.0.AND.YON(K+B   0770
     11).GE.YIN(2)) GO TO 55                                            B   0780
      IF (ABS(XON(K+1)-XIN(2)).LE.1.0E-4*DELS.AND.ABS(YON(K+1)-YIN(2)).LB   0790
     1E.1.0E-4*DELS) GO TO 60                                           B   0800
      K=K+1                                                             B   0810
      GO TO 50                                                          B   0820
   55 IF (ABS(XON(K+1)-XIN(2)).LE.1.0E-3*DELS.AND.ABS(YON(K+1)-YIN(2)).LB   0830
     1E.1.0E-3*DELS) GO TO 60                                           B   0840
      K=K-1                                                             B   0850
   60 K=K+1                                                             B   0860
      DO 65 KAL=KFIRST,K                                                B   0870
   65 ALPHA(KAL)=ALPHA(KAL)/PIO180                                      B   0880
      RETURN                                                            B   0890
      END                                                               B   0900
```

```
      SUBROUTINE FNSTRH (K)                                            C   0000
C                                                                      C   0010
C     FINAL STRAIGHT SEGMENT ON THE HUB AND SHROUD                     C   0020
C                                                                      C   0030
      COMMON /MAIN/ XIN(10),YIN(10),DELSMX,PIO2,DELS1,INLB             C   0040
      COMMON /FOR3SS/IO,DELS,XBK(20),YBK(20),XON(500),YON(500),DYDXO(500C   0050
     1),ALPHA(500),CAPPA(500),SON(500),PIO180                          C   0060
      COMMON /SS/ NBDY1,NBODY2,TYPBDY,NPDYS                            C   0070
      COMMON /FNST/ NFIRST                                             C   0080
      NFIRST=K                                                         C   0090
      DS=DELS1                                                         C   0100
      DELSTR=DELSMX                                                    C   0110
      YTEST=YIN(2)-YIN(1)                                              C   0120
      XTEST=XIN(2)-XIN(1)                                              C   0130
      ASIGN=1.0                                                        C   0140
      IF (XTEST.LT.0.0) ASIGN=-1.0                                     C   0150
      ISTAR=0                                                          C   0160
      SSEG=SQRT(XTEST**2+YTEST**2)                                     C   0170
      IF (XTEST.EQ.0.0) GO TO 10                                       C   0180
      IF (YTEST.EQ.0.0) GO TO 15                                       C   0190
      DYDXC=YTEST/XTEST                                                C   0200
      ALPHAC=ATAN(YTEST/XTEST)                                         C   0210
      SINAL=SIN(ALPHAC)                                                C   0220
      COSAL=COS(ALPHAC)                                                C   0230
      GO TO 20                                                         C   0240
   10 DYDXC=SIGN(99999.,YTEST)                                         C   0250
      ALPHAC=SIGN(PIO2,YTEST)                                          C   0260
      SINAL=SIN(ALPHAC)                                                C   0270
      COSAL=0.0                                                        C   0280
      GO TO 20                                                         C   0290
   15 DYDXC=0.0                                                        C   0300
      ALPHAC=0.0                                                       C   0310
      SINAL=0.0                                                        C   0320
      COSAL=1.0                                                        C   0330
   20 DYDXO(K+1)=DYDXC                                                 C   0340
      ALPHA(K+1)=ALPHAC                                                C   0350
      IF (DS.GT.DELSMX) GO TO 25                                       C   0360
      GO TO 45                                                         C   0370
   25 IF (ISTAR.NE.0) GO TO 45                                         C   0380
      DSLAST=DS                                                        C   0390
      XON(K+1)=XON(K)                                                  C   0400
      YON(K+1)=YON(K)                                                  C   0410
      ICOUNT=0                                                         C   0420
   30 XSAVE=XON(K+1)-XIN(1)                                            C   0430
      YSAVE=YON(K+1)-YIN(1)                                            C   0440
      SSTAR=SQRT(XSAVE**2+YSAVE**2)                                    C   0450
      ASTAR=(SSEG-SSTAR)/DELSTR                                        C   0460
      ATEST=ASTAR-FLOAT(IFIX(ASTAR))                                   C   0470
      IF (ATEST.GT..5) ASTAR=ASTAR+1.0                                 C   0480
      NSTAR=IFIX(ASTAR)                                                C   0490
      ISTAR=1                                                          C   0500
      IF (NSTAR.EQ.0) GO TO 35                                         C   0510
      DS=(SSEG-SSTAR)/FLOAT(NSTAR)                                     C   0520
      IF (DS.GT.DSLAST) GO TO 35                                       C   0530
      IF (ICOUNT.[illegible]) K [illegible]                            C   0540
      GO TO 45                                                         C   0550
```

```
   35 K=K-1                                                           C  0560
      IF (K.GT.NFIRST) GO TO 40                                       C  0570
      K=NFIRST                                                        C  0580
      CALL STRAIT (K,0)                                               C  0590
      K=K-1                                                           C  0600
      GO TO 50                                                        C  0610
   40 DSLAST=SQRT((XON(K-1)-XON(K))**2+(YON(K-1)-YON(K))**2)*1.2      C  0620
      DELSTR=DSLAST                                                   C  0630
      ICOUNT=ICOUNT+1                                                 C  0640
      GO TO 30                                                        C  0650
   45 XON(K+1)=XON(K)+ASIGN*DS*COSAL                                  C  0660
      YON(K+1)=YON(K)+ASIGN*DS*SINAL                                  C  0670
      SON(K+1)=SON(K)+SQRT((XON(K+1)-XON(K))**2+(YON(K+1)-YON(K))**2) C  0680
      CAPPA(K+1)=0.0                                                  C  0690
      IF (ABS(XON(K+1)-XIN(2)).LE..001*DS.AND.XTEST.NE.0.0) GO TO 50  C  0700
      IF (ABS(YON(K+1)-YIN(2)).LE..001*DS.AND.XTEST.EQ.0.0) GO TO 50  C  0710
C                                                                     C  0720
C((((((   IS THE CURRENT POINT PAST THE NEAREST ENDPOINT OF SEGMENT? (PRC  0730
C(((((( VIOUS TESTS WERE ONLY FOR ABSOLUTE PROXIMITY TO ENDPOINT)     C  0740
      IF (ABS(YON(K+1)-YIN(1)).GT.ABS(YTEST))GOTO50
      IF (ABS(XON(K+1)-XIN(1)).GT.ABS(XTEST))GOTO50
      K=K+1                                                           C  0770
      IF (ISTAR.EQ.0) DS=DS*1.2                                       C  0780
      GO TO 20                                                        C  0790
   50 DELSI=DELS                                                      C  0800
      XON(K+1)=XIN(2)                                                 C  0810
      YON(K+1)=YIN(2)                                                 C  0820
      NBDY1=K+1                                                       C  0830
      K=K+1                                                           C  0840
      DO 55 KAL=NFIRST,K                                              C  0850
   55 ALPHA(KAL)=ALPHA(KAL)/PIO180                                    C  0860
      RETURN                                                          C  0870
      END                                                             C  0880
```

```
      SUBROUTINE FRSTSH (K)                                             D   0000
C                                                                       D   0010
C     FIRST STRAIGHT SEGMENT SHROUD                                     D   0020
C                                                                       D   0030
C     IF THERE IS NO HUB INTERCHANGE PCINTS (X1,Y1) AND (X2,Y2)         D   0040
C            AND TREAT LIKE FINAL STRAIGHT SECTION ON THE HUB,          D   0050
C            THEN REVERSE XON AND YON ARRAYS                            D   0060
C                                                                       D   0070
C                                                                       D   0080
      COMMON /MAIN/ XIN(10),YIN(10),DELSMX,PIO2,DELSI,IHUB,DELNEW
      COMMON /FOR3SS/I0,CELS,XBK(20),YBK(20),XON(500),YON(500),DYDXO(5000   0100
     1),ALPHA(500),CAPPA(500),SON(500),PIO180                           D   0110
      COMMON /SS/ NBODY1,NBODY2,TYPBDY,NBDYS                            D   0120
      COMMON /FNST/ NFIRST                                              D   0130
      DIMENSION XA(2), YA(2), DSV(500), ASV(500), XSV(500), YSV(500)    D   0140
      SON(K)=0.0                                                        D   0160
      IF (IHUB.EQ.1.AND.DELNEW.EQ.0.) GOTO25
      DO 10 I=1,2                                                       D   0180
      XA(I)=XIN(I)                                                      D   0190
   10 YA(I)=YIN(I)                                                      D   0200
      XIN(1)=XA(2)                                                      D   0210
      XIN(2)=XA(1)                                                      D   0220
      YIN(1)=YA(2)                                                      D   0230
      YIN(2)=YA(1)                                                      D   0240
      NFB2=K                                                            D   0250
      YON(K)=YIN(1)                                                     D   0260
      XON(K)=XIN(1)                                                     D   0270
      CALL FNSTRH (K)                                                   D   0280
      KSV=K                                                             D   0290
      KTOT=KSV-NFB2+1
      DO 15 II=1,KTOT
      I=NFB2+II-1
      KSR=KSV+1-II
      DSV(KSR)=DYDXO(I)                                                 D   0320
      ASV(KSR)=ALPHA(I)                                                 D   0330
      XSV(KSR)=XON(I)                                                   D   0340
   15 YSV(KSR)=YON(I)                                                   D   0350
      SON(NFB2) = 0.0
      DO 20 I=NFB2,KSV                                                  D   0370
      DYDXO(I)=DSV(I)                                                   D   0380
      ALPHA(I)=ASV(I)                                                   D   0390
      XON(I)=XSV(I)                                                     D   0400
      YON(I)=YSV(I)                                                     D   0430
      CAPPA(I)=0.0                                                      D   0430
   20 CONTINUE                                                          D   0440
      III=NFB2+1
      DO 21 I=III,KSV
      SON(I) = SON(I-1)+SQRT((XON(I)-XON(I-1))**2+(YON(I)-YON(I-1))**2)
   21 CONTINUE
      DELSI=ABS(SON(KSV)-SON(KSV-1))                                    D   0450
      RETURN                                                            D   0470
C                                                                       D   0480
C     IF THERE IS A HUB, USE X VALUES FROM FINAL STRAIGHT               D   0490
C         SECTION ON THE HUB FOR FIRST STRAIGHT SECTION ON              D   0500
C           SHROUD                                                      D   0510
C                                                                       D   0520
```

```
   25 XTEST=XIN(1)-XIN(2)                                              D    0530
      YTEST=YIN(1)-YIN(2)                                              D    0540
      IF (XTEST.EQ.0.0) GO TO 30                                       D    0550
      DYDXC=YTEST/XTEST                                                D    0560
      ALPHAC=ATAN2(YTEST,XTEST)                                        D    0570
      GO TO 35                                                         D    0580
   30 DYDXC=99999.                                                     D    0590
      ALPHAC=PIO2                                                      D    0600
   35 K=K-1                                                            D    0610
      NBDC=NBDY1                                                       D    0620
      IF (TYPBDY.EQ.3.0.AND.NBDYS.EQ.3) NBDC=NBODY2                    D    0630
      DO 50 I=NFIRST,NBDC                                              D    0640
      KEEP=NBDC+NFIRST-I                                               D    0650
      XON(K+1)=XON(KEEP)                                               D    0660
      DYDXO(K+1)=DYDXC                                                 D    0670
      ALPHA(K+1)=ALPHAC                                                D    0680
      IF (I.EQ.NFIRST) GO TO 40                                        D    0690
      YON(K+1)=YON(K)+(XON(K+1)-XON(K))*DYDXC                          D    07C0
      IF(COS(ALPHAC).EQ.C.) WRITE(6,44)
      IF(COS(ALPHAC).EQ.C.) STOP
 44   FORMAT( 53HOVERTICAL LINE NOT PERMITTED AS FIRST SEGMENT ON BODY-)
      SON(K+1)=SON(K)+(XON(K)-XON(K+1))/COS(ALPHAC)                    D    0711
      GO TO 45                                                         D    0720
   40 YON(K+1)=YIN(1)+(XON(K+1)-XIN(1))*DYDXC                          D    0730
   45 CAPPA(K+1)=0.0                                                   D    0740
      ALPHA(K+1)=ALPHA(K+1)/PIO180                                     D    0750
      K=  1                                                            D    0760
   50 CON .NUE                                                         D    0770
      DELSI=SON(K)-SON(K-1)                                            D    0780
      RETURN                                                           D    0790
      END                                                              D    08C0
```

```
      SUBROUTINE TEST (IA)                                              E   0000
      COMMON /MAIN/ XIN(10),YIN(10),DELSMX,PIO2,DELSI,IHUB              E   0010
      COMMON /FOR3SS/IO,DELS,XBK(20),YBK(20),XON(500),YON(500),DYDXO(500E   0020
     1),ALPHA(500),CAPPA(500),SON(500),PIO180                           E   0030
      COMMON /SPREP/ KPREP                                              E   0040
      M=IA-1                                                            E   0050
      IF (XIN(2).EQ.XIN(1)) GO TO 10                                    E   0060
      SLP1=(YIN(2)-YIN(1))/(XIN(2)-XIN(1))                              E   0070
      GO TO 15                                                          E   0080
   10 TEST1=(YIN(2)-YIN(1))/(XIN(M)-XIN(1))                             E   0090
      SLP1=SIGN(99999.,TEST1)                                           E   0100
   15 IF ((XIN(1).LT.XIN(M).AND.XIN(1).LE.XIN(2)).OR.(XIN(1).GE.XIN(2).AE   0110
     1ND.XIN(1).GT.XIN(M))) GO TO 20                                    E   0120
      TIP=XIN(1)-XIN(2)                                                 E   0130
      XIN(1)=XIN(1)+SIGN(50.,TIP)                                       E   0140
                                                YIN(1)=(YIN(2)-YIN(1))*E   0150
     1(XIN(1)-XIN(2))/(XIN(2)-XIN(1)+SIGN(50.,TIP))+YIN(2)              E   0160
   20 SLP2=(YIN(M)-YIN(1))/(XIN(M)-XIN(1))                              E   0170
C                                                                       E   0180
C     ROTATION ONLY                                                     E   0190
C                                                                       E   0200
   25 IF (SLP1.GT.SLP2) RETURN                                          E   0210
C                                                                       E   0220
C     MIRROR INTO XIN(1)                                                E   0230
C                                                                       E   0240
   30 CALL PRELPS (2,0,IA,1,1)                                          E   0250
      RETURN                                                            E   0260
      END                                                               E   0270
```

```
      SUBROUTINE PRELPS (KODE,KAT,IA,K1,K2)                             F   0000
      COMMON /MAIN/ XIN(10),YIN(10),DELSMX,PIO2,DELS1,IHUR              F   0010
      COMMON /FOR3SS/IO,DELS,XBK(20),YBK(20),XON(500),YON(500),DYDXO(500F   0020
     1),ALPHA(500),CAPPA(500),SON(500),PIO180                           F   0030
      COMMON /SPREP/ KPREP                                              F   0040
      KPREP=1                                                           F   0050
      IL=IA                                                             F   0060
      IF (KAT.EQ.1) GO TO 60                                            F   0070
      KID=KODE                                                          F   0080
      X1=XIN(1)                                                         F   0090
      Y1=YIN(1)                                                         F   0100
      IF (IA.EQ.5) GO TO 10                                             F   0110
      XC=XIN(4)                                                         F   0120
      YC=YIN(4)                                                         F   0130
      GO TO 15                                                          F   0140
   10 XC=XIN(5)                                                         F   0150
      YC=YIN(5)                                                         F   0160
      IL=IA+1                                                           F   0170
   15 DO 45 IB=1,IL                                                     F   0180
      IF (IB.NE.6) GO TO 20                                             F   0190
      IF (ABS(XIN(6))+ABS(YIN(6)).LE.1.E-15.OR.YIN(6).EQ.200.) GO TO 45 F   0200
      IF (YIN(6).NE.-100.) GO TO 20                                     F   0210
      XIN(6)=-XIN(6)                                                    F   0220
      GO TO 45                                                          F   0230
   20 GO TO (25,30,35,40,30,40),KODE                                    F   0240
   25 YIN(IB)=YIN(IA)-(YIN(IB)-YIN(IA))                                 F   0250
      GO TO 45                                                          F   0260
   30 XIN(IB)=XIN(1)-(XIN(IB)-XIN(1))                                   F   0270
      GO TO 45                                                          F   0280
   35 YIN(IB)=YIN(1)-(YIN(IB)-YIN(1))                                   F   0290
      GO TO 45                                                          F   0300
   40 XIN(IB)=XIN(IA)-(XIN(IB)-XIN(IA))                                 F   0310
   45 CONTINUE                                                          F   0320
      IF (KODE.EQ.5) GO TO 50                                           F   0330
      IF (KODE.EQ.6) GO TO 55                                           F   0340
      RETURN                                                            F   0350
   50 KODE=1                                                            F   0360
      GO TO 15                                                          F   0370
   55 KODE=3                                                            F   0380
      GO TO 15                                                          F   0390
   60 DO 90 IB=K1,K2                                                    F   0400
      GO TO (65,70,75,80,65,75),KID                                     F   0410
   65 YON(IB)=YC-(YON(IB)-YC)                                           F   0420
      GO TO 85                                                          F   0430
   70 XON(IB)=X1-(XON(IB)-X1)                                           F   0440
      GO TO 85                                                          F   0450
   75 YON(IB)=Y1-(YON(IB)-Y1)                                           F   0460
      GO TO 85                                                          F   0470
   80 XON(IB)=XC-(XON(IB)-XC)                                           F   0480
   85 DYDXO(IB)=-DYDXO(IB)                                              F   0490
   90 CONTINUE                                                          F   0500
      IF (KID.EQ.5) GO TO 95                                            F   0510
      IF (KID.EQ.6) GO TO 100                                           F   0520
      RETURN                                                            F   0530
   95 KID=2                                                             F   0540
      GO TO 60                                                          F   0550
  100 KID=4                                                             F   0560
      GO TO 60                                                          F   0570
      END                                                               F   0580
```

```
      SUBROUTINE ELIPSE (K)                                              6    0000
C                                                                        6    0010
C     THIS SUBROUTINE FITS A SEGMENT OF AN ELLIPSE TO TWO ARBITRARILY    6    0020
C                 ORIENTED STRAIGHT LINES NOT MORE THAN 9C DEGREES APART 6    0030
C                                                                        6    0040
      COMMON /MAIN/ XIN(10),YIN(10),DELSMX,PIO2,DELSI,IHUB               6    0050
      COMMON /FOR3SS/IO,DELS,XBK(20),YBK(20),XON(500),YON(500),DYDXO(5006    0060
     1),ALPHA(500),CAPPA(500),SON(500),PIO180                            6    0070
      COMMON /SS/ NBDY1,NBODY2,TYPBDY,NBDYS                              6    0080
C                                                                        6    0090
C     TRANSLATE INPUT BREAK POINTS SO THAT POINT NO. 2 BECOMES           6    01C0
C        THE ORIGIN                                                      6    0110
C                                                                        6    0120
      DELSIN=DELSI                                                       6    0130
      KOUNT=0                                                            6    0140
      DELS=DELSI                                                         6    0150
      PI=3.141592653                                                     6    0160
      K=K-1                                                              6    0170
      KSTART=K                                                           6    0180
      X2=XIN(2)                                                          6    0190
      Y2=YIN(2)                                                          6    02C0
      DO 10 I=1,4                                                        6    0210
      XIN(I)=XIN(I)-X2                                                   6    0220
   10 YIN(I)=YIN(I)-Y2                                                   6    0230
C                                                                        6    0240
C     ROTATE THE TRANSLATED BREAK POINTS SO THAT THE SLOPE OF THE        6    0250
C        FIRST STRAIGHT LINE IS ZERO                                     6    0260
C                                                                        6    0270
      IF (XIN(2).NE.XIN(1)) GO TO 15                                     6    0280
      SLOPE=99999.                                                       6    0290
      PHI=-PIO2                                                          6    03C0
      IF (YIN(1).GT.YIN(2)) PHI=PIO2                                     6    0310
      GO TO 20                                                           6    0320
   15 SLOPE=(YIN(2)-YIN(1))/(XIN(2)-XIN(1))                              6    0330
      PHI=ATAN(SLOPE)                                                    6    0340
      IF (XIN(1).LT.XIN(2)) PHI=PI+ATAN(SLOPE)                           6    0350
   20 DO 25 I=1,4                                                        6    0360
      XA=XIN(I)                                                          6    0370
      XIN(I)=XA*COS(PHI)+YIN(I)*SIN(PHI)                                 6    0380
   25 YIN(I)=-XA*SIN(PHI)+YIN(I)*COS(PHI)                                6    0390
C                                                                        6    04C0
C     DETERMINE THE ELLIPSE                                              6    0410
C                                                                        6    0420
      IF (XIN(4).NE.XIN(3)) GO TO 30                                     6    0430
      B=YIN(3)                                                           6    0440
      A=ABS(XIN(3))                                                      6    0450
      PHIAB=PIO2                                                         6    0460
      GO TO 35                                                           6    0470
   30 SLOP2=(YIN(4)-YIN(3))/(XIN(4)-XIN(3))                              6    0480
      C3=XIN(3)*SLOP2/YIN(3)                                             6    0500
      RAD = (C3 - 2.0)/C3
C**
      WRITE(6,1000) C3,RAD,SLOP2,SLOPE,PHI,YIN(1),XIN(1),YIN(2),XIN(2),
     1      YIN(3),XIN(3),YIN(4),XIN(4)
 1000 FORMAT(//1X,46HC3 RAD SLOP2 SLOPE PHI Y1 X1 Y2 X2 Y3 X3 Y4 X4,//
     1           13E10.3)
```

```
C**
      IF(RAD .LT. 0.0) GO TO 135
      PHIAB = 2.0 * ATAN(SQRT(RAD))
      A=-XIN(3)/SIN(PHIAB)                                              6  0520
      B=YIN(3)/(1.0-COS(PHIAB))                                         6  0530
C**
      WRITE(6,1100) PHIAB,A,B
 1100 FORMAT(1X//9HPHIAB A B,3E10.3)
C**
   35 THETMX=PHIAB-PIO2                                                 6  0540
      THTMXD=THETMX/PIO180                                              6  0550
      WRITE (6,140)A,B,XIN(1),YIN(1),PHI,THTMXD                         6  0560
C                                                                       6  0570
C     INITIALIZE THE FIRST POINT ON THE ELLIPSE                         6  0580
C                                                                       6  0590
   40 K=KSTART                                                          6  0600
      XON(K+1)=XIN(2)                                                   6  0610
      YON(K+1)=YIN(2)                                                   6  0620
      CAPPA(K+1)=-B/(A**2)                                              6  0630
      ALPHA(K+1)=0.0                                                    6  0640
      DYDXO(K+1)=0.0                                                    6  0650
      KOUNT=KOUNT+1                                                     6  0660
      THET=-PIO2                                                        6  0670
      DSSAVE=DELS                                                       6  0680
      DS=DELS/(1.0+.2*TANH(ABS(CAPPA(I))))                              6  0690
      DTHET=DS/ABS(A)                                                   6  0700
      THET=THET+DTHET                                                   6  0710
C                                                                       6  0720
C     GENERATE THE POINTS ON THE ELLIPSE                                6  0730
C                                                                       6  0740
   45 K=K+1                                                             6  0750
   50 XON(K+1)=-A*COS(THET)                                             6  0760
      YON(K+1)=B*(1.0+SIN(THET))                                        6  0770
      SON(K+1)=SON(K)+SQRT((XON(K+1)-XON(K))**2+(YON(K+1)-YON(K))**2)   6  0780
      IF (ABS(SON(K+1)-SON(K)).GT.1.05*DS) GO TO 55                     6  0790
      IF (ABS(SON(K+1)-SON(K)).LT..95*DS) GO TO 60                      6  0800
      GO TO 65                                                          6  0810
   55 THET=THET-.02*DTHET                                               6  0820
      GO TO 50                                                          6  0830
   60 THET=THET+.02*DTHET                                               6  0840
      GO TO 50                                                          6  0850
   65 IF (THET.EQ.0.0) GO TO 70                                         6  0860
      DYDXO(K+1)=B*COTAN(THET)/A                                        6  0870
      ALPHA(K+1)=ATAN(DYDXO(K+1))                                       6  0880
      GO TO 75                                                          6  0890
   70 DYDXO(K+1)=99999.                                                 6  0900
      ALPHA(K+1)=PIO2                                                   6  0910
   75 CAPPA(K+1)=-A*B/(B*B*COS(THET)**2+A*A*SIN(THET)**2)**1.5          6  0920
      DS=DELS/(1.0+.2*TANH(ABS(CAPPA(I))))                              6  0930
      IF (ABS(DS-DELS).GT..20*DELS) DS=DELS+SIGN(.20*DELS,DS-DELS)      6  0940
      DSSAVE=DS                                                         6  0950
   80 DTHET=DS/SQRT(B*B*COS(THET)**2+A*A*SIN(THET)**2)                  6  0960
      DTS=DTHET                                                         6  0970
      THET=THET+DTHET/2.0                                               6  0980
      DTHET=DS/SQRT(B*B*COS(THET)**2+A*A*SIN(THET)**2)                  6  0990
      IF (ABS(DTHET-DTS).LT..001*DTS) GO TO 85                          6  1000
      GO TO 80                                                          6  1010
```

```
   85 IF (THET.LE.THETMX-DTHET/2.0) GO TO 45                          6    1020
      IF (KOUNT.GT.100) GO TO 115                                     6    1030
      DELSS=DELS                                                      6    1040
      DSTEST=((XON(K+1)-XIN(3))**2+(YON(K+1)-YIN(3))**2)**.5          6    1050
      IF (ABS(DS-DSTEST).LT..01*DS) GO TO 90                          6    1060
      IF (DSTEST.GT.DS) GO TO 110                                     6    1070
      IF (DSTEST.LT..01*DS) GO TO 95                                  6    1080
      IF (DSTEST-.5*DS) 105,90,100                                    6    1090
   90 K=K+1                                                           6    1100
   95 XON(K+1)=XIN(3)                                                 6    1110
      YON(K+1)=YIN(3)                                                 6    1120
      GO TO 115                                                       6    1130
  100 DELS=(FLOAT(K+1-KSTART)*DELS+DSTEST)/FLOAT(K+2-KSTART)          6    1140
      IF (KOUNT.GE.10) DELS=(DELS+DELSS)/2.0                          6    1150
      GO TO 40                                                        6    1160
  105 DELS=DELS+DSTEST/FLOAT(K+2-KSTART)                              6    1170
      IF (KOUNT.GE.10) DELS=(DELS+DELSS)/2.0                          6    1180
      GO TO 40                                                        6    1190
  110 DELS=.8*DELS                                                    6    1200
      GO TO 40                                                        6    1210
  115 DELS1=DS*1.2                                                    6    1220
      IF (DELS1.GT.DELS) DELS1=DELS                                   6    1230
      WRITE (6,145)KOUNT                                              6    1240
      WRITE (6,155)DELSIN,DELS,DELS1,DSTEST                           6    1250
      KEND=K+1                                                        6    1260
      KSTART=KSTART+1                                                 6    1270
C                                                                     6    1280
C     ROTATE AND TRANSLATE BACK                                       6    1290
C                                                                     6    1300
      DO 130 KROT=KSTART,KEND                                         6    1310
      XA=XON(KROT)                                                    6    1320
      XON(KROT)=XA*COS(PHI)-YON(KROT)*SIN(PHI)+X2                     6    1330
      YON(KROT)=XA*SIN(PHI)+YON(KROT)*COS(PHI)+Y2                     6    1340
      ALPHA(KROT)=ALPHA(KROT)+PHI                                     6    1350
      IF (ALPHA(KROT).EQ.PIO2) GO TO 120                              6    1360
      DYDXO(KROT)=TAN(ALPHA(KROT))                                    6    1370
      GO TO 125                                                       6    1380
  120 DYDXO(KROT)=99999.                                              6    1390
  125 ALPHA(KROT)=ALPHA(KROT)/PIO180                                  6    1400
  130 CONTINUE                                                        6    1410
      RETURN                                                          6    1420
  135 WRITE (6,150)SLOP2,XIN(3),YIN(3)                                6    1430
      STOP                                                            6    1440
C                                                                     6    1450
C                                                                     6    1460
  140 FORMAT (1H0,25X,4HA  =,1PE10.3,5X,4HB = ,1PE10.3,5X,5HX0 = ,1PE10.6    14
     13,5X,7HY0    = ,1PE10.3/24X,
     1                          7HPHI  = ,1PE10.3,5X,9HTHETMX = ,1PE10.36    1480
     2)                                                               6    1490
  145 FORMAT (26X,I3,2X,13HITERATIONS---)
  150 FORMAT (1H0,25X,42HCOMBINATION OF SLOPE, X , Y NOT COMPATIBLE/30X,
     1                                                                9
     1HSLOPE2 = ,F7.3,3X,9HXIN(3) = ,F7.3,3X,9HYIN(3) = ,F7.3)        6    1520
  155 FORMAT (11X,10HDELS IN = ,F8.5,3X,7HDELS = ,F8.5,3X,11HDELS OUT = 6    1530
     1,F8.5,3X,9HDSTEST = ,F8.5)                                      6    1540
      END                                                             6    1550
```

```
      SUBROUTINE SUPERC (XBRK,YBRK,ENREED,DELS1,ISTART)               H  0000
      DIMENSION ENREED(2)                                             H  0010
      DIMENSION XBRK(6), YBRK(6), XBK1(13), YBK1(13)                  H  0020
      REAL LOGXOA,LOGYOB                                              H  0030
      COMMON /SUPF/ IFLO                                              H  0040
      COMMON /FOR3SS/IO,DELS,XBK(20),YBK(20),XON(500),YON(500),DYDXO(500H  0050
     1),ALPHA(500),CAPPA(500),SON(500),PIO180                         H  0060
      COMMON/SUPN/ XOA(2),YOB(2),LOGXOA(2),LOGYOB(2)                  H  0070
      COMMON /NHIGH/ NSPHG,NLAST,XLAST(500),YLAST(500)                H  0080
      COMMON/SENSE/ XB(2),YB(2),A,B,INFLEC                            H  0090
      COMMON /SPREP/ KPREP                                            H  0100
      COMMON/PAC/ PACE,DELSML                                         H  0110
      COMMON/TRYFIT/KOUNT                                             H  0120
      IPI=0                                                           H  0130
      PI=3.14159265                                                   H  0140
      IF (IFLO.EQ.1) DELSML=DELS1                                     H  0150
      DELSIN=DELS1                                                    H  0160
      DELS2=DELS1                                                     H  0170
      KOUNT=0                                                         H  0180
   10 II=ISTART                                                       H  0190
      IF (IFLO.GE.1) DS=DELS2                                         H  0200
      KOUNT=KOUNT+1                                                   H  0210
      I=ISTART                                                        H  0220
      X4T=XBRK(4)                                                     H  0230
      Y4T=YBRK(4)                                                     H  0240
C((((((( FOR SUBSEQUENT ITERATIONS, SKIP LL INITIALIZATION.           H  0250
      IF ((LL.GE.5.OR.LL.LT.-5).AND.KOUNT.NE.1) GO TO 50              H  0260
      LL=0                                                            H  0270
C((((((  FLAG FOR ENDPT. CURVATURE MATCH IS YBRK(6)=200.              H  0280
      IF (YBRK(6)-200.) 40,15,40                                      H  0290
   15 CAP=-ABS(XBRK(6))                                               H  0300
      LL=-8                                                           H  0310
      IF (XBRK(6).NE.999.) GO TO 50                                   H  0320
      IF (XBRK(3).NE.XBRK(2)) GO TO 20                                H  0330
C(((((( IF THE PT. WHERE  CURVATURES MUST MATCH HAS INFIN. SLOPE,     H  0340
C(((((( OBTAIN CURVATURE FROM PREVIOUS SEGMENT'S EXPONENTS.           H  0350
      IF (XBRK(2).NE.XBRK(1)) GO TO 25                                H  0360
      LL=-8                                                           H  0370
      CAP=-2.*A/XB(1)/B/B                                             H  0380
      GO TO 50                                                        H  0390
   20 IF (XBRK(5).NE.XBRK(4)) GO TO 25                                H  0400
      LL=-8                                                           H  0410
      CAP=-2.*A/XB(1)/B/B                                             H  0420
      GO TO 50                                                        H  0430
   25 IF(XBRK(6).EQ.999.)XBRK(6)=  CAPPA(ISTART)                      H  0440
      IF (XBRK(6).EQ.0..OR.ABS(XBRK(6)).GE.9999.) GO TO 30            H  0450
      CAP=-ABS(XBRK(6))                                               H  0460
      GO TO 50                                                        H  0470
   30 WRITE (6,35)                                                    H  0480
   35 FORMAT(1H0,3X,99HREQUEST FOR SPECIFIC CURVATURE MUST BE MODIFIED OH  0490
     1R WITHDRAWN. DESIRED CURVATURE CAN'T=0. OR INFINTY)             H  0500
      STOP                                                            H  0510
C((((((   POINT-PLUS-SLOPE FLAG IS    YBRK(6)= -100.                  H  0520
   40 IF (YBRK(6)+100.) 50,45,50                                      H  0530
   45 LL= 5                                                           H  0540
C(((((( ADDITIONAL FLAG FOR INFLECTION-POINT-PLUS-SLOPE IS  YBRK(3)= 100H  0550
```

```
      IF(YBRK(3).EQ.100.) LL=6                                          H   0560
C(((((FOR INITIAL GUESS OF UNKNOWN Y AT INFLECTION POINT,USE ENDPT. AVG.H   0570
      IF(LL.EQ.6) YBRK(3)= YBRK(2)+(XBRK(2)-XBRK(3))/(XBRK(2)-XBRK(4))*(H   0580
     1YBRK(4)-YBRK(2))                                                  H   0590
C(((((( CREATE A DUMMY POINT TO SIMULATE THE GIVEN SLOPE THROUGH        H   0600
C   (XBRK(3),YBRK(3))                                                  H   0610
      YBRK(6)= XBRK(6)*((XBRK(3)+5.)-XBRK(3)) +YBRK(3)                  H   0620
      XBRK(6)= XBRK(3)+5.                                               H   0630
   50 DO 55 J=1,6                                                       H   0640
      XBK1(J+7)=XBRK(J)                                                 H   0650
      YBK1(J+7)=YBRK(J)                                                 H   0660
   55 CONTINUE                                                          H   0670
      IF (XBK1(9).NE.XBK1(8)) GO TO 65                                  H   0680
      IF (YBK1(8).LT.YBK1(9)) GO TO 60                                  H   0690
      SLOP=99999.                                                       H   0700
      SINATD=1.0                                                        H   0710
      SLOPE=99999.                                                      H   0720
      ATDYDD=90.                                                        H   0730
      COSATD=0.0                                                        H   0740
      GO TO 80                                                          H   0750
   60 SLOP=-99999.                                                      H   0760
      SINATD=-1.0                                                       H   0770
      SLOPE=-99999.                                                     H   0780
      ATDYDD=-90.                                                       H   0790
      COSATD=0.0                                                        H   0800
      GO TO 80                                                          H   0810
   65 SLOP=(YBK1(9)-YBK1(8))/(XBK1(9)-XBK1(8))                          H   0820
      SLOPE=ATAN(SLOP)                                                  H   0830
      ATDYDD=SLOPE/PIO18C                                               H   0840
      IF (XBK1(8)-XBK1(9)) 70,75,75                                     H   0850
   70 SLOPE=PI+SLOPE                                                    H   0860
      ATDYDD=-ATDYDD                                                    H   0870
      IPI=1                                                             H   0880
   75 SINATD=SIN(SLOPE)                                                 H   0890
      COSATD=COS(SLOPE)                                                 H   0900
   80 CONTINUE                                                          H   0910
      INFLEC = ABS(INT(SINATD))                                         H   0920
      XBK(8)=XBK1(8)                                                    H   0930
      YBK(8)=YBK1(8)                                                    H   0940
      DO 85 J=9,13                                                      H   0950
      XP=XBK1(J)-XBK1(8)                                                H   0960
      YP=YBK1(J)-YBK1(8)                                                H   0970
      XBK(J)=XBK1(8)+XP*COSATD+YP*SINATD                                H   0980
      YBK(J)=YBK1(8)-XP*SINATD+YP*COSATD                                H   0990
   85 CONTINUE                                                          H   1000
      Q=1.                                                              H   1010
      P=1.                                                              H   1020
      XBK(5)=XBK(8)                                                     H   1030
                                                                        H   1040
      YBK(5)=YBK(8)                                                     H   1050
      XBK(6)=XBK(9)                                                     H   1060
      YBK(6)=YBK(9)                                                     H   1070
      XBK(9)=XBK(10)                                                    H   1080
      YBK(9)=YBK(10)                                                    H   1090
      DELS=DELS1                                                        H   1100
      DSSAVE=DELS                                                       H   1110
      XIM=XBK1(9)                                                       H   1120
```

```
      YTM=YBKI(9)                                                       H   1130
                                                                        H   1140
      B=YBK(11)-YBK(6)                                                  H   1150
      TOMEGA=(XBK(12)-XPK(11))/(YBK(12)-YBK(11))                        H   1160
      IF (ABS(TOMEGA).LE..0001) TOMEGA=0.                               H   1170
      OMEGA=ATAN(TOMEGA)                                                H   1180
      XO=XBK(6)+B*TOMEGA                                                H   1190
      YO=YBK(11)                                                        H   1200
      A=XO-XBK(11)                                                      H   1210
      X19=XO-XBK(9)                                                     H   1220
      ETA9=YO-YBK(9)                                                    H   1230
      Y8(1)=ETA9                                                        H   1240
      X8(1)=X19-ETA9*TOMEGA                                             H   1250
      X8(2)=XO-XBK(13)-TOMEGA*(YO-YBK(13))                              H   1260
      Y8(2)=YO-YBK(13)                                                  H   1270
      BOA=B/A                                                           H   1280
      IF (LL.GE.5) GO TO 90                                             H   1290
      IF (LL.LE.-6) GO TO 105                                           H   1300
      LL=0                                                              H   1310
      IF(ENREED(1).GT.0.) P=ENREED(1)                                   H   1320
      IF(ENREED(2).GT.0.) Q=ENREED(2)                                   H   1330
      IF(P.EQ.1.) LL=1                                                  H   1340
      IF(Q.EQ.1.) LL=LL+2                                               H   1350
      IF (LL.EQ.0) GO TO 115                                            H   1360
   90 XOA(1)=X8(1)/A                                                    H   1370
      YOB(1)=Y8(1)/B                                                    H   1380
      LOGXOA(1)=ALOG(XOA(1))                                            H   1390
      LOGYOB(1)=ALOG(YOB(1))                                            H   1400
      IF (XBRK(6).EQ.0..AND.LL.EQ.3) GO TO 95                           H   1410
      IF (LL.NE.3) GO TO 100                                            H   1420
      XOA(2)=X8(2)/A                                                    H   1430
      YOB(2)=Y8(2)/B                                                    H   1440
      LOGXOA(2)=ALOG(XOA(2))                                            H   1450
      LOGYOB(2)=ALOG(YOB(2))                                            H   1460
      GO TO 100                                                         H   1470
   95 LL=4                                                              H   1480
  100 CALL FUNISO(P,Q,LL)                                               H   1490
      GO TO 115                                                         H   1500
C((((((  FOR CURVATURE MATCH, NO ITERATION REQD. BUT ONE EXPONENT MUST=2H   1510
  105 IF(XBRK(3).EQ.XBRK(2))P=2.                                        H   1520
      IF(XBRK(3).EQ.XBRK(4))Q=2.                                        H   1530
C(((((( OBTAIN OTHER EXPONENT FROM ENDPOINT CURVATURE RELATION          H   1540
  110 IF(P .EQ. 2.)  Q= -2.*B/CAP/A/A                                   H   1550
      IF(Q .EQ. 2. .AND. P .NE. 2.)  P=-2.*A/CAP/B/B                    H   1560
  115 IF (KOUNT.NE.1) GO TO 120                                         H   1570
      WRITE (6,475)P,A,XC,Q,B,YO,OMEGA                                  H   1580
  120 I=I                                                               H   1590
      ILO=I                                                             H   1600
      XON(I)=XON(I)                                                     H   1610
      CION=1./P                                                         H   1620
      BT=B*TOMEGA                                                       H   1630
      DX1=DELS*COSATD                                                   H   1640
      XP=XTM-XBKI(8)                                                    H   1650
      YP=YTM-YBKI(8)                                                    H   1660
      XIIROT=XBKI(8)+XP*COSATD+YP*SINATD                                H   1670
      XI=XO-XIIROT                                                      H   1680
      Y=YO-YBK(6)                                                       H   1690
```

```
      X=XI-Y*TOMEGA
      IF (X.LT.0.0) X=0.0
      DSM=SON(I)-SON(I-1)
  125 XOATON=(X/A)**P
      YOBYON=(Y/B)**Q
C(((((( AVOID (.LE. 0.)**(.LE. 0.)
      IF (P.GE.1.) GO TO 135
      IF (X.NE.0.) GO TO 130
      XNMOAN=99999.
      GO TO 140
  130 XNMOAN=(1./X)**(1.-P)/A**P
      GO TO 140
  135 XNMOAN=X**(P-1.)/A**P
  140 IF (Q.GE.1.) GO TO 150
      IF (Y.NE.0.) GO TO 145
      YNMOBN=9999.
      GO TO 155
  145 YNMOBN=(1./Y)**(1.-Q)/B**Q
      GO TO 155
  150 YNMOBN=Y**(Q-1.)/B**Q
  155 FOFY=XOATON+YOBYON-1.
      IF (ABS(FOFY).LE.1.0E-5) GO TO 160
      FPOFY=Q*YNMOBN-TOMEGA*P*XNMOAN
      YNEW=Y-FOFY/FPOFY
      GO TO 165
  160 YNEW=Y
  165 IF (ABS(Y-YNEW)/YNEW-.1E-4) 175,175,170
  170 Y=YNEW
      X=XI-Y*TOMEGA
      GO TO 125
  175 Y=YNEW
      X=XI-Y*TOMEGA
  180 ETA=Y
      DELS=DELS2
      IPN=1
      IF (X.LT.0.0) X=0.0
C(((((( AVOID (.LE. 0.)**(.LE. 0.)
      IF (P.GE.1.) GO TO 190
      IF (X.NE.0.) GO TO 185
      XOANM1= 99999.
      GO TO 195
  185 XOANM1=(A/X)**(1.-P)
      GO TO 195
  190 XOANM1=(X/A)**(P-1.)
  195 IF (Q.GE.1.) GO TO 205
      IF (Y.NE.0.) GO TO 200
      YOBNM1= 99999.
      GO TO 210
  200 YOBNM1= (B/Y)**(1.-Q)
      GO TO 210
  205 YOBNM1=(Y/B)**(Q-1.)
  210 F1=(XOANM1/A)*P
      F2=(YOBNM1/B)*Q
      F3=TOMEGA*F1
      IF (X.EQ.0.0) GO TO 225
      IF (P.GE.2.) GO TO 215
      F1OX=P*(1./X)**(2.-P)/X**P
```

```
      GO TO 220
  215 F1OX=P*X**(P-2.)/A**P
  220 GO TO 230
  225 IF(P.EQ.2.) F1OX=2.0/(A*A)
      IF(P.GT.2.) F1OX=0.0
  230 IF (Y.EQ.0.0) GO TC 240
      IF (Q.GE.2.) GO TO 235
      F2OY=Q*(1./Y)**(2.-Q)/B**Q
      GO TO 245
  235 F2OY=Q*Y**(Q-2.)/B**Q
      GO TO 245
  240 IF(Q.EQ.2.) F2OY=2./(B*B)
      IF(Q.GT.2.) F2OY=0.
  245 DEN=F2-F3
      IF (DEN.NE.0.0) GO TO 250
      DETDXI=99999.
      GO TO 255
  250 DETDXI=-F1/DEN
  255 DYDXO(IPN)=DETDXI
      CIMEPT=1.-DETDXI*TOMEGA
      IF (IPN.EQ.ISTART.AND.LL.LE.-6) GO TO 260
C((((((  ELIMINATE CASES OF UNDEFINED CURVATURE
      IF (X.EQ.0..AND.P.LT.2.) GO TO 265
      IF (Y.EQ.0..AND.Q.LT.2.) GO TO 265
      G1=(P-1.)*F1OX*CIMEPT
      SAND1=DEN*G1
      SAND2=F1*((Q -1.)*F2OY*DETDXI-G1*TOMEGA)
      IF(ABS(DETDXI).GT.1.E 11) BETDXI=1.E 11
      BKT=(1.+DETDXI**2)**1.5
      CAPPA(IPN)=(SAND2-SAND1)/(DEN**2*BKT)
      IF(P.EQ.2..AND.X.EQ.0.)CAPPA(IPN)=-2.*B/Q/A/A
      IF(Q.EQ.2..AND.ABS(Y).LT.1.E-4)CAPPA(IPN)=-2.*A/P/B/B
      GO TO 270
  260 IF(P.EQ.2.)CAPPA(IPN)=-2.*B/Q/A/A
      IF(Q.EQ.2.)CAPPA(IPN)=-2.*A/P/B/B
      GO TO 270
  265 CAPPA(IPN)=99999.
  270 ALPHA(IPN)=ATAN(DYDXO(IPN))/PIO180
      XON(IPN)=XO-XI
      YON(IPN)=YO-ETA
      DYI=DELS*SINATD
      IF (IFLD.GE.1) GO TO 275
      DS=DELS/(1.0+.20*TANH(ABS(CAPPA(I))))
      GO TO 285
  275 IF (IFLD.GT.1.AND.(IPN-ISTART).GT.3) GO TO 280
      DS=DS-PACE*DS
      GO TO 290
  280 DS=DS+1.5*PACE*DS
      IF (DS.GT.DELSHL) DS=DELSHL
      GO TO 290
  285 IF (ABS(DS-DELS).GT..20*DELS) DS=DELS+SIGN(.20*DELS,DS-DELS)
  290 IF (DS.LT..02*DELS2) DS=.02*DELS2
      IF (IFLD.GT.0.AND.(I-ISTART).GT.200) GO TO 410
      IF (NSPHG.EQ.0) GC TO 295
      DXI=ABS(XLAST(NLAST)-XLAST(NLAST-1))
      DXI1=DXI
      DYI=ABS(YLAST(NLAST)-YLAST(NLAST-1))
```

```
      NLAST=NLAST-1                                                     H  2840
  295 IF (ABS(DETDXI)-1.) 320,320,300                                   H  2850
  300 DYI=DS/SQRT(1.+1./DETDXI**2)                                      H  2860
      IF (NSPHG.NE.0) DY1=DX1                                           H  2870
  305 YTM=YON(I)+DYI                                                    H  2880
      IF (YTM-YBK(11)) 310,390,390                                      H  2890
  310 ETA=Y0-YTM                                                        H  2900
      Y=ETA                                                             H  2910
C                                                                       H  2920
C     STRAIGHT SECTION BETWEEN POINTS 11 AND 12 MUST HAVE SLOPE ABOVE 1 H  2930
C                                                                       H  2940
C                                                                       H  2950
C     X MAY NOT BE TESTED AGAINST XBK(11)                               H  2960
C                                                                       H  2970
      X=A*(1.-(Y/B)**Q)**CION                                           H  2980
      XI=X+Y*TOMEGA                                                     H  2990
      XTM=X0-XI                                                         H  3000
      DX1=XTM-XON(I)                                                    H  3010
      DELTAS=SQRT(DYI**2+DXI**2)                                        H  3020
      IF (DELTAS.GT.1.02*DS.AND.IPN.NE.1.AND.NSPHG.EQ.0) GO TO 315      H  3030
      GO TO 390                                                         H  3040
  315 DY1=DS*DY1/DELTAS                                                 H  3050
      GO TO 305                                                         H  3060
  320 DX1=DS/SQRT(1.+DETDXI**2)                                         H  3070
      IF (NSPHG.NE.0) DXI=DYI                                           H  3080
      IF (NSPHG.NE.0.AND.IPI.EQ.1) DX1=DX11                             H  3090
  325 XTM=XON(I)+SIGN(DXI,DETDXI)                                       H  3100
      IF (DETDXI.EQ.0.) XTM=XON(I)-DX1                                  H  3110
      XI=X0-XTM                                                         H  3120
      Y=Y0-YON(I)                                                       H  3130
  330 X=XI-Y*TOMEGA                                                     H  3140
      IF (X.LT.0.0) X=0.0                                               H  3150
      XOATON=(X/A)**P                                                   H  3160
      YOBTON=(Y/B)**Q                                                   H  3170
C((((((  AVOID (.LE. 0.)**(.LE. 0.)                                     H  3180
      IF (P.GE.1.) GO TO 340                                            H  3190
      IF (X.NE.0.) GO TO 335                                            H  3200
      XNMOAN=99999.                                                     H  3210
      GO TO 345                                                         H  3220
  335 XNMOAN=(1./X)**(1.-P)/A**P                                        H  3230
      GO TO 345                                                         H  3240
  340 XNMOAN=X**(P-1.)/A**P                                             H  3250
  345 IF (Q.GE.1.) GO TO 355                                            H  3260
      IF (Y.NE.0.) GO TO 350                                            H  3270
      YNMOBN=9999.                                                      H  3280
      GO TO 360                                                         H  3290
  350 YNMOBN=(1./Y)**(1.-Q)/B**Q                                        H  3300
      GO TO 360                                                         H  3310
  355 YNMOBN=Y**(Q-1.)/B**Q                                             H  3320
  360 FOFY=XOATON+YOBTON-1.                                             H  3330
      IF (ABS(FOFY).LE.1.0E-5) GO TO 365                                H  3340
      FPOFY=Q*YNMOBN-TOMEGA*P*XNMOAN                                    H  3350
      YNEW=Y-FOFY/FPOFY                                                 H  3360
      GO TO 370                                                         H  3370
  365 YNEW=Y                                                            H  3380
  370 IF (ABS(Y-YNEW)/YNEW-.1E-4) 380,380,375                           H  3390
  375 Y=YNEW                                                            H  3400
```

```
      GO TO 330                                                         H  3410
  380 Y=YNEW                                                            H  3420
      X=XI-Y*TOMEGA                                                     H  3430
      YTM=Y0-Y                                                          H  3440
      XI=X+Y*TOMEGA                                                     H  3450
C     DY1=YTM-YON(I-1)                                                  H  3460
      DYI=YTM-YON(I)                                                    H  3470
      DELTAS=SQRT(DY1**2+DX1**2)                                        H  3480
      IF (DELTAS.GT.1.02*DS.AND.IPN.NE.1.AND.NSPHG.EQ.0) GO TO 385      H  3490
      GO TO 390                                                         H  3500
  385 DXI=DS*DXI/DELTAS                                                 H  3510
      GO TO 325                                                         H  3520
  390 SON(I)=SON(I-1)+DSP                                               H  3530
      DSM=DS                                                            H  3540
      IF (NSPHG.NE.0) DS=DXI                                            H  3550
      IF (ABS(YTM-YBK(11)).LT..001*DS) GO TO 400                        H  3560
      I=I+1                                                             H  3570
      IF (YTM-YBK(11)) 180,395,395                                      H  3580
  395 IHI=I-1                                                           H  3590
      GO TO 405                                                         H  3600
  400 IHI=I                                                             H  3610
      I=I+1                                                             H  3620
  405 XTM=XBK(II)                                                       H  3630
      IIHI=IHI                                                          H  3640
  410 IF (IFLD.LE.0.OR.(I-ISTART).LT.200) GO TO 415                     H  3650
      IF(IFLD.EQ.1)                         PACE=PACE-.25*PACE          H  3660
      IF(IFLD.EQ.2)                         PACE=PACE+.25*PACE          H  3670
      I=ILO-1                                                           H  3680
      GO TO 10                                                          H  3690
  415 DO 455 J=ILO,IHI                                                  H  3700
      XP=XON(J)-XBK1(8)                                                 H  3710
      YP=YON(J)-YBK1(8)                                                 H  3720
      XON(J)=XBK1(8)+XP*COSATD-YP*SINATD                                H  3730
      YON(J)=YBK1(8)+XP*SINATD+YP*COSATD                                H  3740
      DEL22=DELS2                                                       H  3750
      IF (J.NE.IHI) GO TO 445                                           H  3760
      DSTEST=((XON(IHI)-X4T)**2+(YON(IHI)-Y4T)**2)**.5                  H  3770
      IF (NOUNT.GT.150) GO TO 445                                       H  3780
      IF (ABS(DS-DSTEST).LT..1*DS) GO TO 420                            H  3790
      IF (DSTEST.LT..01*DS) GO TO 425                                   H  3800
      IF (IHI.EQ.ILO) GO TO 440                                         H  3810
      IF (ABS(DELS2-DSTEST).LT..001*DS) GO TO 435                       H  3820
      IF (DSTEST.LT..5*DS) GO TO 435                                    H  3830
      IF (DSTEST.GT..5*DS) GO TO 430                                    H  3840
C((((((     VIA BUTTON 175/74                                           H  3850
  420 IIHI=IIHI+1                                                       H  3860
      I=I+1                                                             H  3870
      IONE=IIHI-1                                                       H  3880
      SON(IIHI)=SON(IONE  )+DSTEST                                      H  3890
  425 IIHI=IIHI+1                                                       H  3900
      IONE=IIHI-1                                                       H  3910
      XON(IONE  )=X4T                                                   H  3920
      YON(IONE  )=Y4T                                                   H  3930
      GO TO 445                                                         H  3940
  430 IF (IFLD.GT.1) GO TO 435                                          H  3950
      DELS2=(FLOAT(IHI-ILO)*DELS2+DSTEST)/FLOAT(IHI+1-ILO)              H  3960
      IF (NOUNT.GE.10) DELS2=(DELS2+DEL22)/2.0                          H  3970
```

```
            I=ILO-1                                                       H   3980
            GO TO 10                                                      H   3990
        435 DELS2=DELS2+DSTEST/FLOAT(IHI-ILO)                             H   4000
            IF (KOUNT.GE.10) DELS2=(DELS2+DEL22)/2.0                      H   4010
            IF(IFLD.GT.1.AND.(DS/DEL22).GT.2.)                            H   4020
           1PACE=PACE*(1.+DSTEST/DEL22/(FLOAT(IHI-ILO)*(1.+1.5*PACE)**FLOAT(IHH   4030
           2I-ILO-1)-((1.+1.5*PACE)**FLOAT(IFI-ILO)-1.)/1.5/PACE))        H   4040
            IF(IFLD.GT.1.AND.(DS/DEL22).GT.2.)DELS2=DEL22                 H   4050
            I=ILO-1                                                       H   4060
            GO TO 10                                                      H   4070
        440 DELS2=.8*DELS2                                                H   4080
            I=ILO-1                                                       H   4090
            GO TO 10                                                      H   4100
        445 ALPHA(J)=ALPHA(J)-ATDYDD                                      H   4110
            IF (ABS(ABS(ALPHA(J))-90.).LE.1.CE-4) GO TO 450               H   4120
            DYDXO(J)=TAN(ALPHA(J)*PIO180)                                 H   4130
            GO TO 455                                                     H   4140
        450 DYDXO(J)=SIGN(999.,ALPHA(J))                                  H   4150
        455 CONTINUE                                                      H   4160
            IHI=IONE                                                      H   4170
            IF (KOUNT.GT.150) WRITE (6,480)(XBRK(IUB),YBRK(IUB),IUB=1,5)  H   4180
            DELS1=DS                                                      H   4190
            IF (IFLD.EQ.0) DELS1=1.1*DS                                   H   4200
            IF (DELS1.GT.DELS2.AND.IFLD.LE.1)DELS1=DELS2                  H   4210
            DO 465 J=ILO,IHI                                              H   4220
            IF (J.EQ.1) GO TO 460                                         H   4230
            SON(J)=SON(J-1)+SQRT((XON(J)-XON(J-1))**2+(YON(J)-YON(J-1))**2)   H   4240
            GO TO 465                                                     H   4250
        460 SON(J)=0.0                                                    H   4260
        465 CONTINUE                                                      H   4270
            WRITE (6,485)KOUNT                                            H   4280
            WRITE (6,490)DELSIN,DELS2,DELS1,DSTEST                        H   4290
            IF (IFLD.GT.0) WRITE (6,470)PACE                              H   4300
        470 FORMAT(1H ,30X,*FINAL PACE= *,F8.5)
            IO=1                                                          H   4320
            XB(1)=P                                                       H   4330
            XB(2)=Q                                                       H   4340
            XDIF=XBRK(4)-XBRK(5)                                          H   4350
            IF(ABS(XDIF).LT.1.E-15)XDIF=SIGN(1.E-15,XDIF)                 H   4360
            DYDXO(IO-1)=(YBRK(4)-YBRK(5))/XDIF                            H   4370
            IF(ABS(DYDXO(IO-1)).GT.99999.)DYDXO(IO-1)=SIGN(99999.,DYDXO(IO-1))H   4380
            ALPHA(IO-1)=ATAN(DYDXO(IO-1))/PIC180                          H   4390
            IF(CAPPA(IO-1).EQ.C..AND.P.EQ.2..AND.Q.EQ.2.)CAPPA(IO-1)=-2.*A/P/BH   4400
           1/B                                                            H   4410
            RETURN                                                        H   4420
      C                                                                   H   4430
      C                                                                   H   4440
        475 FORMAT(1X/19X,
           1              4HP = ,E16.8,4X,4HA = ,E16.8,7X,5HXO = ,E16.8/19X,
           2                                                         4HQ H   4450
           1= ,E16.8,4X,4HB = ,E16.8,7X,5HYO = ,E16.8,3X,8HOMEGA = ,E16.8/1X) H   4460
        480 FORMAT (1H0,15X,
           1              60HTHIS SET OF DATA EXCEEDED 150ITERATIONS CALCULATIONH   44
           1S STOPPED/20X,4HXBRK,5X,4HYBRK/2CX,1P10E10.3)
        485 FORMAT (21X,  I3,2X,13HITERATIONS---)
        490 FORMAT(21X,10HDELS IN = ,F8.5,3X,7HDELS = ,F8.5,3X,11HDELS OUT = ,H   450
           1F8.5,3X,9HDS TEST = ,F8.5)                                    H   4510
            END                                                           H   4520
```

```
      SUBROUTINE FONISO(P,Q,LL)
C(((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C
C     FOR THE BI-SUPERELLIPSE,  F(P,Q) = (X/A)**P + (Y/B)**Q - 1 = 0.
C
C     IF LL =             AND GIVEN               FIND
C     1                   Q,X1,Y1                 P
C     2                   P,X1,Y1                 Q
C     3                   X1,Y1,X2,Y2             P AND Q
C     4                   X1,Y1                   P = Q = N
C     5                   X1,Y1,SLOPE AT(X1,Y1)   P AND Q
C     6                   (XI,Y)=INFLECTION PT,   P,Q,AND YI
C                         PLUS SLOPE AT (X1,?)
C
      REAL LOGXOA,LOGYOB
      DIMENSION OX(2),OY(2)
      COMMON/TRYFIT/KOUNT
      COMMON /MAIN/ XIN(10),YIN(10),DELSMX,PIO2,DELS1,IHUB
      COMMON/SUPN/ XOA(2),YOB(2),LOGXOA(2),LOGYOB(2)
      COMMON/SENSE/  X(2),Y(2),A,B,INFLEC
      DATA TOL/.1E-6/
      YI=Y(1)
      XI=X(1)
      OX(1)= XIN(3)
      OX(2)=XIN(6)
      OY(1)= YIN(3)
      OY(2)=YIN(6)
      DO 20 I=1,2
      IF (KOUNT.GT.1) GO TO 20
      IF (I.EQ.2.AND.LL.NE.3) GO TO 20
      OA= XOA(I)
      OB= YOB(I)
      IF ((OB+OA).LE.1.0) GO TO 10
      IF (OB.GE.1.0.OR.OA.GE.1.0) GO TO 15
      GO TO 20
   10 WRITE (6,90)OX(I),OY(I)
      GO TO 20
   15 WRITE (6,85)OX(I),OY(I)
      STOP
   20 CONTINUE
      IF (LL.NE.6) GO TO 25
      P=1.5
      Q=.5
   25 IF (LL.NE.6) GO TO 30
      YOB(1)= Y(1)/B
      LOGYOB(1)=ALOG(YOB(1))
      XOA(1)=X(1)/A
      LOGXOA(1)=ALOG(XOA(1))
   30 XOATP1= XOA(1)**P
      YOBTQ1= YOB(1)**Q
      DFDPA1= XOATP1*LOGXOA(1)
      F1= XOATP1 + YOBTQ1 -1.
      DFDQA1 = YOBTQ1*LOGYOB(1)
      GO TO (55,60,35,65,70,70),LL
   35 XOATP2 = XOA(2)**P
      YOBTQ2 = YOB(2)**Q
```

```
      DFDQA2 = YOBTQ2 * LOGYOB(2)                                       I   0560
      F2 = XOATP2 + YOBTQ2 - 1.                                         I   0570
      DFDPA2 = XOATP2*LOGXOA(2)                                         I   0580
   40 QN = (F1*DFDPA2/(DFDQA2*DFDPA1)-F2/DFDQA2)/(1. - DFDPA2*DFDQA1/(DFI   0590
     1DPA1*DFDQA2)) + Q                                                 I   0600
      PN =((Q-QN)*DFDQA1 -F1)/DFDPA1 +  P                               I   0610
   45 TESTP=ABS(PN-P)/P -TOL                                            I   0620
      TESTQ=ABS(QN-Q)/Q -TOL                                            I   0630
      Q=QN                                                              I   0640
      P=PN                                                              I   0650
      IF (TESTP.GT.0..OR.TESTQ.GT.0.) GO TO 25                          I   0660
   50 RETURN                                                            I   0670
   55 PN= -F1/DFDPA1 + P                                                I   0680
      TESTP=ABS(PN-P)/P -TOL                                            I   0690
      P=PN                                                              I   0700
      IF (TESTP) 50,25,25                                               I   0710
   60 QN= -F1/DFDQA1 +Q                                                 I   0720
      TESTQ=ABS(QN-Q)/Q -TOL                                            I   0730
      Q=QN                                                              I   0740
      IF (TESTQ) 50,25,25                                               I   0750
   65 DFDQA1= YOBTQ1*LOGYOB(1)                                          I   0760
      PN=-F1/(DFDPA1+DFDQA1) +P                                         I   0770
      TESTP= ABS(PN-P)/P -TOL                                           I   0780
      P=PN                                                              I   0790
      Q=P                                                               I   0800
      IF (TESTP) 50,25,25                                               I   0810
   70 DYDX= (Y(2)-Y1  )/(X(2)-X1  )                                     I   0820
      F2=DYDX+  P*XOATP1*Y(1)/Q/YOBTQ1/X(1)                             I   0830
      EOCALD= - P*XOATP1*Y(1)/Q/YOBTQ1/X(1)                             I   0840
      DFDQA2=      - (ALOG(B)-1./Q-ALOG(Y(1)))*EOCALD                   I   0850
      DFDPA2=      - (1./P -ALOG(A)+ ALOG(X(1)))*EOCALD                 I   0860
      IF (LL.EQ.6) GO TO 75                                             I   0870
      GO TO 40                                                          I   0880
   75 EOCALD= - P*XOATP1*Y(1)/Q/YOBTQ1/X(1)                             I   0890
      G=F2                                                              I   0900
      DGDP= DFDPA2                                                      I   0910
      DGDQ= DFDQA2                                                      I   0920
      H = EOCALD*((P-1.)/X(1)+((1.-Q)/Y(1))*EOCALD)                     I   0930
      DHDP  = EOCALD*((P-1.)/X(1)+((1.-Q)/Y(1))*EOCALD)*(1./P-ALOG(A)+ALI   0940
     1OG(X(1)))+EOCALD*(1./X(1) +((1.-Q)/Y(1))*EOCALD*(1./P-ALOG(A)+ALOGI   0950
     2(X(1)))))                                                         I   0960
      DHDQ  = EOCALD*((P-1.)/X(1)+((1.-Q)/Y(1))*EOCALD)*(-1./Q+ALOG(B)-AI   0970
     1LOG(Y(1)))+EOCALD*(-EOCALD/Y(1)+((1.-Q)/Y(1))*EOCALD*(-1./Q+ALOG(BI   0980
     2)-ALOG(Y(1)))))                                                   I   0990
      DFDY= Q/Y(1)*YOBTQ1                                               I   1000
      IF(INFLEC.EQ.1)DFDY=P/X(1)*XOATP1                                 I   1010
      DGDY= EOCALD*(Q-1.)/Y(1)                                          I   1020
      IF(INFLEC.EQ.1)                                                   I   1030
     1              DGDY=EOCALD*(1.-P)/X(1)                             I   1040
      EOCAL2=2.*EOCALD                                                  I   1050
      DHDY= EOCALD*((..-1.)/X(1)+((1.-Q)/Y(1))*EOCAL2)*(1.-Q)/Y(1)+EOCALI   1060
     1D*(Q 1.)/Y(1)/Y(1))                                               I   1070
      IF(INFL..EQ.1)                                                    I   1080
     1              DHDY=EOCALD/X(1)*(P-1.)*((P-2.)/X(1)+2.*EOCALD*(1.-Q)/YI   1090
     1(1)))                                                             I   1100
      ONS  DFDPA1*(DGDQ*DHDY-DGDY*DHDQ)-DFDQA1*(DGDP*DHDY-DGDY*DHDP)+   I   1110
     1DFD  ..  *DF.  DGDQ*DHDP)                                         I   1120
```

```
      QN=Q+(FI*(DGDP*DHDY-DGDY*DHDP)-G*(DFDPA1*DHDY-DFDY*DHDP)+H*(DFDPA1I
     1*DGDY-DFDY*DGDP))/WRONSK                                          I
      PN=P+(-FI*(DGDQ*DHDY-DGDY*DHDQ)+G*(DFDQA1*DHDY-DFDY*DHDQ)-H*(DFDQAI
     11*DGDY-DFDY*DGDQ))/WRONSK                                         I
      YN=Y(I)+(DHDP*(P-PN)+DHDQ*(Q-QN)-H)/DHDY                          I
      IF(INFLEC.EQ.1)                                                   I
     1                YN=X(I)+(DHDP*(P-PN)+DHDQ*(Q-QN)-H)/DHDY          I
      TESTY=ABS(YN-Y(I))/Y(I) - 1.E-05                                  I
      IF(INFLEC.EQ.1)                                                   I
     1                 TESTY=ABS(YN-X(I))/X(I) - 1.E-05                 I
      IF(INFLEC.NE.1 ) Y(I)=YN                                          I
      IF(INFLEC.EQ.1)                                                   I
     1X(I)=YN                                                           I
      IF (TESTY) 45,45,80                                               I
   80 Q=QN                                                              I
      P=PN                                                              I
      GO TO 25                                                          I
   85 FORMAT(1H0,42H THIS POINT IS OUTSIDE THE MAGIC TRIANGLE,,2E15.4/  I
     2                                                              52HI
     1 THIS CONDITION IS NOT VALID FOR ANY CASE. REVISE      )          I
   90 FORMAT(1H0,42H THIS POINT IS BELOW   THE MAGIC TRIANGLE,,2E15.4/52I
     1H THIS CONDITION IS VALID ONLY FOR THE BISUPERELLIPSE/72H WITH INFI
     1LECTION(P OR Q LESS THAN 1.). SUCH A CURVE HAS BEEN GENERATED    ) I
C((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((I
      END                                                               I
```

```
      SUBROUTINE CUBIC (K)                                              J  0000
C                                                                       J  0010
C     FIT A CUBIC BETWEEN 2 STRAIGHT LINES -- RESTRICTION -- THE STRAIGHJ  0020
C                   LINES CANNOT BE VERTICAL                            J  0030
C                                                                       J  0040
      DIMENSION AA(4,4), BB(4)                                          J  0050
      COMMON /MAIN/ XIN(10),YIN(10),DELSMX,PIO2,DELS1,IHLB              J  0060
      COMMON /FOR3SS/IO,DELS,XBK(20),YBK(20),XON(500),YON(500),DYDXO(500J  0070
     1),ALPHA(500),CAPPA(500),SON(500),PIO180                           J  0080
      COMMON /SS/ NBDY1,NBODY2,TYPBDY,NBDYS                             J  0090
      DELSIN=DELS1                                                      J  0100
      DELS=DELS1                                                        J  0110
      KOUNT=0                                                           J  0120
      K=K-1                                                             J  0130
      KSTART=K                                                          J  0140
      X2=XIN(2)                                                         J  0150
      X3=XIN(3)                                                         J  0160
      Y2=YIN(2)                                                         J  0170
      Y3=YIN(3)                                                         J  0180
      SLOP2=(YIN(4)-Y3)/(XIN(4)-X3)                                     J  0190
C                                                                       J  0200
C     SETUP  4  X  4   MATRIX OF COEFFICIENTS                           J  0210
C                                                                       J  0220
      AA(1,1)=1.0                                                       J  0230
      AA(1,2)=X2                                                        J  0240
      AA(1,3)=X2*X2                                                     J  0250
      AA(1,4)=X2**3                                                     J  0260
      AA(2,1)=0.0                                                       J  0270
      AA(2,2)=1.0                                                       J  0280
      AA(2,3)=2.0*X2                                                    J  0290
      AA(2,4)=3.0*X2**2                                                 J  0300
      AA(3,1)=1.0                                                       J  0310
      AA(3,2)=X3                                                        J  0320
      AA(3,3)=X3**2                                                     J  0330
      AA(3,4)=X3**3                                                     J  0340
      AA(4,1)=0.0                                                       J  0350
      AA(4,2)=1.0                                                       J  0360
      AA(4,3)=2.0*X3                                                    J  0370
      AA(4,4)=3.0*X3**2                                                 J  0380
      DO 10 II=1,4                                                      J  0390
   10 CONTINUE                                                          J  0400
C                                                                       J  0410
C     SETUP VECTOR OF ORIGINAL CONSTANTS -- BB                          J  0420
C                                                                       J  0430
      BB(1)=Y2                                                          J  0440
      BB(2)=(Y2-YIN(1))/(X2-XIN(1))                                     J  0450
      BB(3)=Y3                                                          J  0460
      BB(4)=(YIN(4)-Y3)/(XIN(4)-X3)                                     J  0470
      NSIM=4                                                            J  0480
      KSIM=0                                                            J  0490
      CALL SIMQ (AA,BB,NSIM,KSIM)                                       J  0500
      D=BB(1)                                                           J  0510
      C=BB(2)                                                           J  0520
      A=BB(4)                                                           J  0530
      C=BB(2)                                                           J  0540
      B=BB(3)                                                           J  0550
```

```
   15 K=KSTART                                                          J   0560
      KOUNT=KOUNT+1                                                     J   0570
      XON(K+1)=XIN(2)                                                   J   0580
      YON(K+1)=YIN(2)                                                   J   0590
      DYDXO(K+1)=3.0*A*XON(K+1)**2+2.0*B*XON(K+1)+C                     J   0600
      CAPPA(K+1)=(6.0*A*XON(K+1)+2.0*B)/((1.0+DYDXO(K+1)**2)**1.5)      J   0610
      ALPHA(K+1)=ATAN(DYDXO(K+1))                                       J   0620
      DS=DELS/(1.0+.2*TANH(ABS(CAPPA(K+1))))                            J   0630
   20 K=K+1                                                             J   0640
      DXKP1=DS/(SQRT(1.0+DYDXO(K)))                                     J   0650
      IF (XIN(3).LT.XIN(2)) DXKP1=-DXKP1                                J   0660
      XON(K+1)=XON(K)+DXKP1                                             J   0670
      YON(K+1)=A*XON(K+1)**3+B*XON(K+1)**2+C*XON(K+1)+D                 J   0680
      DYDXO(K+1)=3.0*A*XON(K+1)**2+2.0*B*XON(K+1)+C                     J   0690
      CAPPA(K+1)=(6.0*A*XON(K+1)+2.0*B)/((1.0+DYDXO(K+1)**2)**1.5)      J   0700
      DS=DELS/(1.0+.2*TANH(ABS(CAPPA(K+1))))                            J   0710
      ALPHA(K+1)=ATAN(DYDXO(K+1))                                       J   0720
      SON(K+1)=SON(K)+SQRT((XON(K+1)-XON(K))**2+(YON(K+1)-YON(K))**2)   J   0730
      IF (SLOPE.GT.1.0) GO TO 25                                        J   0740
      IF (XIN(4).GE.X3.AND.XON(K+1).GT.X3) GO TO 30                     J   0750
      IF (XIN(4).LT.X3.AND.XON(K+1).LE.X3) GO TO 30                     J   0760
      GO TO 20                                                          J   0770
   25 IF (YIN(4).GE.Y3.AND.YON(K+1).GT.Y3) GO TO 30                     J   0780
      IF (YIN(4).LT.Y3.AND.YON(K+1).LE.Y3) GO TO 30                     J   0790
      GO TO 20                                                          J   0800
   30 IF (KOUNT.GT.100) GO TO 55                                        J   0810
      DELSS=DELS                                                        J   0820
      DSTEST=((XON(K+1)-XIN(3))**2+(YON(K+1)-YIN(3))**2)**.5            J   0830
      IF (ABS(DS-DSTEST).LT..01*DS) GO TO 35                            J   0840
      IF (DSTEST.LT..01*DS) GO TO 40                                    J   0850
      IF (DSTEST-.5*DS) 50,45,45                                        J   0860
   35 K=K-1                                                             J   0870
   40 XON(K+1)=XIN(3)                                                   J   0880
      YON(K+1)=YIN(3)                                                   J   0890
      GO TO 55                                                          J   0900
   45 DELS=DELS+(DS-DSTEST)/FLOAT(K-1-KSTART)                           J   0910
      IF (KOUNT.GE.10) DELS=(DELS+DELSS)/2.0                            J   0920
      GO TO 15                                                          J   0930
   50 DELS=DELS-DSTEST/FLOAT(K-KSTART)                                  J   0940
      IF (KOUNT.GE.10) DELS=(DELS+DELSS)/2.0                            J   0950
      GO TO 15                                                          J   0960
   55 DELS1=DS*1.2                                                      J   0970
      IF (DELS1.GT.DELS) DELS1=DELS                                     J   0980
      WRITE (6,65)KOUNT,A,B,C,D                                         J   0990
      WRITE (6,70)DELSIN,DELS,DELS1,DSTEST                              J   1000
      KEND=K+1                                                          J   1010
      KSTART=KSTART+1                                                   J   1020
      DO 60 I=KSTART,KEND                                               J   1030
      ALPHA(I)=ALPHA(I)/PIO180                                          J   1040
   60 CONTINUE                                                          J   1050
      RETURN                                                            J   1060
C                                                                       J   1070
C                                                                       J   1080
   65 FORMAT (1H0,17X,
     1            I4,2X,10HITERATIONS,2X,4HA = ,1PE12.5,2X,4HB = ,1PE   J   1090
     112.5,2X,4HC = ,1PE12.5,2X,4HD = ,1PE12.5)                         J   1100
   70 FORMAT(18X,10HDELS IN = ,F8.5,3X,7HDELS = ,F8.5,3X,11HDELS OUT = ,J   1110
     1F8.5,3X,9HDSTEST = ,F8.5)                                         J   1120
      END                                                               J   1130
```

```
      SUBROUTINE LEM (K)
C
C     SUBROUTINE TO CALCULATE POINTS ON A LEMNISCATE
C
      COMMON /FOR3SS/IO,DELS,XBK(20),YBK(20),XON(500),YON(500),DYDXO(500
     1),ALPHA(500),CAPPA(500),SON(500),PIO180
      COMMON /SS/ NBDY1,NBODY2,TYPBDY,NBDYS
      COMMON /MAIN/ XIN(10),YIN(10),DELSMX,PIO2,DELSI,IHUB
      DELSIN=DELSI
      K=K-1
      KSTART=K
      DELS=DELSI
      KOUNT=0
      IF (YIN(1).EQ.YIN(2)) GO TO 30
      IF (XIN(1).EQ.XIN(2)) GO TO 10
      SLOPE=(YIN(2)-YIN(1))/(XIN(2)-XIN(1))
      AROT=-TAN(SLOPE)
      GO TO 15
   10 SLOPE=99999.
      AROT=-PIO2
   15 DO 20 IROT=1,3
      XN=XIN(IROT)
      XIN(IROT)=XN*COS(AROT)-YIN(IROT)*SIN(AROT)
   20 YIN(IROT)=XN*SIN(AROT)+YIN(IROT)*COS(AROT)
   25 K=KSTART
   30 XON(K+1)=XIN(2)
      THETMX=ATAN(ABS((YIN(3)-YIN(2))/(XIN(3)-XIN(2))))
      A=SQRT(((XIN(3)-XIN(2))**2+(YIN(3)-YIN(2))**2)/(2.0*SIN(2.0*THETMX
     1)))
      YON(K+1)=YIN(2)
      CAPPA(K+1)=0.0
      DYDXO(K+1)=0.0
      ALPHA(K+1)=0.0
      KOUNT=KOUNT+1
      DSSAVE=DELS
      DS=DELS
      DTHET=DS**2/A**2
      THET=DTHET*.5
   35 R=A*SQRT(2.0*SIN(2.0*THET))
      DSCHEK=R*COS(THET)
      IF (DSCHEK.GT.1.1*DS) GO TO 40
      IF (DSCHEK.LT..9*DS) GO TO 45
      DELS=DS
      GO TO 50
   40 THET=THET-.02*DTHET
      GO TO 35
   45 THET=THET+.02*DTHET
      GO TO 35
   50 K=K+1
   55 R=A*SQRT(2.0*SIN(2.0*THET))
      XON(K+1)=XIN(2)-R*COS(THET)
      YON(K+1)=YIN(2)+R*SIN(THET)
      SON(K+1)=SON(K)+SQRT((XON(K+1)-XON(K))**2+(YON(K+1)-YON(K))**2)
      IF (ABS(SON(K+1)-SCN(K)).GT.1.05*DS) GO TO 60
      IF (ABS(SON(K+1)-SCN(K)).LT..95*DS) GO TO 65
      GO TO 70
```

```
   60 THET=THET-.02*DTHET
      GO TO 55
   65 THET=THET+.02*DTHET
      GO TO 55
   70 DYDXO(K+1)=-TAN(3.0*THET)
      ALPHA(K+1)=-3.0*THET
      CAPPA(K+1)=3.0*SQRT(SIN(2.0*THET)/2.0)/A
      DS=DELS/SQRT(1.0+ABS(CAPPA(K+1)))
      IF (ABS(DS-DSSAVE).GT..25*DSSAVE) DS=DSSAVE+SIGN(.25*DSSAVE,DS-DSS
     1AVE)
      DSSAVE=DS
      DTHET=DS*SQRT(SIN(2.0*THET)/2.0)/A
      THET=THET+DTHET
      IF (THET.LE.THETMX) GO TO 50
      IF (KOUNT.GT.50) GO TO 95
      DSTEST=((XON(K+1)-XIN(3))**2+(YON(K+1)-YIN(3))**2)**.5
      IF (DSTEST.GT.DS) GO TO 90
      IF (DSTEST.LT..0001*DS) GO TO 75
      IF (DSTEST-.5*DS) 85,85,80
   75 YON(K+1)=YIN(3)
      XON(K+1)=XIN(3)
      GO TO 95
   80 DELS=DELS-DSTEST/FLOAT(K+1-KSTART)
      GO TO 25
   85 DELS=DELS+DSTEST/FLOAT(K+1-KSTART)
      GO TO 25
   90 DELS=.8*DELS
      GO TO 25
   95 DELS1=DS*1.2
      IF (DELS1.GT.DELS) DELS1=DELS
      WRITE (6,115)KOUNT,THETMX,A
      WRITE (6,120)DELSIN,DELS,DELS1,DSTEST
      KEND=K+1
      KSTART=KSTART+1
      IF (YIN(2).EQ.YIN(1)) GO TO 105
      DO 100 KROT=KSTART,KEND
      XN=XON(KROT)
      XON(KROT)=XN*COS(AROT)+YON(KROT)*SIN(AROT)
      YON(KROT)=YON(KROT)*COS(AROT)-XN*SIN(AROT)
      ALPHA(KROT)=ALPHA(KROT)-AROT
      DYDXO(KROT)=TAN(ALPHA(KROT))
  100 CONTINUE
  105 DO 110 KAL=KSTART,KEND
  110 ALPHA(KAL)=ALPHA(KAL)/PIO180
      RETURN
C
C
  115 FORMAT(18X,I3,2X,13HITERATIONS---,3X,13HTHETMXCALC = ,F10.5,3X,8HAL
     1CALC = ,F10.5)
  120 FORMAT(18X,10HDELS IN = ,F8.5,3X,7HDELS = ,F8.5,3X,11HDELS OUT = ,
     1F8.5,3X,9HDSTEST = ,F8.5)
      END
```

```
      SUBROUTINE MIRROR (K,KTOT,KSTART,YCL)
C                                                                       M    0010
C           USED  FOR  22Y  -  2-D INLETS                               M    0030
C                                                                       M    0040
      COMMON /FOR3SS/IO,DELS,XBK(20),YBK(20),XON(500),YON(500),DYDXO(500M    0050
     1),ALPHA(500),CAPPA(500),SON(500),PIO180                           M    0060
      COMMON /SS/ NBDY1,NBODY2,TYPBDY,NBDYS                             M    0070
      DO 10 J=1,KTOT
      ISTAR=KSTART+KTOT-J
      XON(K)=XON(ISTAR)                                                 M    0110
      YON(K)=2.0*YCL-YON(ISTAR)                                         M    0120
      CAPPA(K)=-CAPPA(ISTAR)                                            M    0130
      DYDXO(K)=-DYDXO(ISTAR)                                            M    0140
      ALPHA(K)=-ALPHA(ISTAR)                                            M    0150
      SON(K)=SON(ISTAR)                                                 M    0160
      IF(J.NE.KTOT)K=K+1
   10 CONTINUE                                                          M    0170
      RETURN                                                            M    0190
C                                                                       M    0200
      END                                                               M    0220
```

```
      SUBROUTINE XYCALC(KSTART,K2,NIN)
C           -- XYCALC --
C.....GENERATES DATA FILES FOR ON-BODY POINTS.

      INTEGER SGEN
      REAL        X(300),Y(300),        C(300),S(300),SP(400)
      COMPLEX Z(300),DZ(300),D2Z,ZZ,FZTRP,DZTRP
      LOGICAL THIN,EVEN,SPGEN,DONE,BYPASS
      COMMON/SEGNO/NSEG,J,ENREED
      COMMON /MAIN/ XIN(10),YIN(10),DELSMX,PI02,DELSI,INCB
      COMMON /SPGENC/      A,DSMAX,RMAX,THIN,B,TMIN,DSENC
      COMMON /FOR3SS/IO,DELS,XBR(20),YBR(20),XON(500),YON(500),DYDXO(500
     1),ALPHA(500),CAPPA(500),SON(500),PI0180
      LOGICAL TRANSX,TRANSY,SCALEX,SCALEY ,CHANGE
      COMMON/TRANSF/ XTRAN,YTRAN,XSCALE,YSCALE,
     I  TRANSX,TRANSY,SCALEX,SCALEY,CHANGE,XTMAX
      COMMON/TRANXY/ Z
      DATA NSMAX,NSS,EMPTY,ONE/300,200,1.0E20,1.0001/
      NAMELIST /BODYIN/ Z,SI
      NAMELIST /AUXIN/ A,DSMAX,RMAX,THIN,B,NFIN,SP,NSP,DSEND,DONE,
     1                  EVEN,THIN,BYPASS

C.....INITIALIZE PROGRAM.
   10 DO 15 I=1,NSMAX
   15 Z(I)=EMPTY
      SI=0.0
      NMAX=400
      A=.50
      DSMAX=DELSMX
      THIN=.FALSE.
      RMAX=1.2
      DSEND=DSMAX
      DONE=.FALSE.
      EVEN=.FALSE.
      BYPASS=.FALSE.
      B=0.3
      TMIN=0.1
      NFIN=0
      NSP=0

C.....INPUT BODY POINTS AND BODY TYPE.
   20 READ (NIN,BODYIN)
      DO 25 I=1,NSMAX
      IF (REAL(Z(I)).EQ.EMPTY) GO TO 30
   25 NS=I
   30 S(1)=SI
C
C  TRANSFORM THE COORDINATES,Z,IF REQUIRED
C
      IF(CHANGE) CALL XYTRAN(J,NSEG,ENREED,NS)
      IBAD=SGEN(S,Z,NS)
      IF (IBAD.NE.0) WRITE (6,125)IBAD

C.....SET UP DERIVATIVES + CURVATURES.
      DO 35 I=1,NS
   35 DZ(I)=DZTRP(S,Z,S(I),NS)
```

```
      DO 40 I=1,NS                                                     N   0470
      DZZ=DZTRP(S,DZ,S(I),NS)                                          N   0480
   40 C(I)=AIMAG(CONJG(DZ(I))*DZZ)/CABS(DZ(I))**3                      N   0490
                                                                       N   0500
C.....INPUT AUXILLIARY (CONTROL) DATA.                                 N   0510
C
C  IF BYPASS = .TRUE. THE INPUT POINTS ARE THE OUTPUT POINTS
C
   45 READ (NIN,AUXIN)                                                 N   0520
      IF (NFIN.EQ.0) NFIN=NS                                           N   0530
      SFIN=S(NFIN)                                                     N   0540
      DSMAX=AMAX1(DSMAX,DSEND)                                         N   0550
      IF( .NOT. BYPASS) GO TO 47
      NSP=NS
      DO 46 II=1,NS
      X(II)=REAL(Z(II))
      Y(II)=AIMAG(Z(II))
   46 CONTINUE
C
C  SET DELS1 FOR NEXT SEGMENT. SPGEN DOES THIS WHEN BYPASS = .FALSE.
      DXLAST = X(NS) - X(NS-1)
      DYLAST = Y(NS) - Y(NS-1)
      DSLAST = SQRT(DXLAST**2. + DYLAST**2.)
      DELS1 = DSLAST
      GO TO 91
   47 IF (EVEN) GO TO 50                                               N   0560
                                                                       N   0570
C.....GENERATE BODY POINTS ON A SEGMENT.                                N   0580
      IF (.NOT.SPGEN(S,Z,C,NS,SP,NSP,SFIN,NMAX)) GO TO 130             N   0590
      GO TO 80                                                         N   0600
                                                                       N   0610
C.....GENERATE UPPER AND LOWER SURFACES TOGETHER (EVEN).                N   0620
   50 SHALF=SFIN/2.0                                                   N   0630
      IF (NSP.LT.1) GO TO 55                                           N   0640
      IF (SP(NSP).GE.SHALF/ONE) GO TO 60                               N   0650
   55 IF (.NOT.SPGEN(S,Z,C,NS,SP,NSP,SHALF,NMAX)) GO TO 140            N   0660
   60 SREM=(SFIN-SP(NSP))/ONE                                          N   0670
      DO 65 I=1,NSP                                                    N   0680
      IF (SP(I).GE.SREM) GO TO 70                                      N   0690
   65 CONTINUE                                                         N   0700
      GO TO 140                                                        N   0710
   70 IF (NSP+I-1.GT.NMAX) GO TO 150                                   N   0720
   75 I=I-1                                                            N   0730
      IF (I.LT.1) GO TO 80                                             N   0740
      NSP=NSP+1                                                        N   0750
      SP(NSP)=SFIN-SP(I)                                               N   0760
      GO TO 75                                                         N   0770
                                                                       N   0780
C.....TEST FOR FINISH.                                                  N   0790
   80 IF (DONE) GO TO 85                                               N   0800
      GO TO 45                                                         N   0810
                                                                       N   0820
C.....OUTPUT RESULTING ON-BODY POINTS.                                  N   0830
   85 DO 90 I=1,NSP                                                    N   0840
      ZZ=FZTRP(S,Z,SP(I),NS)                                           N   0850
      DZ(I)=DZTRP(S,Z,SP(I),NS)                                        N   0860
      DZZ=DZTRP(S,DZ,SP(I),NS)                                         N   0870
```

```
      C(I)= AIMAG(CONJG(DZ(I))*DZZ)/CABS(DZ(I))**3
      X(I)=REAL(ZZ)
   90 Y(I)=AIMAG(ZZ)
   91 CONTINUE
      K1=KSTART
      K2=NSP+KSTART-1
      DO 110 I=K1,K2
      II=I-KSTART+1
      XON(I)=X(II)
      YON(I)=Y(II)
      CAPPA(I)=C(II)
      IF(ABS(CAPPA(I)).LT..0001)CAPPA(I)=0.
      IF (REAL(DZ(II)).EQ.0.) GO TO 100
      DYDXO(I)=AIMAG(DZ(II))/REAL(DZ(II))
      IF (ABS(DYDXO(I)).GT.999.) GO TO 100
      IF (ABS(DYDXO(I)).LT..0001) GO TO 95
      ALPHA(I)=ATAN(DYDXO(I))*180./3.14157
      GO TO 105
   95 DYDXO(I)=0.
      ALPHA(I)=0.
      GO TO 105
  100 DYDXO(I)=999.
      ALPHA(I)=90.
  105 SON(I)=0.
      IF(I.NE.1)SON(I)=SON(I-1)+SQRT((XON(I)-XON(I-1))**2+(YON(I)-YON(I-
     11))**2)
      IF((NSEG.EQ.0.AND.II.EQ.1).OR.(J.EQ.1.AND.II.EQ.1))SON(I)=0.
      IF (I.EQ.K1) GO TO 110
      IF(XON(I).EQ.XON(I-1))CAPPA(I)=CAPPA(I-1)
      IF (XON(I).EQ.XON(I-1)) GO TO 110
      IF((DYDXO(I)-DYDXO(I-1))/(XON(I)-XON(I-1))*CAPPA(I).LT.0.)CAPPA(I
     1)=-CAPPA(I)
      IF(I.EQ.(K1+1))CAPPA(I -1)=SIGN(CAPPA(I-1),CAPPA(I))
      IF(DYDXO(I)*DYDXO(I-1).LT.0..AND.DYDXO(I).GT.0.)CAPPA(I)=-ABS(CAPP
     1A(I))
      IF (I.LT.(K1+2)) GO TO 110
      IF (DYDXO(I)*DYDXO(I-2).LT.0.) GO TO 110
      IF ((DYDXO(I-1).GT.DYDXO(I).AND.DYDXO(I-1).LT.DYDXO(I-2)).OR.(DYDX
     1O(I-1).LT.DYDXO(I).AND.DYDXO(I-1).GT.DYDXO(I-2))) GO TO 110
      IF ((CAPPA(I-1).LE.CAPPA(I).AND.CAPPA(I-1).GE.CAPPA(I-2)).OR.(CAPP
     1A(I-1).GE.CAPPA(I).AND.CAPPA(I-1).LE.CAPPA(I-2))) GO TO 110
      CAPPA(I-1)=-CAPPA(I-1)
  110 CONTINUE
      WRITE (6,115)NS,NSP,DSMAX,X(1),Y(1),X(NSP),Y(NSP)
  115 FORMAT(1H0,7X,5H--- ,
     1               58HDIRECT INTERPOLATION. FULL POINT-SPACING REQUIREMEN
     1TS MET./24X,19HNO. OF INPUT PTS.= ,I4,2X,20HNO. OF OUTPUT PTS.= ,I
     24,2X,16HDSMAX = DSEND = ,F10.2/24X,14HSTART(X,Y) = (,F10.6,1H,,F10
     3.6,1H)/24X,12HEND(X,Y) = (,F10.6,1H,,F10.6,1H))
      IF (DONE) GO TO 120
  120 RETURN
C.....ERROR MESSAGES.
  125 FORMAT(20HSGEN FAILED.  IBAD= ,I3)
  130 WRITE (6,135)
  135 FORMAT(33HSPGEN UNABLE TO COMPLETE SEGMENT )
      STOP
  140 WRITE (6,145)
  145 FORMAT(29HSPGEN UNABLE TO DO EVEN BODY )
      STOP
  150 WRITE (6,155)NSP
  155 FORMAT(44HTOO MANY POINTS FOR BOTH SURFACES.  NLOWER= ,I4)
      STOP
      END
```

```
      SUBROUTINE XYTRAN(J,NSEG,ENREED,NS)
C
C   CALLED BY SUBROUTINES SCIRCLE AND XYCALC
C   THIS SUBROUTINE TRANSLATES AND/OR SCALES INPUT BODY POINTS
C
C
C       J - INDEX OF SEGMENT LOOP IN MAIN
C    NSEG - NO. OF BODY SEGMENTS (INPUT)
C ENREED - TYPE OF BODY SEGMENT (INPUT)
C      NS - NO. OF Z VALUES INPUT (CALCULATED IN XYCALC)
C
      COMPLEX Z(300)
      COMMON/MAIN/ XIN(10),YIN(10),DELSMX,PIO2,DELS1,IHUB,DELNEW
      LOGICAL TRANSX,TRANSY,SCALEX,SCALEY,CHANGE
      COMMON/TRANSF/ XTRAN,YTRAN,XSCALE,YSCALE,
     1  TRANSX,TRANSY,SCALEX,SCALEY,CHANGE,XTMAX
      COMMON/TRANXY/ Z
      LOGICAL NOGO
C
C   ERROR FLAG -- TRANSFORMATION OF INPUT BODY POINTS IS NOT PERMITTED
C                 FOR CERTAIN BISUPERELLIPSE OPTIONS. SEE OPTIONS
C                 E, F, G, AND H IN TABLE I, TM-73728, P. 37A.
C
      NOGO = .FALSE.
      NOGO = YIN(6) .EQ. -100. .OR. YIN(6) .EQ. 200.
      IF(NOGO) GO TO 900
      IF(.NOT. TRANSX) GO TO 20
      IF((ENREED .EQ. 99) .OR. (NSEG .EQ. 0)) GO TO 16
      DO 15 ITX=1,6
      XI = XIN(ITX)
      IF(XI .LT. XTMAX) XIN(ITX) = XI + XTRAN
   15 CONTINUE
      GO TO 19
   16 CONTINUE
      DO 17 ITX=1,NS
      B = REAL(Z(ITX))
      IF(B .LT. XTMAX) B = B + XTRAN
      C = AIMAG(Z(ITX))
      Z(ITX) = CMPLX(B,C)
   17 CONTINUE
   19 IF(J .EQ. 1) WRITE(6,1000) XTRAN
   20 IF(.NOT. TRANSY) GO TO 30
      IF((ENREED .EQ. 99) .OR. (NSEG .EQ. 0)) GO TO 26
      DO 25 ITY=1,6
      YIN(ITY) = YIN(ITY) + YTRAN
   25 CONTINUE
      GO TO 29
   26 CONTINUE
      DO 27 ITY=1,NS
      B = REAL(Z(ITY))
      C = AIMAG(Z(ITY)) + YTRAN
      Z(ITY) = CMPLX(B,C)
   27 CONTINUE
   29 IF(J .EQ. 1) WRITE(6,1500) YTRAN
   30 IF(.NOT. SCALEX) GO TO 40
      IF((ENREED .EQ. 99) .OR. (NSEG .EQ. 0)) GO TO 36
```

```
      DO 35 ISX=1,6
      XIN(ISX) = XIN(ISX)*XSCALE
   35 CONTINUE
      GO TO 39
   36 CONTINUE
      DO 37 ISX=1,NS
      B = REAL(Z(ISX))*XSCALE
      C = AIMAG(Z(ISX))
      Z(ISX) = CMPLX(B,C)
   37 CONTINUE
   39 IF(J .EQ. 1) WRITE(6,2000) XSCALE
   40 IF(.NOT. SCALEY) GO TO 50
      IF((ENRREED .EQ. 99) .OR. (NSEG .EQ. 0)) GO TO 46
      DO 45 ISY=1,6
      YIN(ISY) = YIN(ISY)*YSCALE
   45 CONTINUE
      GO TO 49
   46 CONTINUE
      DO 47 ISY=1,NS
      B = REAL(Z(ISY))
      C = AIMAG(Z(ISY))*YSCALE
      Z(ISY) = CMPLX(B,C)
   47 CONTINUE
   49 IF(J .EQ. 1) WRITE(6,2500) YSCALE
   50 CONTINUE
      RETURN
  900 WRITE(6,3000)
      STOP
C
C    FORMAT STATEMENTS
C
 1000 FORMAT(1H0,5X,'+ + + X - VALUES OF BODY POINTS TRANSLATED',
     1       1PE12.5,'  UNITS')
 1500 FORMAT(1H0,5X,'+ + + Y - VALUES OF BODY POINTS TRANSLATED',
     1       1PE12.5,'  UNITS')
 2000 FORMAT(1H0,5X,'+ + + X - VALUES OF BODY POINTS SCALED',
     1       ' BY A FACTOR OF ',1PE12.5)
 2500 FORMAT(1H0,5X,'+ + + Y - VALUES OF BODY POINTS SCALED',
     1       ' BY A FACTOR OF ',1PE12.5)
 3000 FORMAT(3X,'*** ERROR IN XYTRAN *** - USER ATTEMPTED',1X,
     1       'TO TRANSFORM INPUT POINTS FOR AN ILLEGAL CASE')
      END
```

```
      LOGICAL FUNCTION SPGEN (S,Z,C,NS,SP,NSP,SFIN,NMAX)                Q  0000
C.....GENERATES TABLE SP HAVING VALUES OF PARAMETER S AS WIDELY SPACED  Q  0010
C     AS POSSIBLE AND YET SATISFYING THE FOLLOWING CONDITIONS ON DS:    Q  0020
C        1   NSP .LE. NMAX                                              Q  0030
C        2   DS  .LE. A/C(S)     (C=CURVATURE)                          Q  0040
C        3   DS  .LE. DSMAX                                             Q  0050
C        4A  DS(I) .LE. DS(I-1)*RMAX                                    Q  0060
C        4B  DS(I) .GE. DS(I-1)/RMAX                                    Q  0070
C     FOR THIN SECTIONS, AN ADDITIONAL CONDITION IS:                    Q  0080
C            DS  .LE. B*TLOC     (TLOC=LOCAL THICKNESS)                 Q  0090
C.....SPGEN = .TRUE. IF ALL CONDITIONS HAVE BEEN SATISFIED.             Q  0100
                                                                        Q  0110
      REAL S(NS),C(NS),SP(NMAX)                                         Q  0120
      COMPLEX Z(NS),FZTRP                                               Q  0130
      LOGICAL THIN,FIN                                                  Q  0140
      COMMON /SPGENC/      A,DSMAX,RMAX,THIN,B,TMIN,DSEND               Q  0150
      COMMON /MAIN/ XIN(10),YIN(10),DELSMX,PIO2,DELS1,IHUB              Q  0160
      DATA ONE,CMIN/1.0001,1.0E-67/                                     Q  0170
                                                                        Q  0180
C.....INITIALIZATION SECTION.                                           Q  0190
      SPGEN=.FALSE.                                                     Q  0200
      J1=MAX0(NSP,2)+1                                                  Q  0210
      IF (NSP.GT.1) GO TO 15                                            Q  0220
      IF (NSP.LT.1) SP(1)=S(1)                                          Q  0230
      DS1=      DELS1                                                   Q  0240
   10 SP(2)=SP(1)+DS1                                                   Q  0250
                                                                        Q  0260
C.....BEGIN MAIN LOOP.                                                  Q  0270
   15 DO 45 J=J1,NMAX                                                   Q  0280
      L=J                                                               Q  0290
   20 I=L                                                               Q  0300
   25 DSLAST=SP(I-1)-SP(I-2)                                            Q  0310
      SBAR=SP(I-1)+DSLAST/2.0                                           Q  0320
      CA=AMAX1(CMIN,ABS(FNTRP(S,C,SBAR,NS)))                            Q  0330
      DSLIM=AMIN1(DS1   ,DSLAST*RMAX)                                   Q  0340
      IF (.NOT.THIN) GO TO 30                                           Q  0350
      TLOC=CABS(FZTRP(S,Z,SBAR,NS)-FZTRP(S,Z,S(NS)-SBAR,NS))            Q  0360
      DSLIM=AMIN1(DSLIM,B*AMAX1(TLOC,TMIN))                             Q  0370
   30 DSFIN=SFIN-SP(I-1)                                                Q  0380
      NEVEN=DSFIN/DSLIM/ONE+1.0                                         Q  0390
      DSEVEN=DSFIN/FLOAT(NEVEN)                                         Q  0400
      DS=AMIN1(A/CA,DSEVEN)                                             Q  0410
      IF (I.NE.J) DS=AMIN1(DS,DSLAST/RMAX)                              Q  0420
                                                                        Q  0430
C.....CALCULATED VALUE OF DS SATISFIES CONDITIONS 2 THRU 4A.TEST FOR 4B.Q  0440
      IF (DS.GE.DSLAST/RMAX) GO TO 40                                   Q  0450
                                                                        Q  0460
C.....IF CONDITION 4B IS NOT SATISFIED, RE-DO EARLIER INTERVALS         Q  0470
C.....USING SMALLER VALUES OF DS.  IF RE-DOING ALL INTERVALS WON'T      Q  0480
C.....WORK, START OVER USING SMALLER STARTING VALUE OF DS (DS1).        Q  0490
   35 L=L-1                                                             Q  0500
      IF (L.GE.J1) GO TO 20                                             Q  0510
      IF (NSP.GT.1) RETURN                                              Q  0520
      DS1=DS1/RMAX                                                      Q  0530
      GO TO 10                                                          Q  0540
                                                                        Q  0550
```

```
C.....IF CONDITIONS 2 THRU 48 ARE SATISFIED, TEST FOR FINISH.            Q   0560
   40 SP(I)=SP(I-1)+DS                                                    Q   0570
      FIN=SFIN/SP(I).LE.ONE                                               Q   0580
      IF (FIN.AND.DS.GT.CSEND) GO TO 35                                   Q   0590
      IF (FIN) GO TO 50                                                   Q   0600
      IF (I.GE.J) GO TO 45                                                Q   0610
      I=I+1                                                               Q   0620
      GO TO 25                                                            Q   0630
   45 CONTINUE                                                            Q   0640
C.....SPGEN =.FALSE. IF CONDITION 1 CANNOT BE SATISFIED.                  Q   0650
      RETURN                                                              Q   0660
                                                                          Q   0670
C.....IF CONDITIONS ARE SATISFIED, UPDATE NSP.                            Q   0680
   50 NSP=I                                                               Q   0690
      DELSI=DS                                                            Q   0700
      SPGEN=.TRUE.                                                        Q   0710
      RETURN                                                              Q   0720
      END                                                                 Q   0730
```

```
      INTEGER FUNCTION SGEN (S,F,NS)                                    O   0000
C.....GENERATES THE PARAMETER ARRAY S FOR THE SET OF POINT-PAIRS F SUCH O   0010
C.....THAT S(I) GIVES THE LINE INTEGRAL ON THE CURVE OF FZTRP (S,F,X,NS)O   0020
C.....WHEN X=S(I).                                                      O   0030
      REAL S(NS)                                                        O   0040
      COMPLEX F(NS),DZTRF                                               O   0050
      DATA MAX,N,FN,TEST/4,10,10.0,0.01/                                O   0060
      DO 10 I=2,NS                                                      O   0070
   10 S(I)=S(I-1)+CABS(F(I)-F(I-1))                                     O   0080
      DO 30 K=1,MAX                                                     O   0090
      SGEN=0                                                            O   0100
      DO 25 I=2,NS                                                      O   0110
      DS=S(I)-S(I-1)                                                    O   0120
      DARG=DS/FN                                                        O   0130
      ARG0=S(I-1)-DARG/2.0                                              O   0140
      SUM=0.0                                                           O   0150
      DO 15 J=1,N                                                       O   0160
      ARG=ARG0+FLOAT(J)*DARG                                            O   0170
   15 SUM=SUM+(CABS(DZTRP(S,F,ARG,NS))-1.0)                             O   0180
      SUM=SUM/FN                                                        O   0190
      ERROR=ABS(SUM)                                                    O   0200
      DS=DS*SUM                                                         O   0210
      DO 20 J=I,NS                                                      O   0220
   20 S(J)=S(J)+DS                                                      O   0230
      IF (ERROR.GT.TEST.AND.SGEN.EQ.0) SGEN=I                           O   0240
   25 CONTINUE                                                          O   0250
      IF (SGEN.EQ.0) RETURN                                             O   0260
   30 CONTINUE                                                          O   0270
C.....SGEN=INDEX IF IT DOESN'T CONVERGE.                                 O   0280
      RETURN                                                            O   0290
      END                                                               O   0300
```

```
      COMPLEX FUNCTION DZTRP (A,F,X,NA)                                 P   0000
C.....COMPLEX DERIVATIVE EVALUATION FOR DOUBLE 3-POINT INTERPOLATION.   P   0010
      COMPLEX F(NA)                                                     P   0020
      COMMON /NTRPC3/ I1,I2,C(4)                                        P   0030
C.....FIRST EVALUATE FUNCTION COEFFICIENTS.                              P   0040
      CALL FNTRPA (A,X,NA)                                              P   0050
      CALL DNTRPC                                                       P   0060
C.....THEN EVALUATE FUNCTION VALUE.                                      P   0070
      DZTRP=0.0                                                         P   0080
      J=0                                                               P   0090
      DO 10 I=I1,I2                                                     P   0100
      J=J+1                                                             P   0110
   10 DZTRP=DZTRP+C(J)*F(I)                                             P   0120
      RETURN                                                            P   0130
      END                                                               P   0140
```

```
      SUBROUTINE DNTRPC                                                 R    0000
C.....CALCULATION OF C COEFFICIENTS FOR DERIVATIVES OF DOUBLE           R    0010
C.....3-POINT INTERPOLATION.                                            R    0020
      COMMON /NTRPC1/ L,I,A11,A12,A13,A14,A22,A23,A24,A33,A34,A44       R    0030
      COMMON /NTRPC3/ I1,I2,C1,C2,C3,C4                                 R    0040
      IF (L.LE.1) GO TO 25                                              R    0050
      IF (L-3) 20,15,10                                                 R    0060
C.....FOR DOUBLE 3-POINT INTERPOLATION.                                 R    0070
   10 C1=+(A22+A33+A22)/A23*A33/A12/A13                                 R    0080
      C4=-(A33+A22+A33)/A23*A22/A34/A24                                 R    0090
      P=A23*A23                                                         R    0100
      C2=-((A11+A33+A11)*A33/A12+(A33*A44+A22*A44+A22*A33)/A24)/P       R    0110
      C3=+((A44+A22+A44)*A22/A34+(A22*A11+A33*A11+A33*A22)/A13)/P       R    0120
      GO TO 30                                                          R    0130
C.....FOR SIMPLE 3-POINT INTERPOLATION.                                 R    0140
   15 C1=+(A33+A22)/A12/A13                                             R    0150
      C2=-(A33+A11)/A12/A23                                             R    0160
      C3=+(A22+A11)/A13/A23                                             R    0170
      GO TO 30                                                          R    0180
C.....FOR 2-POINT INTERPOLATION.                                        R    0190
   20 C1=1.0/A12                                                        R    0200
      C2=-C1                                                            R    0210
      GO TO 30                                                          R    0220
C.....ONLY ONE TABLE VALUE GIVEN.                                       R    0230
   25 C1=0.0                                                            R    0240
   30 I1=I                                                              R    0250
      I2=I+L-1                                                          R    0260
      RETURN                                                            R    0270
      END                                                               R    0280
```

```
      FUNCTION FNTRP (A,F,X,NA)                                         S    0000
C.....FUNCTION EVALUATION FOR DOUBLE 3-POINT INTERPOLATION.             S    0010
      REAL F(NA)                                                        S    0020
      COMMON /NTRPC2/ I1,I2,C(4)                                        S    0030
C.....FIRST EVALUATE FUNCTION COEFFICIENTS.                             S    0040
      CALL FNTRPA (A,X,NA)                                              S    0050
      CALL FNTRPC                                                       S    0060
C.....THEN EVALUATE FUNCTION VALUE.                                     S    0070
      ENTRY FNTRP1 (F)                                                  S    0080
      FNTRP=0.0                                                         S    0090
      J=0                                                               S    0100
      DO 10 I=I1,I2                                                     S    0110
      J=J+1                                                             S    0120
   10 FNTRP=FNTRP+C(J)*F(I)                                             S    0130
      RETURN                                                            S    0140
      END                                                               S    0150
```

```
      COMPLEX FUNCTION FZTRP (A,F,X,NA)                                 T   0000
C.....COMPLEX FUNCTION EVALUATION BY DOUBLE 3-POINT INTERPOLATION.      T   0010
      COMPLEX F(NA)                                                     T   0020
      COMMON /NTRPC2/ I1,I2,C(4)                                        T   0030
C.....FIRST EVALUATE FUNCTION COEFFICIENTS.                             T   0040
      CALL FNTRPA (A,X,NA)                                              T   0050
      CALL FNTRPC                                                       T   0060
C.....THEN EVALUATE FUNCTION VALUE.                                     T   0070
      FZTRP=0.0                                                         T   0080
      J=0                                                               T   0090
      DO 10 I=I1,I2                                                     T   0100
      J=J+1                                                             T   0110
   10 FZTRP=FZTRP+C(J)*F(I)                                             T   0120
      RETURN                                                            T   0130
      END                                                               T   0140
```

```
      SUBROUTINE FNTRPC                                                 U   0000
C.....CALCULATION OF C COEFFICIENTS FOR FUNCTION VALUES BY DOUBLE       U   0010
C.....3-POINT INTERPOLATION.                                            U   0020
      COMMON /NTRPC1/L,I,A11,A12,A13,A14,A22,A23,A24,A33,A34,A44        U   0030
      COMMON /NTRPC2/ I1,I2,C1,C2,C3,C4                                 U   0040
      IF (L.LE.1) GO TO 25                                              U   0050
      IF (L-3) 20,15,10                                                 U   0060
C.....FOR DOUBLE 3-POINT INTERPOLATION.                                 U   0070
   10 C1=+A33/A23*A22/A12*A33/A13                                       U   0080
      C4=-A22/A23*A33/A34*A22/A24                                       U   0090
      P2=A33/A23*A11/A23                                                U   0100
      P3=A22/A23*A44/A23                                                U   0110
      C2=-A33*(P2/A12+P3/A24)                                           U   0120
      C3=+A22*(P3/A34+P2/A13)                                           U   0130
      GO TO 30                                                          U   0140
C.....FOR SIMPLE 3-POINT INTERPOLATION.                                 U   0150
   15 C1=+A22/A12*A33/A13                                               U   0160
      C2=-A11/A12*A33/A23                                               U   0170
      C3=+A11/A13*A22/A23                                               U   0180
      GO TO 30                                                          U   0190
C.....FOR 2-POINT INTERPOLATION.                                        U   0200
   20 C1=+A22/A12                                                       U   0210
      C2=-A11/A12                                                       U   0220
      GO TO 30                                                          U   0230
C.....ONLY ONE TABLE VALUE GIVEN.                                       U   0240
   25 C1=1.0                                                            U   0250
   30 I1=I                                                              U   0260
      I2=I+L-1                                                          U   0270
      RETURN                                                            U   0280
      END                                                               U   0290
```

```
      SUBROUTINE FNTRPA (A,X,NA)                                      V  0000
C.....COMMON SUBROUTINE EVALUATES A COEFFICIENTS IN DOUBLE            V  0010
C.....3-POINT INTERPOLATIONS.                                         V  0020
C        L=NO. OF POINTS IN THE FIT                                   V  0030
C        I=INDEX TO FIRST POINT                                       V  0040
      REAL A(NA)                                                      V  0050
      COMMON /NTRPC1/ L,I,A11,A12,A13,A14,A22,A23,A24,A33,A34,A44     V  0060
C.....GET I AND L BY TABLE LOOK-UP.                                   V  0070
      L=LIMIT (1,NA,3)                                                V  0080
      M=MAX0(1,NA-2)                                                  V  0090
      CALL TLU (A,X,NA,J)                                             V  0100
      IF (J.EQ.LIMIT(2,J,M)) L=4                                      V  0110
      I=LIMIT(1,J-1,M)                                                V  0120
C.....CALCULATE A-ARRAY.                                              V  0130
      A11=A(I)                                                        V  0140
      A22=A(I+1)                                                      V  0150
      A33=A(I+2)                                                      V  0160
      IF (L.NE.4) IF (L-2) 20,15,10                                   V  0170
      A44=A(I+3)                                                      V  0180
      A14=A11-A44                                                     V  0190
      A24=A22-A44                                                     V  0200
      A34=A33-A44                                                     V  0210
      A44=X-A44                                                       V  0220
   10 A13=A11-A33                                                     V  0230
      A23=A22-A33                                                     V  0240
      A33=X-A33                                                       V  0250
   15 A12=A11-A22                                                     V  0260
      A22=X-A22                                                       V  0270
      A11=X-A11                                                       V  0280
   20 RETURN                                                          V  0290
      END                                                             V  0300
```

```
      SUBROUTINE TLU (TABLE,ARG,N,I)                               W  0000
C.....TABLE LOOK UP:  FINDS I SUCH THAT                            W  0010
C     ARG.GE.TABLE(I).AND.ARG.LT.TABLE(I+1)                        W  0020
C       IF I=0, ARG.LT.TABLE(1)                                    W  0030
C       IF I=N, ARG.GE.TABLE(N)                                    W  0040
      REAL TABLE(N)                                                W  0050
      I=LIMIT(1,I,N)                                               W  0060
      IF (ARG.GE.TABLE(I)) GO TO 15                                W  0070
C.....DESCEND IN TABLE.                                            W  0080
   10 I=I-1                                                        W  0090
      IF (I.LE.0) RETURN                                           W  0100
      IF (ARG.GE.TABLE(I)) RETURN                                  W  0110
      GO TO 10                                                     W  0120
C.....ASCEND IN TABLE.                                             W  0130
   15 IF (I.GE.N) RETURN                                           W  0140
      IF (ARG.LT.TABLE(I+1)) RETURN                                W  0150
      I=I+1                                                        W  0160
      GO TO 15                                                     W  0170
      END                                                          W  0180
```

```
      FUNCTION LIMIT (I,J,K)                                       X  0000
C.....INTEGER FUNCTION LIMITS J BETWEEN I AND K.                    X  0010
      LIMIT=I                                                      X  0020
      IF (J.LT.LIMIT) RETURN                                       X  0030
      LIMIT=K                                                      X  0040
      IF (J.GT.LIMIT) RETURN                                       X  0050
      LIMIT=J                                                      X  0060
      RETURN                                                       X  0070
      END                                                          X  0080
```

```
      SUBROUTINE WPUNCH                                                  Y  0010
      COMMON /FOREOD/ IGEOMF,ISIGF,ICURVN,NONEWF,IVORT,AL
      COMMON /SS/ NBDY1,NBODY2,TYPBDY,NBDYS                              Y  0020
      COMMON /FOR3SS/IO,DELS,XBK(20),YBK(20),XON(500),YON(500),DYDXO(500Y  0030
     1),ALPHA(500),CAPPA(500),SON(500),PIO180                            Y  0040
      COMMON /MWRYE/ IFLAG,NDY4,PROG,TITLE(9),BODIES(4),IDENT,YLO(25),YHY  0050
     1I(25),NDY(25),XRAK(25),NBDPTS(9),NOG,NRAKES
      COMMON/ LOT/XMIN,YMIN,ORD,EXEP,XX,YY,LPNCHO,IPLOYA,MM              Y  0070
      COMMON/ONVORT/NLF(10)
      COMMON/Y24Y/ IPAR,IFSY,ISND,IFLLL
      DIMENSION YOFF(200), XOFF(200), FI(500)                            Y  0080
      DIMENSION X(25),Y(25)                                              Y  0090
      COMMON /MNSD/ NNSD,NSDBDY(10)                                      Y  0100
      DATA BODYD/6H-BODY /,IFLAG1/4H I11/,IFLAG2/4H I 1/,Y22Y/6H   23Y/,Y  0110
     1IFLG2A/1H /,IFLG2B/1H1/                                            Y  0120
C                                                                        Y  0130
C     IF YLO AND YHI ARE READ IN AS ZERO,CALCULATE THEM FOR THAT RAKE    Y  0140
C   FOR WAKES BETWEEN BODIES =1 AND =2 ONLY
C                                                                        Y  0160
C     FIND HIGHLIGHT ON THE SHROUD                                       Y  0170
C                                                                        Y  0180
      NINE=9
      NEIGHT=8
      I21=21
      IONE=1
      ITWO=2
      IZER=0
      NB=NBDPTS(1)+1                                                     Y  0190
      IF(IFLAG.EQ.1)NB=1                                                 Y  0200
      NE=NBDPTS(2)                                                       Y  0210
C ( SET JMIN=LAST SHROUD PT., IN CASE X NEVER INCREASES ON SHROUD(VTOL)) Y  0220
      JMIN=NE                                                            Y  0230
      DO 10 I=NB,NE                                                      Y  0240
      IF (XON(I+1).LT.XON(I)) GO TO 10                                   Y  0250
      JMIN=I                                                             Y  0260
      GO TO 15                                                           Y  0270
   10 CONTINUE                                                           Y  0280
   15 DO 20 I=1,NE                                                       Y  0290
   20 FI(I)=I                                                            Y  0300
      NOFF=0                                                             Y  0310
      IF (NRAKES.EQ.0) GO TO 70                                          Y  0320
      DO 65 I=1,NRAKES                                                   Y  0330
      NLO=NOFF+1                                                         Y  0340
      NOFF=NLO+NDY(I)                                                    Y  0350
      ENDY=NDY(I)                                                        Y  0360
      IF (YHI(I).EQ.0.0.OR.YLO(I).EQ.0.0) GO TO 25                       Y  0370
      GO TO 35                                                           Y  0380
   25 IF (YHI(I).NE.0.0) GO TO 30                                        Y  0390
      CALL SINTP (XON(NB),YON(NB),JMIN-NB+1,XRAK(I),YH)                  Y  0400
      CALL SINTP (XON(NB),FI(NB),JMIN-NB+1,XRAK(I),FII)                  Y  0410
      IF=FII                                                             Y  0420
      DS=SQRT((XON(IF)-XON(IF+1))**2+(YON(IF)-YON(IF+1))**2)             Y  0430
      YHI(I)=YH-DS                                                       Y  0440
   30 IF (YLO(I).NE.0.0.OR.XRAK(I).LT.XON(1)) GO TO 35                   Y  0450
      CALL SINTP (XON,YON,NBDPTS(1),XRAK(I),YL)                          Y  0460
      CALL SINTP (XON,FI,NBDPTS(1),XRAK(I),FII)                          Y  0470
```

```
      IF=FII                                                          Y   0480
      DS=SQRT((XON(IF)-XCN(IF+1))**2+(YON(IF)-YON(IF+1))**2)          Y   0490
      YLO(I)=YL+DS                                                    Y   0500
   35 DYI=(YHI(I)-YLO(I))/ENDY                                        Y   0510
      K=0                                                             Y   0520
      DO 55 J=NLO,NOFF                                                Y   0530
      DJM=J-NLO                                                       Y   0540
      XOFF(J)=XRAK(I)                                                 Y   0550
      YOFF(J)=YLO(I)+DYI*DJM                                          Y   0560
      YMAN=YY*ORD+YMIN
      IF(LPNCHO.EQ.2) YOFF(J)=YMAN-(YOFF(J)-YMAN)
      IF (XOFF(J)-XMIN) 55,40,40                                      Y   0570
   40 IF (XOFF(J)-XX*EXEP-XMIN) 45,45,55                              Y   0580
   45 IF (YOFF(J)-YY*ORD-YMIN) 50,50,55                               Y   0590
   50 K=K+1                                                           Y   0600
      X(K)=XOFF(J)                                                    Y   0610
      Y(K)=YOFF(J)                                                    Y   0620
   55 CONTINUE                                                        Y   0630
C                                                                     Y   0640
C((((((   PLOT OFF-BODY POINTS (RAKES)                                Y   0650
C                                                                     Y   0660
   60 CALL LINE(X,Y,K,1,-1,0,XMIN,EXEP,YMIN,ORD)                      Y   0670
   65 CONTINUE                                                        Y   0680
C                                                                     Y   0820
C((((((  PUNCH OPTION ((((((                                          Y   0830
C                                                                     Y   0840
      IF (LPNCHO.EQ.0) RETURN
   70 NTBDY=NBDYS+NNSD+1-IVORT                                        Y   0690
      NLOOP=2-IVORT                                                   Y   0700
      IF (NBDYS.EQ.3.AND.PROG.NE.T22Y.AND.IVORT.EQ.0) GO TO 75
      GO TO 80                                                        Y   0720
   75 NTBDY=NTBDY+1                                                   Y   0730
      NLOOP=3                                                         Y   0740
   80 K=0                                                             Y   0750
      IF(PROG.EQ.T22Y) NLOOP=1
      DO 110 I=1,NLOOP                                                Y   0760
      M=NTBDY-I+1                                                     Y   0770
      IFLAGG=IFLAG2                                                   Y   0780
      IF (M.EQ.NTBDY               .OR.NBDYS.EQ.3.AND.M.EQ.3) IFLAGG=IFLY   0790
     1AG1                                                             Y   0800
      IF (PROG.EQ.T22Y) GO TO 92
      WRITE (17,115)(TITLE(L),L=1,9),M,BODYD,IDENT                    Y   0860
      WRITE (17,120)M,IFLAGG,NOG,IVORT,IDENT                          Y   0870
   90 CONTINUE                                                        Y   0920
                          WRITE (17,125)IDENT
  92  NA=1
      NS=0                                                            Y   0950
      IF (I.NE.1) K=1                                                 Y   0960
      DO 105 J=1,M                                                    Y   0970
      IF (J.GT.NBDYS.AND.NNSD.NE.0) GO TO 95                          Y   0980
      NB=NBDPTS(J)                                                    Y   0990
      GO TO 100                                                       Y   1000
   95 NS=NS+1                                                         Y   1010
      IF (NS.GT.NNSD) NSDBDY(NS)=NBDPTS(NBDYS+1)-NBDPTS(NBDYS)        Y   1020
      NB=NSDBDY(NS)+NA-1                                              Y   1030
  100 NP=NB-NA+1                                                      Y   1040
      IF(PROG.NE.T22Y)GOTO102
```

```
            LASBOD=0
            NO1BOD=2
            IF(J.EQ.1) NO1BOD=1
            IF(J.EQ.M)LASBOD=1
            NNN = I - NLF(J)
            WRITE(17,130) J,IGEOMF,NNN,(TITLE(L),L=1,5),IPAR,IFST,ISND,
           1,NO1BOD,IVOPT,LASBCD,IONE
      C    T R A N S F
      102   CALL WRTXY (NP,IDENT,J,K,XON,YON,NA,NB,PROG)
            NA=NB+1                                                     Y    1070
        105 CONTINUE                                                    Y    1080
            IF(PROG.NE.T22Y                    )GOTO107
            WRITE(17,140) (TITLE(L),L=1,6),NEYGHT
            V2=2.
            WRITE(17,145) AL,ICNE,IONE,V2,IZER,IFLLL,NINE
      107   K=0
            NA=1                                                        Y    1100
            J=0                                                         Y    1120
            NB=NOFF                                                     Y    1110
            IF(PROG.NE.T22Y)GOTO108
            IF(NB.GT.100)NB=100
            I6=0
            IF(NOFF.LE.100)I6=1
                          WRITE(17,150) IONE,(TITLE(L),L=1,6),I6,I21
      C  T R A N S F
      108   CALL WRTXY (NOFF,IDENT,J,K,XOFF,YOFF,NA,NB,PROG)
            IF (PROG.NE.T22Y) GO TO 110                                 Y    1150
            IF(NOFF.LE.100)GOTO110
            WRITE(17,150)ITWO,(TITLE(L),L=1,6),IONE,I21
      C    T R A N S F
            NOF=NOFF-100
            NA=NB+1
            NB=NOFF
            CALL WRTXY(NOF,IDENT,J,K,XOFF,YOFF,NA,NB,PROG)
        110 CONTINUE                                                    Y    1200
            RETURN                                                      Y    1210
      C                                                                 Y    1220
      C     FORMATS                                                     Y    1230
      C                                                                 Y    1240
      C                                                                 Y    1250
      C                                                                 Y    1260
        115 FORMAT (     9A6,I1,A6,2X,A6)                               Y    1270
        120 FORMAT (     I1,A4,I1,8X,I1,47X,A6,I1X)                     Y    1280
        125 FORMAT (     62X,A6,11X)                                    Y    1290
        130 FORMAT(3(I1,2X),1X,5A6,3X,3(I1,2X),6X,I1,5X,I1,2X,I1,3X,I1)
        135 FORMAT (     3H0.0,7X,3H0.0,7X,3H90.)                       Y    1310
        140 FORMAT(10X,6A6,25X,I1)
        145 FORMAT(5X,F10.2,19X,I1,9X,I1,10X,F9.3,5X,I1,I1,I1)
        150 FORMAT(I1,8X,6A6,22X,I1,2X,I2)
       155  FORMAT(8F10.3)
            END                                                         Y    1320
```

```
      SUBROUTINE WRTXY (NP,IDENT,J,K,X,Y,NA,NB,PROG)                    ZO  0000
C                                                                       ZO  0010
C     WRITE X AND Y COORDINATES                                         ZO  0020
C                                                                       ZO  0030
      DIMENSION X(1), Y(1)                                              ZO  0040
      COMMON /FOREOD/ IGEOMF,ISIGF,ICURVN,NONEWF,IVORT                  ZO  0050
      COMMON/OMVORT/ NLF(10)
      DIMENSION V(8)
      DATA T22Y/6H   23Y/
      DATA V/6H(     ,6H      ,6HF10.5,,6H      ,6HX,     ,6H4X,I1,,    ZO  0060
     16H2X,I1,,6H3X,I1)/
      IF (PROG.NE.T22Y) GO TO 10
      LASY=1
      NFUL =(NB-NA+1)/6
      NFULL=NFUL *6
      NREST=NA+NFULL-1
      NDIF=NB-NA+1-NFULL
      NREST1=NREST+1
      IT=3
      NAA=NA-6
      DO 2 LC=1,NFUL
      NAA=NAA+6
      NSTOP=NAA+5
      IF(NSTOP.EQ.NB)GOTO5
      WRITE(17,40)(X(L),L=NAA,NSTOP),IT
 2     CONTINUE
      IF(NDIF.EQ.0)GOTO6
      NDIFM=(6-NDIF)*10
      ENCODE(6,56,DUMP)NDIF
      ENCODE(6,56,DUMP2)NDIFM
      DECODE(6,59,DUMP)VEE
      DECODE(6,59,DUMP2)VE2
      V(2)=VEE
      V(4)=VE2
      WRITE(17,V) (X(LL),LL=NREST1,NB),NDIF,LASY,IT
      GOTO6
 5    WRITE(17,45)(X(L),L=NAA,NSTOP),LASY,IT
  45  FORMAT(6F10.5,7X,I1,3X,I1)
 6     IT=4
      NAA=NA-6
      DO 300 LC=1,NFUL
      NAA=NAA+6
      NSTOP=NAA+5
      IF(NSTOP.EQ.NB)GOTO7
      WRITE(17,40)(Y(L),L=NAA,NSTOP),IT
 300  CONTINUE
      IF(NDIF.EQ.0)RETURN
      WRITE(17,V) (Y(LL),LL=NREST1,NB),NDIF,LASY,IT
      RETURN
 7    WRITE(17,45)(Y(L),L=NAA,NSTOP),LASY,IT
      RETURN
 10   WRITE (17,25)IGEOMF,ISIGF,ICURVN,NONEWF,NP,IDENT                  ZO  0070
      IF(J.EQ.0) NNN=0
      IF(J.EQ.0) GO TO 15
      NNN=NLF(J)
   15 WRITE (17,30)J,K,NNN,IDENT                                        ZO  0140
```

```
   20 IF (K.EQ.1) RETURN                                    Z0 0150
      WRITE (17,35)(X(L),L=NA,NB)                          Z0 0160
      WRITE (17,35)(Y(L),L=NA,NB)                          Z0 0170
      RETURN                                                Z0 0180
C                                                           Z0 0190
C     FORMAT STATEMENTS                                     Z0 0200
C                                                           Z0 0210
C                                                           Z0 0220
C                                                           Z0 0230
   25 FORMAT (4I1,3X,I3,52X,A6,11X)                         Z0 0240
   30 FORMAT (     3(9X,I1),32X,A6,11X)                    Z0 0250
   35 FORMAT(6E13.8)                                        Z0 0260
  40  FORMAT(6F10.5,11X,I1)
 55   FORMAT(1H+,64X,I1,2X,I1,3X,I1)
  56  FORMAT(I6)
  69  FORMAT(A6)
      END                                                   Z0 0290
```

```
      SUBROUTINE AREAA                                                  Z1  0000
      COMMON /MWRTE/ IFLAG,NDY4,PROG,TITLE(9),BODIES(4),IDENT,YLO(25),YHA    0100
     1I(25),NDY(25),XRAK(25),NBDPTS(9),NO6,NRAKES
      COMMON /SS/ NBDY1,NBODY2,TYPBDY,NBDYS,DZTEST
      COMMON /FOR3SS/IO,CELS,XBK(20),YBK(20),XON(500),YON(500),DYDXO(500Z1  0020
     1),ALPHA(500),CAPPA(500),SON(500),PIO180                           Z1  0030
      COMMON /MNSD/ NNSD,NSDBDY(10)                                     Z1  0040
      COMMON/ LOT/XMIN,YMIN,ORD,EXEP,XX,YY,LPNCHD,IPLOTA,MM             Z1  0050
      COMMON/BURLEY/ELREF,AREF
      DIMENSION JMAX(20), JMIN(20),                   AREAS(20), YAR(20)Z1  0060
     1,XXA(200),           DISC(200),ANULUS(200)                        Z1  0070
      PI=3.14159265                                                     Z1  0080
      JPLA=0                                                            Z1  0090
      NBP1=NBDPTS(NBDYS-1)+1
      IF(NBDYS.EQ.1)NBP1=1
      NBP2=NBDPTS(NBDYS)
      DO 10 J=NBP1,NBP2
      JJSS=J
C((((((  .LT. CHANGED TO LE. TO AVOID AREA PLOTS FROM GOING UP VSTOL LIPZ1  0130
      IF (XON(J).LE.XON(J+1)) GO TO 15                                  Z1  0140
   10 CONTINUE                                                          Z1  0150
   15 WRITE(6,98)
      IF(ELREF.NE.1.0) WRITE(6,100) ELREF
      IF(AREF.NE.1.0) WRITE(6,110) AREF
      WRITE (6,75)                                                      Z1  0160
      IF(NBDYS.LE.2) GOTO40
C                                                                       Z1  0180
C     SEARCH FOR MINIMUM AND MAXIMUM X ON EACH BODY
C                                                                       Z1  0200
      NBDSM1=NBDYS-1
      NB=1
      DO 35 I=1,NBDSM1
      NE=NBDPTS(I)
      XMIN=XON(NB)                                                      Z1  0240
      JMIN(I)=NB                                                        Z1  0250
      XMAX=XON(NB)                                                      Z1  0260
      JMAX(I)=NB                                                        Z1  0270
      NEM1=NE-1                                                         Z1  0280
      DO 30 J=NB,NEM1                                                   Z1  0290
      IF (XON(J).GT.XMAX) GO TO 20                                      Z1  0300
      IF (XON(J).LT.XMIN) GO TO 25                                      Z1  0310
      GO TO 30                                                          Z1  0320
   20 XMAX=XON(J)                                                       Z1  0330
      JMAX(I)=J                                                         Z1  0340
      GO TO 30                                                          Z1  0350
   25 XMIN=XON(J)                                                       Z1  0360
      JMIN(I)=J                                                         Z1  0370
   30 CONTINUE                                                          Z1  0380
      NB=NE+1                                                           Z1  0390
      WRITE (6,80)XMIN,XMAX,JMIN(I),JMAX(I)                             Z1  0410
   35 CONTINUE                                                          Z1  0420
   40 DO 65 J=NBP1,JJSS
      IA=1                                                              Z1  0440
      YAR(IA)=0.                                                        Z1  0444
      IF(XON(1).GT.XON(3))GOTO42
      CALL SINTP (XON,YON,NBDY1,XON(J),YAR(IA))                         Z1  0450
```

```
   42     IF(NBDYS.LE.2) GOTC50
          DO 45 I=2,NBDSM1
          JEND=NBDPTS(I)
          JMI=JMIN(I)                                                      Z1   0490
          JMA=JMAX(I)                                                      Z1   0510
          IF (XON(J).GT.XON(JMA).OR.XON(J).LT.XON(JMI)) GO TO 45           Z1   0520
          IA=IA+1                                                          Z1   0530
          CALL SINTP (XON(JMA),YON(JMA),JMI-JMA+1,XON(J),YAR(IA))          Z1   0540
          IA=IA+1                                                          Z1   0550
          CALL SINTP (XON(JMI),YON(JMI),JEND-JMI+1,XON(J),YAR(IA))         Z1   0560
       45 CONTINUE                                                         Z1   0570
       50 IA=IA+1                                                          Z1   0580
          YAR(IA)=YON(J)                                                   Z1   0590
          IS=0                                                             Z1   0600
          AREA=0.0                                                         Z1   0610
          DO 55 I=1,IA,2                                                   Z1   0620
          IS=IS+1                                                          Z1   0630
          AREAS(IS)=(YAR(I+1)**2-YAR(I)**2)*PI                             Z1   0640
          IF(PROG.EQ.D2TEST)AREAS(IS)=YAR(I+1)-YAR(I)
          AREA=AREA+AREAS(IS)                                              Z1   0650
       55 CONTINUE                                                         Z1   0660
          AREAD=AREA+YAR(1)**2*PI                                          Z1   0670
          IF(PROG.EQ.D2TEST)AREAD=AREA+YAR(1)
          AREA = AREA/AREF
          AREAD = AREAD/AREF
          XPLT = XON(J)/ELREF
          YPLT = YON(J)/ELREF
          YHPLT = YAR(1)/ELREF
          IF((YAR(1)/YON(J)).EQ.-1.) GO TO 56
          ENSUBK=CAPPA(J)*(YON(J)-YAR(1))/SQRT(1.+YAR(1)/YON(J))           Z1   0680
       56 IF(CAPPA(J).GT.90000.)ENSUBK=99999.                              Z1   0690
          IF(NBDYS.LE.2)GOTO60
          WRITE (6,85)(AREAS(I),I=1,IS)                                    Z1   0710
       60 WRITE (6,90)J,XPLT,YPLT,YHPLT,AREA,AREAD,ENSUBK                  Z1   0720
          WRITE (6,95)                                                     Z1   0730
          IF (XON(J).GT.(XX*EXEP+XMIN)) GO TO 65                           Z1   0740
          JPLA=JPLA+1                                                      Z1   0750
          IF(IPLOTA.LE.0) GO TO 65
          XXA(JPLA)=XON(J)                                                 Z1   0760
          ANULUS(JPLA)=AREA                                                Z1   0770
          DISC(JPLA)=AREAD                                                 Z1   0780
       65 CONTINUE                                                         Z1   0790
C(((((    IF AREA PLOT IS NOT REQUIRED, GO TO 70                           Z1   0800
          IF (IPLOTA.LE.0) GO TO 70                                        Z1   0810
          CALL CSCALE(DISC,YY,JPLA,1,10,EXPIN,DEEX)                        Z1   0820
          CALL CSCALE(ANULUS,YY,JPLA,1,10,EXMIN,DEEX)                      Z1   0830
          DISC(JPLA+1)=EXMIN                                               Z1   0840
          DISC(JPLA+ 2)=DEEX                                               Z1   0850
          CALL PLOT(XX,0.,-3)                                              Z1   0860
          CALL PLOXIS(XX,YY,EXEP,DISC(JPLA+2),XMIN,DISC(JPLA+1),.25,.25,0,0,Z1   0870
         14,3,0,0)                                                         Z1   0880
C(((((  PLOT THE DISC AREA VS. X                                           Z1   0890
          CALL LINE(XXA,DISC,JPLA,1,1,3,XMIN,EXEP,DISC(JPLA+1),DISC(JPLA+2))Z1  0910
C((((( PLOT  THE ANNULUS AREA VS. X                                        Z1   0910
          CALL LINE(XXA,ANULUS,JPLA,1,1,3,XMIN,EXEP,DISC(JPLA+1),DISC(JPLA+2Z1  0920
         1))                                                               Z1   0930
       70 RETURN                                                           Z1   0940
```

```
      C                                                                  Z1   0950
   75 FORMAT (1HO//9X,1H1,14X,3HXON,18X,3HYON,16X,4HYONH,12X,4HAREA,14X,Z1   0960
     19HDISC AREA,10X,6HENSUBK)                                          Z1   0970
   80 FORMAT (1HO,5X,7HXMIN = ,1PE14.5,5X,7HXMAX = ,1PE14.5,5X,7HJMIN = Z1   0980
     I,I6,5X,7HJMAX = ,I6)                                               Z1   0990
   85 FORMAT (74X,1PE19.4)                                               Z1   1000
   90 FORMAT (8X,I3,1P6E19.4)                                            Z1   1010
   95 FORMAT (1HO)                                                       Z1   1020
   98 FORMAT(1H1)
  100 FORMAT(1HO,//,6X,'XON,YON,YONH HAVE BEEN SCALED BY L = ',OPF10.4)
  110 FORMAT(1HO,5X,'AREA AND DISC AREA HAVE BEEN SCALED BY AREF = ',
     1        OPF10.4)
      END                                                                Z1   1030
```

```
      SUBROUTINE DRAW(KR,KK )                                            Z2   0000
C(((((( SUBROUTINE ADDED TO DRAW PICTURE OF INLET VIA CALCOMP PLOTTER.   Z2   0010
C(((((( CALLED ONCE FOR EACH SEGMENT                                     Z2   0020
C                                                                        Z2   0030
      DIMENSION X(200),Y(200)                                            Z2   0040
        COMMON /SS/ NBDY1,NBODY2,TYPBDY,NBDYS                            Z2   0050
      COMMON /FOR3SS/IO,DELS,XBK(20),YBK(20),XON(500),YON(500),DYDXO(500Z2   0060
     1),ALPHA(500),CAPPA(500),SON(500),PIO180                            Z2   0070
      COMMON/ LOT/XMIN,YMIN,ORD,EXEP,XX,YY,LPNCHO,IPLOTA,MM              Z2   0080
      COMMON/TOL/BAGS(15),BAGX(15),ZAP(15),NZAP(15)                      Z2   0090
      KL=KR+1                                                            Z2   0100
      II=0                                                               Z2   0110
      DO 20 I=1,KL                                                       Z2   0120
      N=KK+I-1                                                           Z2   0130
      IF (II.GE.200.OR.N.GT.500) GO TO 30                                Z2   0140
C((((((   TEST EACH (X,Y) PT. EXCLUDE THOSE BEYOND (XX*EXEP+XMIN) INCHESZ2   0150
      YMAN=YY*ORD+YMIN
      IF(LPNCHO.EQ.2)  YON(N)=YMAN-( YON(N)-YMAN)
      IF(XON(N).LT.XMIN) GO TO 20
      IF(YON(N).LT.YMIN) GO TO 20
      IF (XON(N)-XX*EXEP-XMIN) 10,10,20                                  Z2   0160
   10 IF (YON(N)-YY*ORD-YMIN) 15,15,20                                   Z2   0170
   15 II=II+1                                                            Z2   0180
      X(II)=XON(N)                                                       Z2   0190
      Y(II)=YON(N)                                                       Z2   0200
      IF (II.NE.1.OR.TYPBDY.EQ.1.) GO TO 20                              Z2   0210
C((((((  STORE CURVATURE VALUES OF SEGMENT'S FIRST PT. FOR USE WITH      Z2   0220
C SUBSEQUENT CURVATURE PLOTS.                                            Z2   0230
      MM=MM+1                                                            Z2   0240
      BAGX(MM)=XON(N)                                                    Z2   0250
      BAGS(MM)=SON(N)                                                    Z2   0260
      ZAP(MM)=CAPPA(N)                                                   Z2   0270
      NZAP(MM)=N                                                         Z2   0280
   20 CONTINUE                                                           Z2   0290
C((((((  DRAW A SEGMENT MARKER AT FIRST PT. OF SEGMENT                   Z2   0300
      IF (XON(KK).GT.(XX*EXEP+XMIN).OR.YON(KK).GT.(YY*ORD+YMIN)) GO TO 2Z2   0310
     *5                                                                  Z2   0320
      XSYM= (X(1)-XMIN)/EXEP                                             Z2   0330
      YSYM=(Y(1)-YMIN)/ORD                                               Z2   0340
      CALL SYMBOL(XSYM,YSYM,.2,1,0.,-1)                                  Z2   0350
   25 CALL LINE(X,Y,II,1,1,3,XMIN,EXEP,YMIN,ORD)                         Z2   0360
      RETURN                                                             Z2   0370
   30 WRITE (6,35)II,N                                                   Z2   0380
   35 FORMAT(1HO,' SCIRCLE ERROR EXIT - DATA POINTS EXCEED 200 ON A SEG Z2   0390
     1OR EXCEED 500 ON TOTAL INLET - ' /2I8)                             Z2   0400
      STOP                                                               Z2   0410
      END                                                                Z2   0420
```

```
      SUBROUTINE PLOXIS(XX,YY,EXEP,ORD,OFSETA,OFSET,SLETRS,SNOSZ,K5,K6,MZ3   0000
     1,L,NK,NL)                                                        Z3   0010
C                                                                      Z3   0020
C***********************************************************************Z3   0030
C******   SUBROUTINE ADDED TO DRAW AND LABEL AXIS FRAMES FOR ALL PLOTS *Z3   0040
      COMMON/TITL/ TTITL(9,6)                                          Z3   0050
      UP =11.-YY-2.*SNOSZ                                              Z3   0060
      M1=XX                                                            Z3   0070
      M2=YY                                                            Z3   0080
      CALL PLOT(4.,-11.,-3)                                            Z3   0090
      CALL PLOT(0.,UP,-3)                                              Z3   0100
      DO 25 I=1,M1                                                     Z3   0110
      X=I                                                              Z3   0120
      P=EXEP*X +OFSETA                                                 Z3   0130
      CALL PLOT(X,0.,2)                                                Z3   0140
      CALL PLOT(X, .2,2)                                               Z3   0150
      M=I/2                                                            Z3   0160
      B=FLOAT(I)-FLOAT(M)-X/2.                                         Z3   0170
      IF (B) 10,10,25                                                  Z3   0180
   10 IF (K5) 15,15,20                                                 Z3   0190
   15 CALL NUMBER(X-SNOSZ,-SNOSZ-.10,SNOSZ,P,0.,NK)                    Z3   0200
      GO TO 25                                                         Z3   0210
   20 SN = 1.333*SNOSZ                                                 Z3   0220
      CALL NUMBER(X-SNOSZ-SNOSZ,-SN-SNOSZ-.10,SN,10.,0.,-1)            Z3   0230
      CALL NUMBER(999.0,-SNOSZ -.10,SNOSZ,P,0.,NK)                     Z3   0240
   25 CALL PLOT(X,0.,3)                                                Z3   0250
      B = (XX-54.*SLETRS)/2.                                           Z3   0260
      CALL SYMBOL(B,-SNOSZ-SLETRS-.15-.6,SLETRS,TTITL(1,K),0.,54)      Z3   0270
      CALL PLOT(0.,0.,3)                                               Z3   0280
      DO 45 J=1,M2                                                     Z3   0290
      Y=J                                                              Z3   0300
      O=ORD*Y+OFSET                                                    Z3   0310
      CALL PLOT(0.,Y,2)                                                Z3   0320
      CALL PLOT(.2,Y,2)                                                Z3   0330
      N=J/2                                                            Z3   0340
      B=FLOAT(J)-FLOAT(N)-Y/2.                                         Z3   0350
      IF (B) 30,30,45                                                  Z3   0360
   30 IF (K6) 35,35,40                                                 Z3   0370
   35 CALL NUMBER(-4.*SNOSZ         -.15,Y,SNOSZ,O,0.,NL)              Z3   0380
      GO TO 45                                                         Z3   0390
   40 SN = 1.333*SNOSZ                                                 Z3   0400
      CALL NUMBER(-.15 -SN-SN-SN    ,Y-SNOSZ,SN,10.,0.,-1)             Z3   0410
      CALL NUMBER(999.0,Y+SN-SNOSZ,SNOSZ,O,0.,NL)                      Z3   0420
   45 CALL PLOT(0.,Y,3)                                                Z3   0430
      C = (YY-54.*SLETRS)/2.                                           Z3   0440
      CALL SYMBOL(-SNOSZ-SNOSZ-SNOSZ-.15-.6,C,SLETRS,TTITL(1,L),90.,54) Z3   0450
      CALL PLOT(0.,YY,3)                                               Z3   0460
      CALL PLOT(XX,YY,2)                                               Z3   0470
      CALL PLOT(XX,0.,2)                                               Z3   0480
      DO 50 J=1,M2,2                                                   Z3   0490
      Y=J                                                              Z3   0500
      IF (Y.EQ.YY) GO TO 55                                            Z3   0510
      CALL PLOT(XX,Y,3)                                                Z3   0520
      CALL PLOT(0.,Y,2)                                                Z3   0530
      IF ((Y+1.).EQ.YY) GO TO 55                                       Z3   0540
      CALL PLOT(0.,Y+1.,3)                                             Z3   0550
```

```
   50 CALL PLOT(XX,Y+1.,2)                                              Z3  0560
   55 CONTINUE                                                          Z3  0570
      DO 60 J=1,MI,2                                                    Z3  0580
      X=J                                                               Z3  0590
      IF (X.EQ.XX) GO TO 65                                             Z3  0600
      CALL PLOT(XX-X,YY,3)                                              Z3  0610
      CALL PLOT(XX-X,0.,2)                                              Z3  0620
      IF ((XX-X-1.).EQ.0.) GO TO 65                                     Z3  0630
      CALL PLOT(XX-X-1.,0.,3)                                           Z3  0640
   60 CALL PLOT(XX-X-1.,YY,2)                                           Z3  0650
   65 RETURN                                                            Z3  0660
      END                                                               Z3  0670
```

```
      SUBROUTINE SINTP (Z,W,N,X1,Y1)                                    Z4  0000
C((((((( ENLARGED FROM THE ORIGINAL (200)                               Z4  0010
      DIMENSION A(13)                                                   Z4  0020
      DIMENSION X(250), Y(250), Z(250), W(250)                          Z4  0030
      DATA EODFF/6HENDOFF/                                              Z4  0040
      DO 10 I=1,N                                                       Z4  0050
      X(I)=Z(I)                                                         Z4  0060
   10 Y(I)=W(I)                                                         Z4  0070
      CALL SORTXY (X,Y,N)                                               Z4  0080
C                                                                       Z4  0090
      DO 15 I=1,N                                                       Z4  0100
      K=I                                                               Z4  0110
      IF (X1.GT.X(I)) GO TO 15                                          Z4  0120
      IF (X1.EQ.X(I)) GO TO 20                                          Z4  0130
      IF (X1.LT.X(I)) GO TO 25                                          Z4  0140
   15 CONTINUE                                                          Z4  0150
   20 Y1=Y(K)                                                           Z4  0160
      GO TO 30                                                          Z4  0170
   25 IF (K.EQ.1) GO TO 35                                              Z4  0180
      IF (K.EQ.N) K=N-1                                                 Z4  0190
      W1=(X1-X(K))*(X1-X(K+1))/(X(K-1)-X(K))/(X(K-1)-X(K+1))            Z4  0200
      W2=(X1-X(K-1))*(X1-X(K+1))/(X(K)-X(K-1))/(X(K)-X(K+1))            Z4  0210
      W3=(X1-X(K-1))*(X1-X(K))/(X(K+1)-X(K-1))/(X(K+1)-X(K))            Z4  0220
      Y1=Y(K-1)*W1+Y(K)*W2+Y(K+1)*W3                                    Z4  0230
   30 RETURN                                                            Z4  0240
   35 Y1=0.0                                                            Z4  0250
      RETURN                                                            Z4  0260
      ENTRY ECHO                                                        Z4  0270
      CALL ERTRAN(6,'@ASG,T 25. . ')                                    Z4  0280
      WRITE (6,40)                                                      Z4  0290
   40 FORMAT(1H1,23X,' INPUT FILE DUMP'/)                               Z4  0300
   45 READ (5,50,END=60)A                                               Z4  0310
   50 FORMAT(13A6)                                                      Z4  0320
      WRITE (6,55)A                                                     Z4  0330
      WRITE (25,50)A                                                    Z4  0340
   55 FORMAT(1H ,13A6)                                                  Z4  0350
      GO TO 45                                                          Z4  0360
   60 WRITE (25,50)EODFF                                                Z4  0370
       REWIND 25                                                        Z4  0380
      RETURN                                                            Z4  0390
      END                                                               Z4  0400
```

```
      SUBROUTINE SORTXY(X,Y,NPTS)                                      ZS  0000
      DIMENSION X(100),Y(100)                                          ZS  0010
   10 N=NPTS                                                           ZS  0020
   15 NN=N-1                                                           ZS  0030
   20 DO 55 KT=1,NN                                                    ZS  0040
      XMIN=X(KT)                                                       ZS  0050
      JAD=KT                                                           ZS  0060
      JKL=KT+1                                                         ZS  0070
   25 DO 45 JK=JKL,N                                                   ZS  0080
   30 IF (XMIN-X(JK)) 45,45,35                                         ZS  0090
   35 XMIN=X(JK)                                                       ZS  0100
   40 JAD=JK                                                           ZS  0110
   45 CONTINUE                                                         ZS  0120
   50 YMIN=Y(JAD)                                                      ZS  0130
      X(JAD)=X(KT)                                                     ZS  0140
      Y(JAD)=Y(KT)                                                     ZS  0150
      X(KT)=XMIN                                                       ZS  0160
      Y(KT)=YMIN                                                       ZS  0170
   55 CONTINUE                                                         ZS  0180
      RETURN                                                           ZS  0190
      END                                                              ZS  0200
```

```
C  NEWMAN PROGRAM - TWO DIMENSIONAL MULTIELEMENT AIRFOILS               EXEC001
C                                                                       EXEC002
C  THIS IS THE EXECUTIVE ROUTINE FOR THE NEW NUEMANN                    EXEC003
C                                                                       EXEC004
C                                                                       EXEC005
      COMMON /BFLAG/ IDP(10), INL(10), IFL(10), NL(10), LIFT(10),       EXEC006
     1               IBMF(10), ISAV1(10), ISAV2(10), ISAV3(10),         EXEC007
     2               BTITLE(10, 7), IBT, IBST, IBTOT, NELTOT,           EXEC008
     3               ITRB(10), INMB(10), CHORDB(10), IBD(10), LIFTOT    EXEC009
     4               ,IPRB(10), IFST(10), ISEC(10), FTITLE(15), IPVR(10) EXEC010
C                                                                       EXEC011
C                                                                       EXEC012
C                                                                       EXEC013
      COMMON /COMBOD/CCL, INCLT, CLT, ALPHA, SUMDS(10), TLU(10,12),IND  EXEC014
     1             , ALPHAO, CNU(10), SMDSWF(10), MIO(10)               EXEC015
C                                                                       EXEC016
      COMMON /FILEID/ IFILE1, IFILE2, IFILE3, IFILE4, IFILE5,           EXEC017
     1                IFILE6, IFILE7, IFILE8, IFILE9, IFIL10,           EXEC018
     2                IFIL11, IFIL12, IFIL13, IFIL14, IFIL15            EXEC019
     3               ,IFIL16, IFIL17, IFIL18, IFIL19, IFIL20            EXEC020
      COMMON /MDATA/ ISOL,IOFF,NONU,NBNU,IPRINT,MORE,M                  EXEC021
      COMMON/ROTAT/NROT, ROTRAD(10)                                     EXEC022
C                                                                       EXEC023
C  SET UP THE VARIOUS STORAGE UNITS REQUIRED BY THE PROGRAM             EXEC024
      CALL FILES                                                        EXEC025
   10 CONTINUE                                                          EXEC026
      REWIND   IFILE1                                                   EXEC027
      REWIND   IFILE2                                                   EXEC028
      REWIND   IFILE3                                                   EXEC029
      REWIND   IFILE4                                                   EXEC030
      REWIND   IFILE8                                                   EXEC031
      REWIND   IFILE9                                                   EXEC032
      REWIND   IFIL10                                                   EXEC033
      REWIND   IFIL11                                                   EXEC034
      REWIND   IFIL12                                                   EXEC035
      REWIND   IFIL13                                                   EXEC036
      REWIND   IFIL14                                                   EXEC037
      REWIND   IFIL15                                                   EXEC038
      REWIND   IFIL16                                                   EXEC039
      REWIND IFIL17                                                     EXEC040
      CALL TSETV                                                        EXEC041
C                                                                       EXEC042
C                                                                       EXEC043
      CALL  MAIN1                                                       EXEC044
C                                                                       EXEC045
C                                                                       EXEC046
C  OBTAIN SIGMA SOLUTIONS                                               EXEC047
      ISIZE = 11413                                                     EXEC048
      CALL SOLVE (NELTOT, M, ISIZE, ISOL)                               EXEC049
C                                                                       EXEC050
C                                                                       EXEC051
      CALL TIMEV(T)                                                     EXEC052
      WRITE(6,70) T                                                     EXEC053
   70 FORMAT (1H0, 'SOLVE COMPLETE, READ FLOW TITLE & CONTROL CARD, ',  EXEC054
     1        'CALL COMBO, T = ', F9.3, 'SECONDS.')                     EXEC055
C                                                                       EXEC056
      CALL MAIN3                                                        EXEC057
C                                                                       EXEC058
C                                                                       EXEC059
      IF (MORE .EQ. 1)  GO TO 10                                        EXEC060
      STOP                                                              EXEC061
      END                                                               EXEC062
```

```
      FUNCTION  ABFORM (XI, YI, DSI, SINI, COSI, IFST,               ABF0001
     1                  XJ, YJ, DSJ, SINJ, COSJ, ISEC,               ABF0002
     2                  AI, AC, AP, BO, BI, BC, BP)                  ABF0003
C                                                                    ABF0004
C                                                                    ABF0005
C  THIS ROUTINE ACTUALLY CALCULATES THE INDUCED VELOCITY             ABF0006
C  ARRAY ELEMENTS A AND B.                                           ABF0007
C                                                                    ABF0008
C                                                                    ABF0009
      DATA E1,E2/169.0,11.111/,E0,EY/0.0001,0.01/                    ABF0010
C                                                                    ABF0011
C                                                                    ABF0012
      DX = XI - XJ                                                   ABF0013
      DY = YI - YJ                                                   ABF0014
      ROSQ = DX**2 + DY**2                                           ABF0015
      DSJSQ = DSJ**2                                                 ABF0016
C                                                                    ABF0017
      IF (ROSQ .LT. DSJSQ*E1)  GO TO 10                              ABF0018
C                                                                    ABF0019
C  USE FAR FIELD FORMULAS                                            ABF0020
      VX = 2.*DSJ/ROSQ                                               ABF0021
      VY = VX*DY                                                     ABF0022
      VX = VX*DX                                                     ABF0023
      AO = -VX*SINI + VY*COSI                                        ABF0024
      ABFORM = AO                                                    ABF0025
      BO =  VX*COSI + VY*SINI                                        ABF0026
      IF ((IFST .EQ. 0) .AND. (ISEC .EQ. 0))  RETURN                 ABF0027
      SC = SINI*COSJ                                                 ABF0028
      SS = SINI*SINJ                                                 ABF0029
      CS = COSI*SINJ                                                 ABF0030
      CC = COSI*COSJ                                                 ABF0031
      SSCC = (SS - CC)*VX*VY/12.0                                    ABF0032
      CSSC = (CS + SC)*VX*VY/12.0                                    ABF0033
      DXR = 1.0 - 2.0*DX**2/ROSQ                                     ABF0034
      DYR = 1.0 - 2.0*DY**2/ROSQ                                     ABF0035
      DDR = DSJSQ/(ROSQ*6.0)                                         ABF0036
C                                                                    ABF0037
      AI = DDR*(SC*DXR - CS*DYR) - SSCC                              ABF0038
      BI =-DDR*(CC*DXR + SS*DYR) + CSSC                              ABF0039
      AC =-DDR*(SS*DXR + CC*DYR) - CSSC                              ABF0040
      BC = DDR*(CS*DXR - SC*DYR) - SSCC                              ABF0041
      AP = AO/12.0                                                   ABF0042
      BP = BO/12.0                                                   ABF0043
      RETURN                                                         ABF0044
C                                                                    ABF0045
C  USE NEAR FIELD FORMULAS                                           ABF0046
   10 X = DX*COSJ + DY*SINJ                                          ABF0047
      Y = DY*COSJ - DX*SINJ                                          ABF0048
      S = SINI*COSJ - COSI*SINJ                                      ABF0049
      C = COSI*COSJ + SINI*SINJ                                      ABF0050
C                                                                    ABF0051
      IF (ROSQ .GT. DSJSQ*E2)  GO TO 20                              ABF0052
C                                                                    ABF0053
C  USE EXACT FORMULAS                                                ABF0054
      XB = X/DSJ                                                     ABF0055
      YB = Y/DSJ                                                     ABF0056
```

```
      RO = XB**2 + YB**2                                                ABF0057
      RD = XB**2 - YB**2                                                ABF0058
      R1 = RO + XB + 0.25                                               ABF0059
      R2 = RO - XB + 0.25                                               ABF0060
C                                                                       ABF0061
      VX = +ALOG((ROSQ+DSJ*X+0.25*DSJSQ)/(ROSQ-DSJ*X+0.25*DSJSQ))       ABF0062
      Y = Y*DSJ                                                         ABF0063
      X = ROSQ - 0.25*DSJSQ                                             ABF0064
      VY = 2.0*ATAN2(Y,X)                                               ABF0065
      IF ((IFST .EQ. 0) .AND. (ISEC .EQ. 0))  GO TO 30                  ABF0066
C                                                                       ABF0067
      VXO = VX                                                          ABF0068
      VYO = VY                                                          ABF0069
      VX1 = YB*VYO + XB*VXO - 2.0                                       ABF0070
      VY1 = XB*VYO - YB*VXO                                             ABF0071
      VXC = -2.0*VY1 + XB*YB/(R1*R2)                                    ABF0072
      VYC = 2.0*(VX1 + 1.0)-2.*(RO**2 - 0.25*RD)/(R1*R2)                ABF0073
      VXP = RD*VXO +  2.0*XB*(YB*VYO - 1.0)                             ABF0074
      VYP = RD*VYO -  2.0*YB*(XB*VXO - 1.0)                             ABF0075
C                                                                       ABF0076
      GO TO 30                                                          ABF0077
C                                                                       ABF0078
C                                                                       ABF0079
C  USE MULTIPOLE FORMULAS                                               ABF0080
   20 AE = X*DSJ/ROSQ                                                   ABF0081
      BE = Y*DSJ/ROSQ                                                   ABF0082
      ASQ = X**2/ROSQ                                                   ABF0083
      ESQ =DSJSQ/ROSQ                                                   ABF0084
      VX  = 2.0*AE*(1.0 + (ASQ - 0.75)*ESQ/3.0)                         ABF0085
      VY  = 2.0*BE*(1.0 + (ASQ - 0.25)*ESQ/3.0)                         ABF0086
      IF ((IFST .EQ. 0) .AND. (ISEC .EQ. 0))  GO TO 30                  ABF0087
      A4 = 0.15*(8.0*ASQ*(ASQ - 1.0) + 1.0)*ESQ                         ABF0088
      VX1 = ESQ*(2.0*ASQ - 1.0 + A4)/6.0                                ABF0089
      VY1 = AE*BE*(1.0 + 0.3*(2.0*ASQ - 1.0)*ESQ)/3.0                   ABF0090
      VXC = AE*BE*(1.0 + 0.3*(6.0*ASQ - 1.0)*ESQ)/3.0                   ABF0091
      BSQ = Y**2/ROSQ                                                   ABF0092
      VYC = ESQ*(2.0*BSQ - 1.0 -0.5*A4)/6.0                             ABF0093
      VYP = (1.0 + (ASQ - 0.25)*ESQ*0.6)/6.0                            ABF0094
      VXP = AE*VYP                                                      ABF0095
      VYP = BE*VYP                                                      ABF0096
C                                                                       ABF0097
C                                                                       ABF0098
   30 CONTINUE                                                          ABF0099
      ABFORM = -S*VX + C*VY                                             ABF0100
      BO = C*VX + S*VY                                                  ABF0101
      IF ((IFST .EQ. 0) .AND. (ISEC .EQ. 0))  RETURN                    ABF0102
      A1 = -S*VX1 + C*VY1                                               ABF0103
      AC = -S*VXC + C*VYC                                               ABF0104
      AP = -S*VXP + C*VYP                                               ABF0105
C                                                                       ABF0106
      B1= C*VX1 + S*VY1                                                 ABF0107
      BC = C*VXC + S*VYC                                                ABF0108
      BP = C*VXP + S*VYP                                                ABF0109
C                                                                       ABF0110
C                                                                       ABF0111
      RETURN                                                            ABF0112
      END                                                               ABF0113
```

```
      FUNCTION ARSIN(X)
      ARSIN=ASIN(X)
      RETURN
      END
```

```
      SUBROUTINE ASSEMB                                                 ASEM001
C                                                                       ASEM002
C  THIS ROUTINE READS IN NORMAL AND TANGENTIAL ONSET FLOWS              ASEM003
C  IN ROW ORDER, AND ASSEMBLES THEM IN COLUMN ORDER AND WRITES          ASEM004
C  THEM BACK OUT                                                        ASEM005
C                                                                       ASEM006
C  NORMAL ONSET FLOWS ON UNIT IF11                                      ASEM007
C                                                                       ASEM008
C  TANGENTIAL ONSET FLOWS ON UNIT IF12                                  ASEM009
C                                                                       ASEM010
C  THE NORMAL ONSET FLOWS ARE ALSO WRITTEN ON IFO4 (RHS TAPE)           ASEM011
C  FOR USE IN MATRIX SOLUTION.                                          ASEM012
C                                                                       ASEM013
C                                                                       ASEM014
C  ALSO, NON-UNIFORM ONSET FLOWS ARE READ IN AND WRITTEN ON UNITS.      ASEM015
C                                                                       ASEM016
C                                                                       ASEM017
      DIMENSION  VN(500,12), VNUF(500), VTUF(500), II(5), IE(5)         ASEM018
      DIMENSION XO(500),YO(500),DS(500),SA(500),CA(500)                 ASEM019
C                                                                       ASEM020
      EQUIVALENCE (VN(1,1), VNUF(1)), (VN(1,2), VTUF(1))                ASEM021
C                                                                       ASEM022
C                                                                       ASEM023
C                                                                       ASEM024
      COMMON /COMBOD/CCL, INCLT, CLT, ALPHA, SUMDS(10), ILU(10,12),IND  ASEM025
     1            , ALPHAO, CNU(10), SNDSUF(10), NIO(10)                ASEM026
C                                                                       ASEM027
      COMMON /BFLAG/ IDB(10), INL(10), IFL(10), NL(10), LIFT(10),       ASEM028
     1                 IBMF(10), ISAV1(10), ISAV2(10), ISAV3(10),       ASEM029
     2                BTITLE(10, 7), IBT, IBST, IBTOT, NELTOT,          ASEM030
     3                 ITRB(10), INNB(10), CHORDB(10), IBD(10), LIFTOT  ASEM031
     4                ,IPRB(10), IFST(10), ISEC(10), FTITLE(15), IPVR(10) ASEM032
      COMMON /FILEIO/  IFO1, IFO2, IFO3, IFO4, IFO5,                    ASEM033
     1                          IFO6, IFO7, IFO8, IFO9, IF10,           ASEM034
     2                     IF11, IF12, IF13, IF14, IF15                 ASEM035
     3                   ,IF16, IF17, IF18, IF19, IF20                  ASEM036
      COMMON/ROTAT/NROT, ROTRAD(10)                                     ASEM037
      COMMON /MDATA/ ISOL,IOFF,MONU,MBNU,IPRINT,MORE,M                  ASEM038
      COMMON/ELDATA/ XO, YO, DS, SA, CA, CURV(500), DL(500)             ASEM039
C                                                                       ASEM040
C  NORMAL ONSET FLOWS                                                   ASEM041
      REWIND IF11                                                       ASEM042
      DO 10 I = 1,NELTOT                                                ASEM043
   10 READ(IF11)  (VN(I,K), K=1,M )                                     ASEM044
      REWIND IF11                                                       ASEM045
      REWIND IFO4                                                       ASEM046
      MTOT = M                                                          ASEM047
      IF (MONU .GT. 0)  MTOT = MTOT + MONU                              ASEM048
      WRITE(IFO4)  MTOT                                                 ASEM049
      DO 20 K = 1,M                                                     ASEM050
      CALL SAVE(IF11, 1, 1, NELTOT, VN(1,K), 1, VNA)                    ASEM051
   20 CALL SAVE(IFO4, 1, 1, NELTOT, VN(1,K), 1, VNA)                    ASEM052
C                                                                       ASEM053
C  TANGENTIAL ONSET FLOWS                                               ASEM054
      REWIND IF12                                                       ASEM055
      DO 50 I = 1,NELTOT                                                ASEM056
```

```
   30 READ(IF12)  (VN(I,K), K= 1,M )                                 ASEM057
      REWIND IF12                                                    ASEM058
      DO 40 K = 1,M                                                  ASEM059
   40 CALL SAVE(IF12, 1, 1, NELTOT, VN(1,K), 1, VNA)                 ASEM060
C                                                                    ASEM061
C                                                                    ASEM062
C  CHECK IF NON-UNIFORM CNSET FLOWS INPUT                            ASEM063
  340 IF (NONU .LE. 0)  RETURN                                       ASEM064
C                                                                    ASEM065
      DO 560 L = 1,NONU                                              ASEM066
      M = M + 1                                                      ASEM067
C                                                                    ASEM068
C  PRESET ALL VELOCITIES TO ZERO                                     ASEM069
      DO 350 I = 1,NELTOT                                            ASEM070
      VNUF(I) = 0.0                                                  ASEM071
  350 VTUF(I) = 0.0                                                  ASEM072
      CALL SAVE(IF11, 1, 1, NELTOT, VNLF, 1, VN)                     ASEM073
C                                                                    ASEM074
C                                                                    ASEM075
C  PRESET TLU ARRAYS TO ZERO                                         ASEM076
      DO 355 I = 1,IBTOT                                             ASEM077
  355 TLU(I,M) = 0.0                                                 ASEM078
C                                                                    ASEM079
C                                                                    ASEM080
C  READ IN COMBINATION CONSTANTS FOR NON-UNIFORM FLOWS               ASEM081
      ITYP = 10                                                      ASEM082
  360 READ(5,440) (CNU(I), I = 1, 6), ITYPE                          ASEM083
      IF (ITYPE .NE. ITYP)  CALL TYPE(ITYP, ITYPE)                   ASEM084
C                                                                    ASEM085
      DO 460 NB = 1,NBNU                                             ASEM086
      LB = 0                                                         ASEM087
C  READ BODY CONTROL CARD FOR NON-UNIFORM FLOW                       ASEM088
      ITYP = 11                                                      ASEM089
      READ(5,370) IBOD, IN, IT,MN, (I1(I),IE(I), I = 1,5),CB,   ITYPE ASEM090
  370 FORMAT (I1,1X3I1,  10(2X13), F10.5, 5X12)                      ASEM091
      IF (ITYPE .NE. ITYP)  CALL TYPE(ITYP, ITYPE)                   ASEM092
C                                                                    ASEM093
C  SEARCH FOR BODY ID ANC SET LIFTING BODY COUNTER                   ASEM094
      DO 380 IB = 1,IBTOT                                            ASEM095
      IF (LIFT(IB) .NE. 0)  LB = LB + 1                              ASEM096
      IIB = IB                                                       ASEM097
      IF (IBOD .EQ. IDB(IB))  GO TO 400                              ASEM098
  380 CONTINUE                                                       ASEM099
      WRITE(6,390) IBOD                                              ASEM100
  390 FORMAT(1HO, 'NON-UNIFORM FLOW INPUT, BODY WITH ID = ', I1,     ASEM101
     1            ' DOES NOT EXIST.  RUN TERMINATED. ')              ASEM102
      STOP                                                           ASEM103
C                                                                    ASEM104
  400 MO = MIO(IIB) - 1                                              ASEM105
      IF (NROT .GT. 0)  GO TO 455                                    ASEM106
C                                                                    ASEM107
      DO 450 LL= 1,NN                                                ASEM108
      IO = MO + I1(LL)                                               ASEM109
      IF = MO + IE(LL)                                               ASEM110
C                                                                    ASEM111
      IF (IN .EQ. 0)  GO TO 420                                      ASEM112
C  READ IN NORMAL VELOCITIES                                         ASEM113
```

```
      ITYP = 12                                                         ASEM114
      M2 = IO - 1                                                       ASEM115
  410 M1 = M2 + 1                                                       ASEM116
      M2 = M1 + 5                                                       ASEM117
      READ(5,440) (VNUF(I), I = M1,M2), ITYPE                           ASEM118
      IF (ITYPE .NE. ITYP)  CALL TYPE(ITYP, ITYPE)                      ASEM119
      IF (M2 .LT. IF)  GO TO 410                                        ASEM120
  420 IF (IT .EQ. 0)  GO TO 450                                         ASEM121
C                                                                       ASEM122
C  READ IN TANGENTIAL VELOCITIES                                        ASEM123
      ITYP = 13                                                         ASEM124
      M2 = IO - 1                                                       ASEM125
  430 M1 = M2 + 1                                                       ASEM126
      M2 = M1 + 5                                                       ASEM127
      READ(5,440) (VTUF(I), I = M1,M2), ITYPE                           ASEM128
  440 FORMAT( 6F10.0, 10XI2)                                            ASEM129
      IF (ITYPE .NE. ITYP)  CALL TYPE(ITYP, ITYPE)                      ASEM130
      IF (M2 .LT. IF)  GO TO 430                                        ASEM131
  450 CONTINUE                                                          ASEM132
      GO TO 457                                                         ASEM133
C                                                                       ASEM134
C  SPECIAL ROTATING FLOW - INPUT GENERATED                              ASEM135
  455 CONTINUE                                                          ASEM136
C  COUNTERS FOR MATRIX STORAGE                                          ASEM137
      IO = MO + 1                                                       ASEM138
      IF = MO + NL(IIB)                                                 ASEM139
C                                                                       ASEM140
C  COUNTER FOR BODY GEOMETRY                                            ASEM141
      JI = INL(IIB) - 1                                                 ASEM142
C  ROTATION RADIUS                                                      ASEM143
      ROTRAD(IIB) = CB                                                  ASEM144
      CB = 1.0/CB                                                       ASEM145
      DO 456 I = IO,IF                                                  ASEM146
      JI = JI + 1                                                       ASEM147
      VNUF(I) = -YO(JI)*SA(JI) - XO(JI)*CA(JI)                          ASEM148
      VTUF(I) =  YO(JI)*CA(JI) - XO(JI)*SA(JI)                          ASEM149
  456 CONTINUE                                                          ASEM150
      WRITE(6,900) NONU,NROT,NBNU,L,NB,IO,IF                            ASEM151
  900 FORMAT(1H0, 7HNONU = ,I3,5X7HNROT = ,I3,5X7HNBNU = ,I3,           ASEM152
     1  4HL = ,I3,5X5HNB = ,I3, 5X5HIO = ,I3,5X5HIO = ,I3)              ASEM153
  457 CONTINUE                                                          ASEM154
C                                                                       ASEM155
C  ALL NON-U VELOCITIES FOR BODY ID=IBOD READ IN.                       ASEM156
C                                                                       ASEM157
C  SCALE VELOCITIES                                                     ASEM158
      IF (ABS(CB) .LT. 1.E-6)  GO TO 480                                ASEM159
      DO 470 I = IO,IF                                                  ASEM160
      VNUF(I) = VNUF(I)*CB                                              ASEM161
  470 VTUF(I) = VTUF(I)*CB                                              ASEM162
  480 CONTINUE                                                          ASEM163
C                                                                       ASEM164
C  RESET TLU IF LIFTING BODY                                            ASEM165
      IF (LIFT(IIB) .EQ. 0)  GO TO 460                                  ASEM166
      IO = MO + 1                                                       ASEM167
      IF = MO + NL(IIB)                                                 ASEM168
      TLU(LB,M) = VTUF(IO) + VTUF(IF)                                   ASEM169
  460 CONTINUE                                                          ASEM170
```

```
      C                                                                 ASEM171
      C  ALL VELOCITIES FOR A GIVEN NON-U FLOW READ IN.                  ASEM172
            CALL SAVE(IF04, 1, 1, NELTOT, VNUF, 1, VN)                  ASEM173
            CALL SAVE(IF12, 1, 1, NELTOT, VTUF, 1, VN)                  ASEM174
      C                                                                 ASEM175
      C                                                                 ASEM176
        490 J2 = 0                                                      ASEM177
        495 WRITE(6,500)                                                ASEM178
        500 FORMAT(1H1)                                                 ASEM179
            WRITE(6,510) L, M                                           ASEM180
        510 FORMAT(1H0, 15X, 'NON-UNIFORM FLOW NUMBER ', I2,', M = ', I3//   ASEM181
           1       T15,'I', T27,' VN', T44,' VT',                       ASEM182
           2       T75,'I', T87,' VN',T104,' VT')                       ASEM183
            J1 = J2 + 1                                                 ASEM184
            JMX = J1 + 49                                               ASEM185
            N2 = (NELTOT + 1 + J2)/2                                    ASEM186
            IF (JMX .GT. N2) JMX = N2                                   ASEM187
            J2 = JMX                                                    ASEM188
            DO 530 J = J1,JMX                                           ASEM189
            J2 = J2 + 1                                                 ASEM190
            IF (J2 .GT. NELTOT)  GO TO 540                              ASEM191
            WRITE(6,520) J,VNUF(J),VTUF(J), J2, VNUF(J2),VTUF(J2)       ASEM192
        520 FORMAT(1H , 11XI3, 2(5XF12.6), 23XI3, 2(5XF12.6))           ASEM193
        530 CONTINUE                                                    ASEM194
            IF (J2 .LT. NELTOT)  GO TO 495                              ASEM195
            GO TO 550                                                   ASEM196
        540 WRITE(6,520) J, VNUF(J), VTUF(J)                            ASEM197
        550 CONTINUE                                                    ASEM198
      C                                                                 ASEM199
      C                                                                 ASEM200
        560 CONTINUE                                                    ASEM201
      C                                                                 ASEM202
      C                                                                 ASEM203
            RETURN                                                      ASEM204
            END                                                         ASEM205
```

```
      SUBROUTINE COMBO(N, LT, MT, NONU)                                 COMB001
C                                                                       COMB002
C                                                                       COMB003
C CALCULATE COMBINATION CONSTANTS CCK, SYSTEM ANGLE OF ATTACK           COMB004
C AND CL, SYSTEM LIFT CURVE CONSTANTS (K1, K2, & K3), AND CK CONSTANTS. COMB005
C                                                                       COMB006
C     N = TOTAL NUMBER OF ELEMENTS                                      COMB007
C    LT = NUMBER OF LIFTING BODIES                                      COMB008
C    MT = NUMBER OF ONSET FLOWS                                         COMB009
C   MT1 = MT + 1                                                        COMB010
C   CCL = CHORD FOR CLT CALCULATION                                     COMB011
C   INCLT FLAG,  = 0, ALPHA INPUT (COMES IN AS CLT)                     COMB012
C              NOT = 0, CLT INPUT                                       COMB013
C                                                                       COMB014
C                                                                       COMB015
      DIMENSION  BLU(500), DV(10,12), A(10,10), CCK(10,12), SIG(500)    COMB016
C                                                                       COMB017
C                                                                       COMB018
      COMMON /FILEIO/ IF01, IF02, IF03, IF04, IF05, IF06, IF07, IF08,   COMB019
     1                IF09, IF10, IF11, IF12, IF13, IF14, IF15          COMB020
     2               ,IFIL16, IFIL17, IFIL18, IFIL19, IFIL20            COMB021
C                                                                       COMB022
C                                                                       COMB023
      COMMON /COMBOD/CCL, INCLT, CLT, ALPHA, SUMDS(10), TLU(10,12),IND  COMB024
     1           , ALPHAO, CNU(10), SMDSWF(10), NIO(10)                 COMB025
      COMMON /SIGMAS/CSIG(500), CK(12)                                  COMB026
      EQUIVALENCE (BLU(1), CSIG(1))                                     COMB027
C                                                                       COMB028
      DATA PI, RC/3.1415927, 1.7453293E-2/                              COMB029
      NRHS = 2                                                          COMB030
      IF (NONU .GT. 0)  NRHS = 3                                        COMB031
C                                                                       COMB032
      PIC = 0.0                                                         COMB033
      RK1 = 0.0                                                         COMB034
      RK2 = 0.0                                                         COMB035
      RK3 = 0.0                                                         COMB036
      ALPHAO = 0.0                                                      COMB037
      IF (LT .EQ. 0)  GO TO 140                                         COMB038
C                                                                       COMB039
C                                                                       COMB040
C CALCULATE TRAILING EDGE VELOCITY DIFFERENCE ARRAY                     COMB041
C                                                                       COMB042
C INITIALIZE DV ARRAY TO TLU VALUES                                     COMB043
      DO 5 L = 1,LT                                                     COMB044
      DO 5 K = 1,MT                                                     COMB045
    5 DV(L,K) = TLU(L,K)                                                COMB046
C                                                                       COMB047
      REWIND IF13                                                       COMB048
      DO 30 L = 1,LT                                                    COMB049
      CALL GETT (IF13, 1, N, BLU, 1, 0)                                 COMB050
      REWIND IF14                                                       COMB051
      DO 20 K = 1,MT                                                    COMB052
      CALL GETT (IF14, 1, N, SIG, 1, 0)                                 COMB053
      DO 10 J = 1,N                                                     COMB054
   10 DV(L,K) = DV(L,K) + BLU(J)*SIG(J)                                 COMB055
   20 CONTINUE                                                          COMB056
```

```
   30 CONTINUE                                                  COMB057
C                                                               COMB058
C                                                               COMB059
C  SET ARRAYS TO OBTAIN COMBINATION CONSTANTS CCK. USE MIS1     COMB060
C     A(LT,LT) IS COEF. ARRAY                                   COMB061
C     CCK(LT,3) IS INPUT AS RHS, RETURNED AS CCK.               COMB062
C                                                               COMB063
      DO 70 L = 1,LT                                            COMB064
      CCK(L,1) = -DV(L,1)                                       COMB065
      CCK(L,2) = -DV(L,2)                                       COMB066
      CCK(L,3) = 0.0                                            COMB067
      IF (NONU .LE.0)  GO TO 50                                 COMB068
      M = LT + 2                                                COMB069
      DO 40 K = 1,NONU                                          COMB070
      M = M + 1                                                 COMB071
   40 CCK(L,3) = CCK(L,3) - DV(L,M)*CNU(K)                      COMB072
   50 CONTINUE                                                  COMB073
C                                                               COMB074
      DO 60 K = 1,LT                                            COMB075
      K2 = K + 2                                                COMB076
   60 A(L,K) = DV(L,K2)                                         COMB077
C                                                               COMB078
   70 CONTINUE                                                  COMB079
C                                                               COMB080
      IF (LT .GT. 1)  GO TO 90                                  COMB081
C                                                               COMB082
C ONLY ONE LIFTING BODY.  CALCULATE COMBINATION                 COMB083
C CONSTANTS STRAIGHT AWAY.                                      COMB084
      DO 80 K = 1,MRHS                                          COMB085
   80 CCK(1,K) = CCK(1,K)/A(1,1)                                COMB086
      GO TO 110                                                 COMB087
C                                                               COMB088
C  CALL MIS1 FOR SOLUTION                                       COMB089
   90 CONTINUE                                                  COMB090
      D = 1.0                                                   COMB091
      L10 = 10                                                  COMB092
      CALL MIS1 (A, LT,L10, CCK, MRHS, NERR, D)                 COMB093
C                                                               COMB094
C  CHECK FOR SINGULAR CASE                                      COMB095
      WRITE (6,100)  NERR                                       COMB096
  100 FORMAT(1H0, 'ON RETURN FROM MIS1, NERR = ', I2)           COMB097
C                                                               COMB098
C  CALCULATE SYSTEM ANGLE OF ATTACK (ALPHA) AND TOTAL LIFT (CLT).  COMB099
C                                                               COMB100
  110 CONTINUE                                                  COMB101
      DO 130 L = 1,LT                                           COMB102
      RK1 = RK1 + SMDSWF(L) * CCK(L,1)                          COMB103
      RK2 = RK2 + SMDSWF(L) * CCK(L,2)                          COMB104
  120 RK3 = RK3 + SMDSWF(L) * CCK(L,3)                          COMB105
C                                                               COMB106
  130 CONTINUE                                                  COMB107
C                                                               COMB108
      PIC = 8.*PI/CCL                                           COMB109
      ALPHAO  = ATAN2(RK1, RK2)                                 COMB110
C                                                               COMB111
C  CHECK IF ALPHA OR CLT INPUT                                  COMB112
      IF (INCLT .EQ. 0)  GO TO 140                              COMB113
```

```
C                                                                       COMB114
C  CLT INPUT.  DETERMINE ALPHA                                          COMB115
      RK4 = (CLT/PIC - RK3)/SQRT(RK1**2 + RK2**2)                       COMB116
      ALPHA = -ALPHAO + ARSIN(RK4)                                      COMB117
      GO TO 150                                                         COMB118
C                                                                       COMB119
C  ALPHA INPUT, CALCULATE CLT (REPEATED IF INPUT)                       COMB120
  140 ALPHA = ALPHA*RC                                                  COMB121
  150 COSA = COS(ALPHA)                                                 COMB122
      SINA = SIN(ALPHA)                                                 COMB123
      IF (INCLT .EQ. 0)                                                 COMB124
     1CLT = PIC*(RK1*COSA + RK2*SINA + RK3)                             COMB125
      ALPHA = ALPHA/RC                                                  COMB126
      ALPHAO = ALPHAO/RC                                                COMB127
C                                                                       COMB128
C  CALCULATE COEFFICIENTS  CK(K) FOR LIFTING BODIES                     COMB129
      CK(1) = COSA                                                      COMB130
      CK(2) = SINA                                                      COMB131
C                                                                       COMB132
      IF (LT .LE. 0)  GO TO 165                                         COMB133
      DO 160 L = 1,LT                                                   COMB134
      K = L + 2                                                         COMB135
      CK(K) = CCK(L,1)*COSA + CCK(L,2)*SINA + CCK(L,3)                  COMB136
  160 CONTINUE                                                          COMB137
C                                                                       COMB138
C  SET NON-U FLOW CK'S TO ASSUMED VALUE OF 1.0                          COMB139
  165 CONTINUE                                                          COMB140
      IF (NONU .LE. 0)  GO TO 180                                       COMB141
      K = LT + 2                                                        COMB142
      DO 170 J = 1,NONU                                                 COMB143
      K = K + 1                                                         COMB144
  170 CK(K) = CNU(J)                                                    COMB145
  180 CONTINUE                                                          COMB146
C                                                                       COMB147
C  CALCULATE  COMBINED SIGMAS                                           COMB148
      DO 190 J = 1,N                                                    COMB149
  190 CSIG(J) = 0.0                                                     COMB150
      REWIND IF14                                                       COMB151
      DO 210 K = 1,NT                                                   COMB152
      CALL GETT (IF14, 1, N, SIG, 1, 0)                                 COMB153
      DO 200 J = 1,N                                                    COMB154
  200 CSIG(J) = CSIG(J) + SIG(J)*CK(K)                                  COMB155
  210 CONTINUE                                                          COMB156
C                                                                       COMB157
C                                                                       COMB158
C  PRINT OUT SOME STUFF FOR CHECKOUT PURPOSES                           COMB159
      WRITE (6,220)  ALPHA                                              COMB160
  220 FORMAT (1H1, 'COMBINATION CONSTANTS'//T10,'ALPHA  = ',            COMB161
     1        T30,'0', T50,'90', T64, F12.6)                            COMB162
C                                                                       COMB163
      IF (LT .LE. 0)  GO TO 245                                         COMB164
      DO 230 L = 1, LT                                                  COMB165
      K = L + 2                                                         COMB166
  230 WRITE (6,240)  L, CCK(L,1), CCK(L,2), CK(K)                       COMB167
  240 FORMAT (1H0, T14,I2, T24,F12.6, T44,F12.6, T64,F12.6)             COMB168
C                                                                       COMB169
  245 CONTINUE                                                          COMB170
      WRITE (6,250) RK1, RK2, RK3, ALPHAO, ALPHA, CLT                   COMB171
  250 FORMAT (1H0, ///' LIFT CURVE CONSTANTS'//T10,'RK1 = ',F12.6,      COMB172
     1        T30,'RK2 = ',F12.6, T50,'RK3 = ',F12.6//T10,              COMB173
     2       'ALPHAO = ', F12.6//T10,'ALPHA  = ',F12.6//T10,            COMB174
     3       'CLT    = ', F12.6)                                        COMB175
C                                                                       COMB176
C                                                                       COMB177
C                                                                       COMB178
      RETURN                                                            COMB179
      END                                                               COMB180
```

```
      SUBROUTINE    ELFORM (SUMDS)                                      ELF0001
C                                                                       ELF0002
      DIMENSION X(500),Y(500),X0(500),Y0(500),DS(500),SA(500),CA(500),  ELF0003
     1          IDR(10), INL(10), IFL(10), NL(10), LIFT(10), IBMF(10),  ELF0004
     2          ISAV1(10), ISAV2(10), ISAV3(10), ITITLE( 7),            ELF0005
     3          PTITLE(10, 7), SUMDS(10)                                ELF0006
     5         ,XP(500), YP(500)                                        ELF0007
C                                                                       ELF0008
C                                                                       ELF0009
      COMMON /ELDATA/ X0, Y0, DS, SA, CA, CURV(500), DL(500)            ELF0010
      COMMON/GCOEFS/ CD(500), CF(500), CG(500), CI(500), NF(500)        ELF0011
C                                                                       ELF0012
      COMMON /FILEID/ IF1, IF2, IF3, IF4, IF5, IF6, IF7, IF8, IF9,      ELF0013
     1  IF10, IF11, IF12, IF13, IF14, IF15, IF16, IF17, IF18, IF19, IF20ELF0014
C                                                                       ELF0015
      COMMON /BFLAG/ IDB, INL, IFL, NL, LIFT, IBMF, ISAV1, ISAV2,       ELF0016
     1                   ISAV3, BTITLE, IBI, IBST                       ELF0017
     2                  ,IBOTOT, NELTOT, ITRB(10), INMB(10), CHORDB(10), ELF0018
     3                   IBD(10), LIFTOT                                ELF0019
     4                  ,IPRB(10), IFST(10), ISEC(10), FTITLE(15), IPVR(10) ELF0020
C                                                                       ELF0021
      COMMON /GEOMD/ X, Y                                               ELF0022
C                                                                       ELF0023
      EQUIVALENCE  (XP(1), CD(1)), (YP(1), CF(1))                       ELF0024
C                                                                       ELF0025
      DATA        IBMAX,MAXEL/    10, 500/, DR/1.74532925E-2/           ELF0026
     1    , EPS/1.0E-7/                                                 ELF0027
C                                                                       ELF0028
C                                                                       ELF0029
C                                                                       ELF0030
C                                                                       ELF0031
C                                                                       ELF0032
      ISAVU = IF8                                                       ELF0033
C                                                                       ELF0034
C                                                                       ELF0035
C  READ IN BODY TITLE ANC CONTROL CARD                                  ELF0036
   10 ITYP = 1                                                          ELF0037
      READ (5,20) ID, ISV, ILIFT, TTITLE                                ELF0038
     1            ,IPARA, IFIRST, ISECND                                ELF0039
     2            ,ITR, INORM, IBOD, IDOLD, IPVOR, LAST                 ELF0040
     3            ,ITYPE                                                ELF0041
C                                                                       ELF0042
   20 FORMAT (3(I1,2X), 1X7A4, 5X9(I1,2X), 1XI1)                        ELF0043
      IF (ITYPE .NE. ITYP)  CALL TYPE(ITYP, ITYPE)                      ELF0044
C                                                                       ELF0045
C  READ IN COORDINATE TRANSFORMATION CARD IF REQUIRED                   ELF0046
      ITYP = 2                                                          ELF0047
      CHORD = 0.0                                                       ELF0048
      IF (ITR .EQ. 0 .OR. ITR .EQ. 2)  GO TO 40                         ELF0049
      READ (5,30) CHORD, XMULT, YMULT, DX, DY, THETA, XTO, YTO, ITYPE   ELF0050
   30 FORMAT (7(F8.0,1X), F8.0, I1)                                     ELF0051
      IF (ITYPE .NE. ITYP)  CALL TYPE(ITYP, ITYPE)                      ELF0052
C                                                                       ELF0053
C  DETERMINE STORAGE SEQUENCE                                           ELF0054
      IF(IBOD.LT.1.OR.IBOD.GT.6) GO TO 49
   40 GO TO (60,70,100,140,140,210), IBOD                               ELF0055
```

```
C                                                                        ELF0056
   49 WRITE (6,50) IBOD                                                  ELF0057
   50 FORMAT (1H1,'THE OPTION IBOD = ',I3,' IS NOT LEGITIMATE.')         ELF0058
      GO TO 640                                                          ELF0059
C                                                                        ELF0060
C  NEW GEOMETRY, START A NEW SEQUENCE                                    ELF0061
   60 IBT = 0                                                            ELF0062
      IBST = 0                                                           ELF0063
      IBS = 0                                                            ELF0064
      REWIND ISAVU                                                       ELF0065
C                                                                        ELF0066
C  NEW GEOMETRY, CONTINUE SEQUENCE                                       ELF0067
   70 IBT = IBT + 1                                                      ELF0068
      IF (IBT .GT. 10) GO TO 80                                          ELF0069
      IB = IBT                                                           ELF0070
      IBMF(IB) = 1                                                       ELF0071
      GO TO 250                                                          ELF0072
   80 WRITE (6,90) IBT, IBMAX                                            ELF0073
   90 FORMAT (1H1, 'ATTEMPTED TO LOAD THE ',I2,'TH BODY.  MAXIMUM ',     ELF0074
     1              'ALLOWABLE NUMBER OF BODIES IS ',I3)                 ELF0075
      GO TO 640                                                          ELF0076
C                                                                        ELF0077
C  NEW GEOMETRY, OLD SEQUENCE                                            ELF0078
  100 IF (IBT .LE. 0) GO TO 120                                          ELF0079
      DO 110 IB = 1,IBT                                                  ELF0080
      IF (IDB(IB) .EQ. IDOLD) GO TO 240                                  ELF0081
  110 CONTINUE                                                           ELF0082
  120 WRITE (6,130) IBOD, IDOLD                                          ELF0083
  130 FORMAT (1H1,'OPTION IBOD = ',I3,', GEOMETRY WITH ID = ', I3,       ELF0084
     1              ' NOT PREVIOUSLY LOADED.')                           ELF0085
      GO TO 640                                                          ELF0086
C                                                                        ELF0087
C  OLD GEOMETRY, OLD (OR CONTINUE) SEQUENCE                              ELF0088
  140 IF (IBT .LE. 0) GO TO 120                                          ELF0089
      DO 150 IB = 1,IBT                                                  ELF0090
      IF (IDB(IB) .EQ. IDOLD) GO TO 160                                  ELF0091
  150 CONTINUE                                                           ELF0092
      GO TO 120                                                          ELF0093
C                                                                        ELF0094
C  DESIRE OLD GEOMETRY. IF SAVED, RETRIEVE                               ELF0095
  160 IF (ISAV3(IB) .LT. 0)  GO TO 190                                   ELF0096
      IBS = ISAV3(IB)                                                    ELF0097
      REWIND ISAVU                                                       ELF0098
      DO 170 I = 1,IBS                                                   ELF0099
      LX = ISAV2(IBS)                                                    ELF0100
      CALL  GETT(ISAVU, 3, LX, X, LX, Y)                                 ELF0101
  170 CONTINUE                                                           ELF0102
      IF (IBOD .EQ. 5)  GO TO 180                                        ELF0103
      IBMF(IB) = 2                                                       ELF0104
      GO TO 360                                                          ELF0105
  180 IBT = IBT + 1                                                      ELF0106
      IF (IBT .GT. 10) GO TO 80                                          ELF0107
      IB = IBT                                                           ELF0108
      IBMF(IB) = 1                                                       ELF0109
      ISAV1(IB) = IDOLD                                                  ELF0110
      ISAV3(IB) = -1                                                     ELF0111
      GO TO 360                                                          ELF0112
```

```
      C                                                                 ELF0113
  190 WRITE (6,200) IBOD, IDOLD                                         ELF0114
  200 FORMAT (1H1,'OPTION IBOD = ',I3,', GEOMETRY WITH ID = ',I3,       ELF0115
     1                  ' NOT PREVIOUSLY SAVED.')                       ELF0116
      GO TO 640                                                         ELF0117
C                                                                       ELF0118
C  DELETE AN EXISTING BODY                                              ELF0119
  210 IF (IBT .LE. 0) GO TO 120                                         ELF0120
      DO 220 IB = 1,IBT                                                 ELF0121
      IF (IDB(IB) .EQ. IDOLD) GO TO 230                                 ELF0122
  220 CONTINUE                                                          ELF0123
  230 IBMF(IB) = -1                                                     ELF0124
      GO TO 600                                                         ELF0125
C                                                                       ELF0126
C  NEW GEOMETRY TO BE READ IN.                                          ELF0127
  240 IBMF(IB) = 2                                                      ELF0128
C  POSITION SAVE UNIT IF NEW GEOMETRY TO BE SAVED.                      ELF0129
  250 IF (ISV .EQ. 0) GO TO 270                                         ELF0130
      IF (IBS .EQ. IBST) GO TO 270                                      ELF0131
      DO 260 II = IBS,IBST                                              ELF0132
      LX = ISAV2(II)                                                    ELF0133
  260 CALL  GETT(ISAVU, 3, LX, X, LX, Y)                                ELF0134
  270 CONTINUE                                                          ELF0135
C                                                                       ELF0136
C  CHECK IF ELLIPSE TO BE GENERATED                                     ELF0137
      IF (ITR .GT. 1)  GO TO 320                                        ELF0138
C                                                                       ELF0139
C  DATA ON UNIT 5.  X-COORDS FIRST                                      ELF0140
      L = 0                                                             ELF0141
      ITYP = 3                                                          ELF0142
  280 READ (5,290)  (X(L+I), I=1,6), INO, ISTAT, ITYPE                  ELF0143
  290 FORMAT (6F10.0, 4XI1, 2XI1, 3XI1)                                 ELF0144
      IF (ITYPE .NE. ITYP)  CALL TYPE(ITYP, ITYPE)                      ELF0145
      IF (INO .LE. 0  .OR.  INO .GT. 6) INO = 6                         ELF0146
      L = L + INO                                                       ELF0147
      IF (ISTAT .EQ. 0) GO TO 280                                       ELF0148
      LX = L                                                            ELF0149
C                                                                       ELF0150
C  NOW READ IN Y-COORDS                                                 ELF0151
      L = 0                                                             ELF0152
      ITYP = 4                                                          ELF0153
  300 READ (5,290)  (Y(L+I), I=1,6), INO, ISTAT, ITYPE                  ELF0154
      IF (ITYPE .NE. ITYP)  CALL TYPE(ITYP, ITYPE)                      ELF0155
      IF (INO .LE. 0  .OR.  INO .GT. 6) INO = 6                         ELF0156
      L = L + INO                                                       ELF0157
      IF (ISTAT .EQ. 0) GO TO 300                                       ELF0158
      LY = L                                                            ELF0159
C                                                                       ELF0160
C  CHECK FOR INPUT CONSISTENCY                                          ELF0161
      IF (LY .EQ. LX) GO TO 350                                         ELF0162
      WRITE (6,310) LY, LX                                              ELF0163
  310 FORMAT (1H1,'THE NUMBER OF Y-COORDINATES (',I3,') READ DOES ',    ELF0164
     1             'NOT EQUAL THE NUMBER OF X-COORDINATES READ (',I3,')')ELF0165
      GO TO 640                                                         ELF0166
C                                                                       ELF0167
C  ELLIPSE TO BE GENERATED.  READ IN DEFINITION CARD.                   ELF0168
  320 ITYP = 5                                                          ELF0169
```

```
      READ (5,330) LX, ELPSTH, ITYPE                                  ELF0170
  330 FORMAT (2XI3, 5XF10.5, 51XI1)                                   ELF0171
      IF (ITYPE .NE. ITYP)  CALL TYPE(ITYP, ITYPE)                    ELF0172
      ITR = ITR - 2                                                   ELF0173
      IF (ITR .NE. 1)  ITR = 0                                        ELF0174
C                                                                     ELF0175
      DANGLE = 6.2831853072/(LX - 1)                                  ELF0176
      ANGLE = DANGLE                                                  ELF0177
      DO 340 I = 1,LX                                                 ELF0178
      ANGLE = ANGLE - DANGLE                                          ELF0179
      X(I) = COS(ANGLE)                                               ELF0180
  340 Y(I) = SIN(ANGLE)*ELPSTH                                        ELF0181
C                                                                     ELF0182
C                                                                     ELF0183
C  SAVE THE BASIC GEOMETRY IF REQUESTED                               ELF0184
  350 ISAV3(IB) = -1                                                  ELF0185
      IF (ISV .EQ. 0)  GO TO 360                                      ELF0186
C                                                                     ELF0187
C                                                                     ELF0188
      IBST = IBST + 1                                                 ELF0189
      IBS = IBST                                                      ELF0190
      ISAV1(IB) = ID                                                  ELF0191
      ISAV2(IBS) = LX                                                 ELF0192
      ISAV3(IB) = IBS                                                 ELF0193
      CALL  SAVE(ISAVU, 3, 1, LX, X, LX, Y)                           ELF0194
C                                                                     ELF0195
C                                                                     ELF0196
  360 CONTINUE                                                        ELF0197
C                                                                     ELF0198
C  WRITE OUT BASIC GEOMETRY DATA                                      ELF0199
      IP = 1                                                          ELF0200
      CALL PRINTG (IP, LX, ID, TTITLE)                                ELF0201
C                                                                     ELF0202
C                                                                     ELF0203
C  TRANSFORM COORDINATES IF REQUESTED                                 ELF0204
      IF (ITR .EQ. 1) GO TO 370                                       ELF0205
      IF (INORM .EQ. 0) GO TO 410                                     ELF0206
      XMULT = 0.0                                                     ELF0207
      YMULT = 0.0                                                     ELF0208
      XTO = 0.0                                                       ELF0209
      YTO = 0.0                                                       ELF0210
      THETA = 0.0                                                     ELF0211
      DX = 0.0                                                        ELF0212
      DY = 0.0                                                        ELF0213
  370 CONTINUE                                                        ELF0214
C                                                                     ELF0215
      IF (ABS(XMULT) .LT. EPS)      XMULT = 1.0                       ELF0216
      IF (ABS(YMULT) .LT. EPS)      YMULT = 1.0                       ELF0217
      XSF = XMULT                                                     ELF0218
      YSF = YMULT                                                     ELF0219
      IF (INORM .EQ. 0)  GO TO 390                                    ELF0220
C                                                                     ELF0221
      IF (ABS(CHORD) .GT. EPS)  GO TO 380                             ELF0222
      CHORD = RMAX (LX, X, Y, TMAX)                                   ELF0223
  380 XSF = XSF/CHORD                                                 ELF0224
      YSF = YSF/CHORD                                                 ELF0225
C                                                                     ELF0226
```

```
  390 COST = COS(THETA*DR)                                              ELF0227
      SINT = SIN(THETA*DR)                                              ELF0228
      DO 400 I = 1,LX                                                   ELF0229
      XTOD = X(I) - XTO                                                 ELF0230
      YTOD = Y(I) - YTO                                                 ELF0231
      X(I) = (XTO + XTOD*COST - YTOD*SINT + DX)*XSF                     ELF0232
      Y(I) = (YTO + XTOD*SINT + YTOD*COST + DY)*YSF                     ELF0233
  400 CONTINUE                                                          ELF0234
C                                                                       ELF0235
C                                                                       ELF0236
C  FORM ELEMENT DATA FOR THIS BODY.                                     ELF0237
C  DEFINE STORAGE LOCATIONS AND CROSS CHECK                             ELF0238
  410 IST = 0                                                           ELF0239
      IF (IB .GT. 1)  IST = IFL(IB-1)                                   ELF0240
      LX1 = LX-1                                                        ELF0241
C  FIRST CHECK AGAINST EXCEEDING MAXIMUM STORAGE                        ELF0242
      IF ((IST + LX1) .LE. MAXEL)  GO TO 430                            ELF0243
      WRITE (6,420) LX1, ID, MAXEL                                      ELF0244
  420 FORMAT (1H1,'THE NUMBER OF ELEMENTS (',I4,') FOR BODY ID = ',I2,  ELF0245
     1       ' WILL EXCEED ALLOWABLE STORAGE (',I4,') WHEN ADDED ',     ELF0246
     2       'TO THE DATA SET.')                                        ELF0247
      GO TO 640                                                         ELF0248
C                                                                       ELF0249
C  NOW, IF USING OLD STORAGE SEQUENCE, CHECK THAT NEW GEOMETRY          ELF0250
C  DOES NOT RUN INTO THE NEXT BODY.                                     ELF0251
  430 IF (IB .EQ. IBT) GO TO 450                                        ELF0252
      IF ((IST + LX1) .LT. INL(IB+1))  GO TO 460                        ELF0253
      WRITE (6,440) LX1, ID, NL(IB), IDOLD                              ELF0254
  440 FORMAT (1H1,'THE NUMBER OF ELEMENTS (',I4,') FOR THE NEW BODY ',  ELF0255
     1       'ID = ',I2,' EXCEEDS THE NUMBER (',I4,') FOR THE BODY ',   ELF0256
     2       'IT IS REPLACING, IDOLD = ',I2)                            ELF0257
      GO TO 640                                                         ELF0258
C                                                                       ELF0259
C                                                                       ELF0260
  450 INL(IB) = IST + 1                                                 ELF0261
      IFL(IB) = IST + LX1                                               ELF0262
  460 NL(IB) = LX1                                                      ELF0263
      LIFT(IB) = ILIFT                                                  ELF0264
      IDB(IB) = ID                                                      ELF0265
      ITRB(IB) = ITR                                                    ELF0266
      INMB(IB) = INORM                                                  ELF0267
      IBD(IB)  = IBOD                                                   ELF0268
      CHORDB(IB) = CHORD                                                ELF0269
      IPRB(IB) = IPARA                                                  ELF0270
      IFST(IB) = IFIRST                                                 ELF0271
      ISEC(IB) = ISECND                                                 ELF0272
      IPVR(IB) = IPVOR                                                  ELF0273
C                                                                       ELF0274
      DO 470 I = 1,7                                                    ELF0275
  470 BTITLE(IB,I) = TTITLE(I)                                          ELF0276
C                                                                       ELF0277
C  CALCULATE ELEMENT DATA AND PRINT RESULTS.                            ELF0278
C                                                                       ELF0279
      DO 480 I = 2,LX                                                   ELF0280
      IST = IST + 1                                                     ELF0281
      XO(IST) = 0.5*(X(I) + X(I-1))                                     ELF0282
      YO(IST) = 0.5*(Y(I) + Y(I-1))                                     ELF0283
```

```
      XD = X(I) - X(I-1)                                               ELF0284
      YD = Y(I) - Y(I-1)                                               ELF0285
      DL(IST)  = SQRT(XD**2 + YD**2)                                   ELF0286
      DS(IST)  = DL(IST)                                               ELF0287
      SA(IST)  = YD/DL(IST)                                            ELF0288
      CA(IST)  = XD/DL(IST)                                            ELF0289
      CURV(IST) = 0.0                                                  ELF0290
  480 CONTINUE                                                         ELF0291
C                                                                      ELF0292
C                                                                      ELF0293
C  PARABOLIC ELEMENT                                                   ELF0294
      IST = INL(IB)                                                    ELF0295
      LEND = LX - 2                                                    ELF0296
      I = 1                                                            ELF0297
  490 I = I + 1                                                        ELF0298
      IST = IST + 1                                                    ELF0299
      I1 = I - 1                                                       ELF0300
      I2 = I                                                           ELF0301
      I3 = I + 1                                                       ELF0302
      I4 = I + 2                                                       ELF0303
      L1 = I1                                                          ELF0304
      L2 = I2                                                          ELF0305
      L3 = I3                                                          ELF0306
C                                                                      ELF0307
  500 DLY = (X(L3)-X(L1))*(Y(L2)-Y(L1)) - (X(L2)-X(L1))*(Y(L3)-Y(L1))  ELF0308
      DLX = (X(L2)-X(L1))*(X(L3)-X(L1)) + (Y(L2)-Y(L1))*(Y(L3)-Y(L1))  ELF0309
      DRY = (X(I2)-X(I4))*(Y(I3)-Y(I4)) - (X(I3)-X(I4))*(Y(I2)-Y(I4))  ELF0310
      DRX = (X(I2)-X(I4))*(X(I3)-X(I4)) + (Y(I3)-Y(I4))*(Y(I2)-Y(I4))  ELF0311
      DYO = 0.0                                                        ELF0312
      DLL =SQRT(((X(L2)-X(L1))**2 + (Y(L2)-Y(L1))**2)                  ELF0313
     1          *((X(L3)-X(L1))**2 + (Y(L3)-Y(L1))**2))                ELF0314
      DRL =SQRT(((X(I2)-X(I4))**2  +  (Y(I2)-Y(I4))**2)                ELF0315
     1          * ((X(I3)-X(I4))**2  +   (Y(I3)-Y(I4))**2))            ELF0316
      ETAL =  0.5*DL(IST)*DLY/(DLL  + DLX)                             ELF0317
      ETAR = -0.5*DL(IST)*DRY/(DRL  + DRX)                             ELF0318
C                                                                      ELF0319
C  DEFINE AVERAGE OFFSET DYO USING RMS.                                ELF0320
      ETA = ETAL*ETAR                                                  ELF0321
      IF (ETA .LT. 0.0)  GO TO 510                                     ELF0322
      ETA = SQRT(ETA)                                                  ELF0323
      DYO = SIGN(ETA, ETAL)                                            ELF0324
      CURV(IST) = -4.0*DYO/DL(IST)**2                                  ELF0325
C                                                                      ELF0326
C                                                                      ELF0327
  510 CONTINUE                                                         ELF0328
      XP(IST) = XO(IST) - DYO*SA(IST)                                  ELF0329
      YP(IST) = YO(IST) + DYO*CA(IST)                                  ELF0330
      DS(IST) = DL(IST)*(1.0 + (CURV(IST)*DL(IST))**2/6.0)             ELF0331
C                                                                      ELF0332
      IF (I - LEND)490,520,530                                         ELF0333
C                                                                      ELF0334
C  SPECIAL HANDLING FOR FIRST AND LAST ELEMENTS                        ELF0335
C                                                                      ELF0336
C  FIRST ELEMENT                                                       ELF0337
  520 IST = INL(IB)                                                    ELF0338
      I2 = 1                                                           ELF0339
      I3 = 2                                                           ELF0340
```

```
      I4 = 3                                                            ELF0341
      L1 = I2                                                           ELF0342
      L2 = I3                                                           ELF0343
      L3 = I4                                                           ELF0344
      I = LEND + 1                                                      ELF0345
      GO TO 500                                                         ELF0346
C                                                                       ELF0347
C  LAST ELEMENT                                                         ELF0348
  530 IF (I .GT. LEND+1)  GO TO 540                                     ELF0349
      IST = IFL(IB)                                                     ELF0350
      L1 = LX - 2                                                       ELF0351
      L2 = LX - 1                                                       ELF0352
      L3 = LX                                                           ELF0353
      I2 = L1                                                           ELF0354
      I3 = L2                                                           ELF0355
      I4 = L3                                                           ELF0356
      I = I + 1                                                         ELF0357
      GO TO 500                                                         ELF0358
C                                                                       ELF0359
C  SAVE PARABOLIC SURFACE DATA ON UNIT IF17                             ELF0360
  540 CONTINUE                                                          ELF0361
C                                                                       ELF0362
      I1 = INL(IB)                                                      ELF0363
      I2 =  NL(IB)                                                      ELF0364
C  XP AND YP COORDS                                                     ELF0365
      CALL SAVE(IF17, 3, I2, I2, XP(I1), I2, YP(I1))                    ELF0366
      I2 = IFL(IB)                                                      ELF0374
C  CHECK IF PARABOLIC ELEMENT DESIRED                                   ELF0375
      IF (IPARA .NE. 1)  GO TO 560                                      ELF0376
      DO 550 I =I1,I2                                                   ELF0377
      XO(I) = XP(I)                                                     ELF0378
  550 YO(I) = YP(I)                                                     ELF0379
C                                                                       ELF0380
C  PRINT ELEMENT DATA                                                   ELF0381
  560 I = 1                                                             ELF0382
      WRITE (6,610)  IDB(IB),TTITLE                                     ELF0383
      WRITE (6,620)  I, X(I), Y(I)                                      ELF0384
      LCNT = 1                                                          ELF0385
      SUMDS(IB) = 0.0                                                   ELF0386
      ALPH1 = 0.0                                                       ELF0387
      DO 580 IST = I1, I2                                               ELF0388
      I = I + 1                                                         ELF0389
      IF (LCNT .LT. 49)  GO TO 570                                      ELF0390
      LCNT = 0                                                          ELF0391
      WRITE (6,610)  IDB(IB),TTITLE                                     ELF0392
  570 CONTINUE                                                          ELF0393
      SUMDS(IB) = SUMDS(IB) + DS(IST)                                   ELF0394
      ALPHA = ATAN2(SA(IST), CA(IST))/CR                                ELF0395
      DADS =  (ALPHA - ALPH1)/DS(IST)*CR                                ELF0396
      ALPH1 = ALPHA                                                     ELF0397
      CURV2 = 2.0*CURV(IST)                                             ELF0398
      WRITE (6,630) XO(IST), YO(IST), CL(IST), DS(IST),                 ELF0399
     1              SA(IST), CA(IST), CURV2                             ELF0400
      WRITE (6,620)  I, X(I), Y(I)                                      ELF0401
      LCNT = LCNT + 2                                                   ELF0402
  580 CONTINUE                                                          ELF0403
C                                                                       ELF0404
```

```
C                                                                    ELF0405
      WRITE (6,590) SUMDS(IB)                                        ELF0406
  590 FORMAT(1H0, T13, 'SUMDS = ', F12.6)                            ELF0407
C                                                                    ELF0408
C  DETERMINE GEOMETRY COMBINATION COEFFICIENTS                       ELF0409
      CALL GEOMCF (INL(IB), IFL(IB), DL, IFST(IB), ISEC(IB), DL)     ELF0410
C                                                                    ELF0411
C                                                                    ELF0412
C  CHECK IF MORE BODIES TO BE INPUT                                  ELF0413
  600 IF (LAST .NE. 1)  GO TO 10                                     ELF0414
C                                                                    ELF0415
C                                                                    ELF0416
C  WRITE OUT BODY SUMMARY                                            ELF0417
      CALL PRNTEL                                                    ELF0418
C                                                                    ELF0419
      GO TO 660                                                      ELF0420
C                                                                    ELF0421
C                                                                    ELF0422
C                                                                    ELF0423
  610 FORMAT (1H1, 20X,'ELEMENT COORDINATE DATA FOR BODY ID = ', I2, ELF0424
     1       ',  ',  7A4//T10,'I', T22,'X(I)', T39,'Y(I)', T56,'DL', ELF0425
     2       T73,'DS', T86,' SIN(ALF) ', T105,'COS(ALF)',            ELF0426
     3       T121,'CURVATURE'//)                                     ELF0427
C                                                                    ELF0428
  620 FORMAT (1H , 6XI3, 2(5XF12.6))                                 ELF0429
C                                                                    ELF0430
  630 FORMAT (1H , 9X,   7(5XF12.6))                                 ELF0431
C                                                                    ELF0432
  640 WRITE (6,650)                                                  ELF0433
  650 FORMAT (1H0, 'BECAUSE OF THE ABOVE ERROR, THIS RUN IS TERMINATED')ELF0434
      STOP                                                           ELF0435
C                                                                    ELF0436
C                                                                    ELF0437
  660 CONTINUE                                                       ELF0438
C                                                                    ELF0439
      RETURN                                                         ELF0440
      END                                                            ELF0441
```

```
      SUBROUTINE    FILES                                                FILE001
      COMMON /FILEIO/ IFILE1, IFILE2, IFILE3, IFILE4, IFILE5,           FILE002
     1                IFILE6, IFILE7, IFILE8, IFILE9, IFIL10,           FILE003
     2                IFIL11, IFIL12, IFIL13, IFIL14, IFIL15            FILE004
     3                IFIL16, IFIL17, IFIL18, IFIL19, IFIL20            FILE005
C                                                                       FILE006
C IFILE1, IFILE2, AND IFILE3 ARE TEMPORARY SCRATCH UNITS USED IN QUASI  FILE007
      IFILE1 = 18                                                       FILE008
      IFILE2 = 2                                                        FILE009
      IFILE3 = 3                                                        FILE010
C IFILE4 IS INPUT FILE FOR RIGHT SIDE MATRIX IN QUASI                   FILE011
      IFILE4 = 4                                                        FILE012
C                                                                       FILE013
C IFILE5, IFILE6, AND IFILE7 ARE STANDARD SYSTEM I/O                    FILE014
      IFILE5 = 5                                                        FILE015
      IFILE6 = 6                                                        FILE016
      IFILE7 = 7                                                        FILE017
C                                                                       FILE018
C IFILE8 IS INPUT GEOMETRY SAVE UNIT                                    FILE019
      IFILE8 = 8                                                        FILE020
C                                                                       FILE021
C IFILE9 IS MATRIX A(I,J)                                               FILE022
C IFIL10 IS MATRIX B(I,J)                                               FILE023
      IFILE9 = 9                                                        FILE024
      IFIL10 = 10                                                       FILE025
C IFIL11 IS INDUCED NORMAL VELOCITY  N(I).                              FILE026
C IFIL12 IS INDUCED TANGENTIAL VELOCITY T(I)                            FILE027
      IFIL11 = 11                                                       FILE028
      IFIL12 = 12                                                       FILE029
C                                                                       FILE030
C IFIL13 IS SPECIAL B ROWS FOR LIFTING BODIES, BLU(J)                   FILE031
      IFIL13 = 13                                                       FILE032
C                                                                       FILE033
C IFIL14 CONTAINS SIGMA SOLUTIONS ON OUTPUT FROM QUASI                  FILE034
      IFIL14 = 14                                                       FILE035
C                                                                       FILE036
C IFIL15 CONTAINS BOTH UPPER AND LOWER TRIANGULAR MATRICES              FILE037
C ON OUPUT FROM QUASI                                                   FILE038
      IFIL15 = 15                                                       FILE039
C                                                                       FILE040
C IFIL16 IS USED FOR OFFBODY CALCULATIONS.  /ELDATA/  AND  /GCOEFS/     FILE041
C ARE STORED  (SEE  SUBROUTINE ELFORM).                                 FILE042
      IFIL16 = 16                                                       FILE043
C                                                                       FILE044
C IFIL17 IS USED TO SAVE SURFACE COORDS (X, Y, AND DS)  AT WHICH        FILE045
C THE FLOW VELOCITY IS ASSUMED TO ACT.                                  FILE046
      IFIL17 = 17                                                       FILE047
C                                                                       FILE048
C IFILS 18, 19, AND 20 HAVE NOT BEEN ASSIGNED.                          FILE049
C                                                                       FILE050
      RETURN                                                            FILE051
      END                                                               FILE052
```

```
      SUBROUTINE FLOWS (N, M, IPUN)
C                                                                       FLOW001
C INDIVIDUAL FLOWS (IND = 1) AND COMBINED FLOWS CALCULATED.             FLOW002
C                                                                       FLOW003
C                                                                       FLOW004
      DIMENSION  A(500), B(500), VN(500), VT(500), VNC(500), VTC(500),  FLOW005
     1         SIG(500)                                                 FLOW006
     2   , X(500), Y(500), DS(500), SA(500), CA(500)                    FLOW007
      DIMENSION  ELGC(3500), ELGD(2500), CPJ(500), XP(500), YP(500),    FLOW008
     1    V1(500),V2(500),V3(500),V4(500),V5(500)                       FLOW009
C
      COMMON /BFLAG/ IDB(10), INL(10), IFL(10), NL(10), LIFT(10),       FLOW010
     1              IBMF(10), ISAV1(10), ISAV2(10), ISAV3(10),          FLOW011
     2            BTITLE(10,  7), IBT, IBST, IBTOT, NELTOT,             FLOW012
     3              ITRB(10), INMB(10), CHORD(10), IBD(10), LIFTOT,     FLOW013
     4              IPRB(10), IFST(10), ISEC(10), FTITLE(15), IPVR(10)  FLOW014
C                                                                       FLOW015
      COMMON /SIGMAS/ CSIG(500), CK(12)                                 FLOW016
      COMMON /FILEID/ IF01, IF02, IF03, IF04, IF05, IF06, IF07, IF08,   FLOW017
     1                IF09, IF10, IF11, IF12, IF13, IF14, IF15          FLOW018
     2             ,IF16, IF17, IF18, IF19, IF20                        FLOW019
      COMMON /COMROD/ CCL,INCLT,CLT,ALPHA,SUMDS(10),TLU(10,12), IND     FLOW020
     1             , ALPHAO, CNU(10), SMDSWF(10), MIO(10)               FLOW021
      COMMON/ROTAT/NROT, ROTRAD(10)                                     FLOW022
      COMMON /GCF/    CD(500), CF(500), CG(500), CI(500), WF(500)       FLOW023
      COMMON /ELD/    X, Y, DS, SA, CA, CURV(500), DL(500)              FLOW024
      COMMON/COM/IFLLL                                                  FLOW025
      EQUIVALENCE  (ELGC(1), X(1)),  (ELGD(1), CD(1)),(CPJ(1), SIG(1))
      EQUIVALENCE  (A(1), XP(1)),  (B(1), YP(1))                        FLOW026
C                                                                       FLOW027
C REWIND UNITS FOR NORMAL AND TANGENTIAL ONSET VELOCITIES               FLOW028
      REWIND IF11                                                       FLOW029
      REWIND IF12                                                       FLOW030
C                                                                       FLOW031
C REWIND SIGMA UNIT                                                     FLOW032
      REWIND IF14                                                       FLOW033
C                                                                       FLOW034
C ZERO OUT VNC AND VTC ARRAYS                                           FLOW035
      DO 10 I = 1,N                                                     FLOW036
      VNC(I) = 0.0                                                      FLOW037
   10 VTC(I) = 0.0                                                      FLOW038
C                                                                       FLOW039
C                                                                       FLOW040
C THE PROCEDURE IS TO FIRST CALCULATE THE INDIVIDUAL FLOWS              FLOW041
C AND THEN THE COMBINED FLOW.  THE ONSET VELOCITIES ARE                 FLOW042
C COMBINED DURING THE INDIVIDUAL FLOWS CYCLE.                           FLOW043
C                                                                       FLOW044
C READ IN SURFACE COORDS AND GEOMETRY COEFFICIENTS FROM UNIT 16         FLOW045
      REWIND IF16                                                       FLOW046
      READ(IF16) ELGC, ELGD                                             FLOW047
C                                                                       FLOW048
C                                                                       FLOW049
C                                                                       FLOW050
C BEGIN INDIVIDUAL FLOWS CYCLE                                          FLOW051
      VNA = 0.0                                                         FLOW052
      DO 90 K = 1,M                                                     FLOW053
                                                                        FLOW054
```

```
C                                                                       FLOW055
C READ IN NORMAL AND TANGENTIAL ONSET VELOCITIES.                         FLOW056
      CALL  GETT(IF11, 1, N, VN, 1, VNA)                                FLOW057
      CALL  GETT(IF12, 1, N, VT, 1, VNA)                                FLOW058
C                                                                       FLOW059
C CALCULATE COMBINED NORMAL & TANGENTIAL VELOCITIES                      FLOW060
      DO 30 I = 1,N                                                     FLOW061
      VNC(I) = VNC(I) - VN(I)*CK(K)                                     FLOW062
   30 VTC(I) = VTC(I) + VT(I)*CK(K)                                     FLOW063
C                                                                       FLOW064
C CHECK IF INDIVIDUAL FLOW DESIRED                                      FLOW065
      IF (IND .NE. 1)  GO TO 90                                         FLOW066
C                                                                       FLOW067
C REWIND A,B ARRAY UNITS                                                FLOW068
      REWIND IF10                                                       FLOW069
C                                                                       FLOW070
C READ IN ROW OF SIGMAS                                                 FLOW071
      CALL  GETT(IF14, 1, N, SIG,1, VNA)                                FLOW072
C                                                                       FLOW073
      DO 50 I = 1,N                                                     FLOW074
C                                                                       FLOW075
C READ IN ROW OF A,B ARRAYS                                             FLOW076
      CALL  GETT(IF10, 1, N, A,  1, VNA)                                FLOW077
      CALL  GETT(IF10, 1, N, B,  1, VNA)                                FLOW078
C                                                                       FLOW079
      VN(I) = -VN(I)                                                    FLOW080
      VNA = 0.0                                                         FLOW081
      VTB = 0.0                                                         FLOW082
      DO 40 J = 1,N                                                     FLOW083
      VNA = VNA + A(J)*SIG(J)                                           FLOW084
   40 VTB = VTB + B(J)*SIG(J)                                           FLOW085
C                                                                       FLOW086
      VN(I) = VN(I) + VNA                                               FLOW087
   50 VT(I) = VT(I) + VTB                                               FLOW088
C                                                                       FLOW089
C                                                                       FLOW090
C PRINT OUT INDIVIDUAL FLOW                                             FLOW091
      WRITE (6,60) K                                                    FLOW092
   60 FORMAT(1H1, 'INDIVIDUAL FLOW NUMBER', I3//' PT.NO.',              FLOW093
     1       T19,'VN', T39, 'VT',  T58, 'SIGMA')                        FLOW094
      DO 70 I = 1,N                                                     FLOW095
      IF(K.EQ.1) V1(I)=VT(I)
      IF(K.EQ.2) V2(I)=VT(I)
      IF(K.EQ.3) V3(I)=VT(I)
      IF(K.EQ.4) V4(I)=VT(I)
      IF(K.EQ.5) V5(I)=VT(I)
   70 WRITE(6,80) I, VN(I), VT(I), SIG(I)                               FLOW096
   80 FORMAT(1H , I4, 3(8XF12.6))                                       FLOW097
C                                                                       FLOW098
C                                                                       FLOW099
   90 CONTINUE                                                          FLOW100
C                                                                       FLOW101
C INDIVIDUAL FLOWS COMPLETE.  NOW DO COMBINED FLOW                      FLOW102
C                                                                       FLOW103
      IF(IFLLL.EQ.0) GO TO 161
      REWIND IF10                                                       FLOW104
      DO 110 I = 1,N                                                    FLOW105
```

```
      CALL  GETT(IF10, 1, N, A,  1, VNA)                                FLOW106
      CALL  GETT(IF10, 1, N, B,  1, VNA)                                FLOW107
C                                                                       FLOW108
      VNA = 0.0                                                         FLOW109
      VTB = 0.0                                                         FLOW110
      DO 100 J = 1,N                                                    FLOW111
      VNA = VNA + A(J)*CSIG(J)                                          FLOW112
  100 VTB = VTB + B(J)*CSIG(J)                                          FLOW113
C                                                                       FLOW114
      VN(I) = VNC(I) + VNA                                              FLOW115
      VT(I) = VTC(I) + VTB                                              FLOW116
  110 CONTINUE                                                          FLOW117
C                                                                       FLOW118
C                                                                       FLOW119
C  PRINT THE OUTPUT DATA  (PER BODY).                                   FLOW120
      ALPH = ALPHA*0.017453293                                          FLOW121
      COSA = COS(ALPH)                                                  FLOW122
      SINA = SIN(ALPH)                                                  FLOW123
      CMT = 0.0                                                         FLOW124
      CMTP = 0.0                                                        FLOW125
      XM = 0.0                                                          FLOW126
      YM = 0.0                                                          FLOW127
      NI = 0                                                            FLOW128
      NF = 0                                                            FLOW129
      REWIND  IF17                                                      FLOW130
      DO 160 IB = 1,IBTOT                                               FLOW131
      IF (IBNF(IB) .LT. 0)  GO TO 160                                   FLOW132
      NB = NL(IB)                                                       FLOW133
      NI = NF                                                           FLOW134
      NF = NF + NB                                                      FLOW135
      J2 = NI                                                           FLOW136
      S = 0.0                                                           FLOW137
      S1 = 0.0                                                          FLOW138
      I = 0                                                             FLOW139
      CX = 0.0                                                          FLOW140
      CN = 0.0                                                          FLOW141
      CML= 0.0                                                          FLOW142
      N1 = NI + 1                                                       FLOW143
      CALL  GETT(IF17, 1, NB, XP(N1), NP, YP(N1))                       FLOW144
      IF (NROT .NE. 0)  ROT2 = ROTRAD(IB)**2                            FLOW145
  120 J1 = J2 + 1                                                       FLOW146
      J2 = J1 + 48                                                      FLOW147
      IF (J2 .GT. NF)  J2 = NF                                          FLOW148
      WRITE(IF06,130)  FTITLE, ALPHA, ALPHA0, IBTOT, CLT, CCL,          FLOW149
     1   NELTOT, IDB(IB), (BTITLE(IB,II),II=1,7),  NB                   FLOW150
C                                                                       FLOW151
  130 FORMAT(1H1,'   DOUGLAS  AIRCRAFT  COMPANY  TWO-DIMENSIONAL ',     FLOW152
     1   'POTENTIAL  FLOW  PROGRAM'/1H0,'COMBINED  FLOW     ', 15A4/    FLOW153
     2   1H0, 'ALPHA =', F11.6, T30,'ALPHA C =', F11.6,                 FLOW154
     3   T60,'NO.  OF  BODIES    ', I2/                                 FLOW155
     4   1H0, '  CL  =', F11.6, T30,'  CHORD =', F11.6,                 FLOW156
     5   T60,'TOTAL  ELEMENTS  ',  I3/                                  FLOW157
     6   1H0,'BODY ID = ', I2, T20, 7A4, T60,'NO. OF ELEMENTS  ', I3/   FLOW158
     7   1H0, T4,'I', T14,'X', T28,'Y', T42,'S', T56,'VT',              FLOW159
     8   T70,'CP', T93,'-', T101,'SIGMA', T117,'VN')                    FLOW160
C                                                                       FLOW161
      DO 140 J = J1,J2                                                  FLOW162
```

```
      I = I + 1                                                         FLOW163
      SD = DS(J)/(2.0*SUMDS(IB))                                        FLOW164
      S = S1 + SD                                                       FLOW165
      S1 = S + SD                                                       FLOW166
      CP = 1.0 - VT(J)**2                                               FLOW167
      CPJ(J) = CP                                                       FLOW168
      DCP = 0.0                                                         FLOW169
      IF (NROT .EQ. 0)  GO TO 145                                       FLOW170
      DCP = (X(J)**2 + Y(J)**2)/ROT2                                    FLOW171
      DCP = DCP - 2.0*(X(J)*SINA - Y(J)*COSA)/ROTRAD(IB)                FLOW172
  145 CP = CP + DCP                                                     FLOW173
      XX = XP(J)                                                        FLOW174
      YY = YP(J)                                                        FLOW175
      T = CP*DS(J)                                                      FLOW176
      CN = CN - T*CA(J)                                                 FLOW177
      CX = CX + T*SA(J)                                                 FLOW178
      CML = CML + T*(CA(J)*(X(J)-XM) + SA(J)*(Y(J)-YM))                 FLOW179
      IF (IPUN .EQ. 7)  WRITE(7, 155)  XX, YY, CP, I                    FLOW180
  140 WRITE(IF06,150)  I,XX,YY, S, VT(J), CP, J, CSIG(J), VN(J)         FLOW181
  150 FORMAT(1H , I3, 5(3XF11.6), 16XI1, 2(3XF11.6))                    FLOW182
C                                                                       FLOW183
  155 FORMAT(3F10.5, 46XI4)                                             FLOW184
C                                                                       FLOW185
      IF (J2 .NE. NF)  GO TO 120                                        FLOW186
      CLI = CN*COSA - CX*SINA                                           FLOW187
      CDI = CN*SINA + CX*COSA                                           FLOW188
      WRITE(6,200)  CN, CX, CLI, CDI, CML                               FLOW189
  200 FORMAT(1H0,17HINTEGRATED VALUES/                                  FLOW190
     1       1H0, 5HCY = , F10.5, 5X5HCX = , F10.5/                     FLOW191
     2       1H0, 5HCL = , F10.5, 5X5HCD = , F10.5, 5X5HCM = , F10.5)   FLOW192
      CMT = CMT + CML                                                   FLOW193
C                                                                       FLOW194
C  CALCULATION OF FORCES & MOMENTS USING TRAPEZOIDAL INTEGRATION        FLOW195
C  COMPLETE.                                                            FLOW196
C  NOW CALCULATE FORCES & MOMENTS USING PARABOLIC INTEGRATION.          FLOW197
C  SPECIAL HANDLING FOR FIRST AND LAST ELEMENTS                         FLOW198
C                                                                       FLOW199
C  NOTE.  DEFINITIONS FOR CD & CF INCLUDE MULTIPLICATION BY DL,         FLOW200
C         FOR  CG & CI INCLUDE MULTIPLICATION BY DL**2.                 FLOW201
C                                                                       FLOW202
      CX = 0.0                                                          FLOW203
      CY = 0.0                                                          FLOW204
      CML = 0.0                                                         FLOW205
      NF1 = NF - 1                                                      FLOW206
C  FIRST ELEMENT                                                        FLOW207
      J2 = NI + 1                                                       FLOW208
      J1 = J2 + 1                                                       FLOW209
      J3 = J2 + 2                                                       FLOW210
C                                                                       FLOW211
  151 CONTINUE                                                          FLOW212
      CP1 = CD(J2)*CPJ(J1) - (CF(J2)+CD(J2))*CPJ(J2) + CF(J2)*CPJ(J3)   FLOW213
      CP2 = CG(J2)*CPJ(J1) - (CG(J2)+CI(J2))*CPJ(J2) + CI(J2)*CPJ(J3)   FLOW214
      DLSQ = DL(J2)**2                                                  FLOW215
      CXX = CURV(J2)*CP1*DLSQ/6.0                                       FLOW216
      CYY = -DL(J2)*(CPJ(J2) + CP2 /12.0)                               FLOW217
      CMO = -CP1*DLSQ*(10.0 + 3.0*(CURV(J2)*DL(J2))**2)/120.0           FLOW218
C                                                                       FLOW219
```

```
      DCX =      CXX*CA(J2) - CYY*SA(J2)
      CX = CX + DCX                                              FLOW220
      DCY =      CXX*SA(J2) + CYY*CA(J2)                         FLOW221
      CY = CY + DCY                                              FLOW222
      CML = CML -DCX*(YM-Y(J2)) -DCY*(X(J2)-XM) - CMO            FLOW223
C                                                                FLOW224
C                                                                FLOW225
      IF (J2 - NF1) 152,153,154                                  FLOW226
C                                                                FLOW227
  152 J1 = J2                                                    FLOW228
      J2 = J2 + 1                                                FLOW229
      J3 = J2 + 1                                                FLOW230
      GO TO 151                                                  FLOW231
C                                                                FLOW232
C  LAST ELEMENT                                                  FLOW233
  153 J2 = NF                                                    FLOW234
      J1 = NF - 1                                                FLOW235
      J3 = NF - 2                                                FLOW236
      GO TO 151                                                  FLOW237
C                                                                FLOW238
C                                                                FLOW239
  154 CLI = CY*COSA - CX*SINA                                    FLOW240
      CDI = CY*SINA + CX*COSA                                    FLOW241
      CMIP = CMIP + CML                                          FLOW242
C                                                                FLOW243
      WRITE(6,205)                                               FLOW244
  205 FORMAT(1H0,25HPARABOLIC INTEGRATION     )                  FLOW245
C                                                                FLOW246
      WRITE(6,200) CY, CX, CLI, CDI, CML                         FLOW247
C                                                                FLOW248
  160 CONTINUE                                                   FLOW249
  161 REWIND 7                                                   FLOW250
      WRITE(7,1500) (X(J),J=1,N)
      WRITE(7,1500) (Y(J),J=1,N)
      WRITE(7,1500) (V1(J),J=1,N)
      WRITE(7,1500) (V2(J),J=1,N)
      WRITE(7,1500) (V3(J),J=1,N)
      WRITE(7,1500) (V4(J),J=1,N)
      WRITE(7,1500) (V5(J),J=1,N)
      IF(IFLLL.NE.0) WRITE(6,210)  CMT
  210 FORMAT(1H0,11HTOTAL CM = , F10.5)                          FLOW251
 1500 FORMAT(0P6E13.6)                                           FLOW252
      IF(IFLLL.NE.0) WRITE(6,215) CMTP
  215 FORMAT(1H0,11HTOTAL CM = , F10.5, 5X11H(PARABOLIC))        FLOW253
C                                                                FLOW254
      RETURN                                                     FLOW255
      END                                                        FLOW256
                                                                 FLOW257
```

```
      SUBROUTINE  GEOMCF (INL, IFL, DS, IFST, ISEC, DL)
C
C  CALCULATES REQUIRED GEOMETRY COMBINATION COEFFICIENTS
C  ASSOCIATED WITH PARABOLIC DISTRIBUTIONS.
C
C  IFST = 1 OR 3, NEED  CD, CE, AND CF
C
C  ISEC = 1 OR 3, NEED  CG, CH, AND CI
C
C  (HOWEVER, ALL ARE CALCULATED FOR NOW)
C
C
C  SPECIAL HANDLING FOR FIRST AND LAST ELEMENTS.
C  (FOR NOW BODY SLOPE ASSUMED CONTINUOUS)
C
C
      DIMENSION DS(1), DL(1)
C
      COMMON/GCOEFS/ CD(500), CF(500), CG(500), CI(500)
C
C
      I1 = IFL
      I2 = INL
C
   10 I3 = I2 + 1
   20 C1 = DS(I1) + DS(I2)
      C2 = (DS(I2) + DS(I2))*DL(I2)/DS(I2)
      C3 = DS(I2) + DS(I3)
      C4 = C2/(C1 + C3)
      C5 = C4/C1
      C6 = C4/C3
C
      CD(I2) = -C5*C3
      CF(I2) =  C6*C1
      CG(I2) =  C5*C2
      CI(I2) =  C6*C2
C
      I1 = I2
      I2 = I1 + 1
      IF (I2 - IFL)10,30,40
C
   30 I3 = INL
      GO TO 20
C
   40 CONTINUE
C
C  REDO FIRST AND LAST POINTS FOR ONE-SIDED FITS
      I1 = INL
      I2 = I1 + 1
      I3 = I1 + 2
      C1 = DS(I1) + DS(I2)
      C3 = DS(I3) + DS(I2)
      C2 = 2.*DS(I1)*DL(I1)/DS(I1)
      CD(I1) = C2*(1./C1 + 1./C3)
      CF(I1) = C2*(1./(C1+C3) - 1./C3)
      C2 = (C2/(2.*DS(I2)*DL(I2)/DS(I2)))**2
```

```
      CG(I1) = -(CG(I2) + CI(I2))*C2                                    GEOM057
      CI(I1) = CI(I2)*C2                                                GEOM058
C                                                                       GEOM059
      I3 = IFL                                                          GEOM060
      I2 = I3 - 1                                                       GEOM061
      I1 = I3 - 2                                                       GEOM062
      C1 = DS(I3) + DS(I2)                                              GEOM063
      C3 = DS(I2) + DS(I1)                                              GEOM064
      C2 = 2.*DS(I3)*DL(I3)/DS(I3)                                      GEOM065
      CD(I3) = -C2*(1./C1 + 1./C3)                                      GEOM066
      CF(I3) = C2*(1./C3 - 1./(C1 + C3))                                GEOM067
      C2 = (C2/(2.*DS(I2)*DL(I2)/DS(I2)))**2                            GEOM068
      CG(I3) = -(CG(I2) + CI(I2))*C2                                    GEOM069
      CI(I3) = CG(I2)*C2                                                GEOM070
C                                                                       GEOM071
C                                                                       GEOM072
      RETURN                                                            GEOM073
      END                                                               GEOM074
```

```
      SUBROUTINE GETT(IU, IT, N1, A1, N2, A2)                           GETT001
C                                                                       GETT002
      DIMENSION  A1(N1), A2(N2)                                         GETT003
C                                                                       GETT004
      GO TO (10,20,30,40), IT                                           GETT005
C                                                                       GETT006
C  READ A1                                                              GETT007
   10 READ(IU) A1                                                       GETT008
      RETURN                                                            GETT009
C                                                                       GETT010
C  READ N1 AND A1                                                       GETT011
   20 READ(IU) N1, A1                                                   GETT012
      RETURN                                                            GETT013
C                                                                       GETT014
C  READ A1 AND A2                                                       GETT015
   30 READ(IU) A1, A2                                                   GETT016
      RETURN                                                            GETT017
C                                                                       GETT018
C  READ IDUM AND A1                                                     GETT019
   40 READ(IU) IDUM, A1                                                 GETT020
      RETURN                                                            GETT021
      END                                                               GETT022
```

```
      SUBROUTINE    MAFORM     ( M, NONU, NBNU, ISOL, IPRINT)              MAF0001
C                                                                          MAF0002
C                                                                          MAF0003
C  THIS ROUTINE FORMS AND STORES MATRICES A AND B                          MAF0004
C  ALSO CALCULATES AND STORES ALPHA AND CIRCULATORY ONSET VELOCITIES       MAF0005
C                                                                          MAF0006
C  THIS IS FIRST ATTEMPT AND IS SIMPLE ONE-TIME PASS.                      MAF0007
C  CAPABILITY TO ONLY CHANGE SELECTED ARRAY ITEMS                          MAF0008
C  WILL BE ADDED LATER.                                                    MAF0009
C                                                                          MAF0010
C                                                                          MAF0011
      REAL*4    N                                                          MAF0012
C                                                                          MAF0013
C                                                                          MAF0014
      DIMENSION     A(500),   B(500),   XO(500),  YO(500),                 MAF0015
     1              DS(500),  SA(500),   CA(500),  BLU(500),               MAF0016
     2              N(12),     T(12)                                       MAF0017
C                                                                          MAF0018
      COMMON /COMBOD/CCL, INCLT, CLT, ALPHA, SUMDS(10), TLU(10,12),IND     MAF0019
     1              , ALPHAO, CNU(10), SMDSWF(10), MIO(10)                 MAF0020
C                                                                          MAF0021
      COMMON /BFLAG/ IDB(10), INL(10), IFL(10), ML(10), LIFT(10),          MAF0022
     1              IBMF(10), ISAV1(10), ISAV2(10), ISAV3(10),             MAF0023
     2              RTITLE(10,  7), IBT, IBST, IBTOT, NELTOT,              MAF0024
     3              ITRB(10), INMB(10), CHORDB(10), IBD(10), LIFTOT        MAF0025
     4              ,IPRB(10), IFST(10), ISEC(10), FTITLE(15), IPVR(10)    MAF0026
      COMMON /FILEID/  IF01, IF02, IF03, IF04, IF05,                       MAF0027
     1                    IF06, IF07, IF08, IF09, IF10,                    MAF0028
     2                 IF11, IF12, IF13, IF14, IF15                        MAF0029
     3                 ,IF16, IF17, IF18, IF19, IF20                       MAF0030
      COMMON/ROTAT/NROT, ROTRAD(10)                                        MAF0031
C                                                                          MAF0032
      COMMON/ELDATA/ XO, YO, DS, SA, CA, CURV(500), DL(500)                MAF0033
      COMMON/GCOEFS/ CD(500), CF(500), CG(500), CI(500), WF(500)           MAF0034
C                                                                          MAF0035
C                                                                          MAF0036
C  BEGIN NEW CASE. II IS ROW COUNT, JJ IS COLUMN COUNT.                    MAF0037
      REWIND IF09                                                          MAF0038
      REWIND IF10                                                          MAF0039
      REWIND IF11                                                          MAF0040
      REWIND IF12                                                          MAF0041
      REWIND IF13                                                          MAF0042
      M = LIFTOT + 2                                                       MAF0043
      I2 = 0                                                               MAF0044
      II = 0                                                               MAF0045
      LB = 0                                                               MAF0046
      INT = 0                                                              MAF0047
      NROT = 0                                                             MAF0048
C  INPUT VALUE OF NONU .GT. 6 IS USED TO FLAG                              MAF0049
C  A ROTATING NON-UNIFORM FLOW                                             MAF0050
      IF (NONU .LE. 6)  GO TO 5                                            MAF0051
      NROT = 1                                                             MAF0052
      NONU = 1                                                             MAF0053
    5 CONTINUE                                                             MAF0054
      DO 210 IIB = 1,IBTOT                                                 MAF0055
C  CHECK IF BODY IIB IS STILL IN DATA SET.                                 MAF0056
```

```
      IF (IBMF(IIB) .LT. 0)  GO TO 210
      INT = INT + 1
C
C
      IO = INL(IIB)
      IF = IO + NL(IIB)- 1
      NIO(IIB) = II + 1
      DO 180 I = IO,IF
      JJ = 0
      K = 2
      II = II + 1
      IF (IPRINT .EQ. 2)
     1WRITE (6,10) II
   10 FORMAT(1H1, T4,'I = ',I3/1H0, T4,'J', T15,'A0', T31,'A1',
     1   T47,'AC', T63,'AP', T79,'B0',T95,'B1',T111,'BC',T127,'BP')
C
      DO 150 IB = 1,IBTOT
      IF (IBMF(IB) .LT. 0)  GO TO 150
C
C  COUNTER FOR ELEMENT GEOMETRY
      J = INL(IB) - 1
C  COUNTERS FOR A,B ARRAYS
      JI = JJ + 1
      JF = JJ + NL(IB)
C
C  JJ IS COUNTER FOR THE CURRENT ELEMENT
C
C  ZERO OUT A&B ARRAYS
      DO 20 JJ1 = JI,JF
      A(JJ1) = 0.0
   20 B(JJ1) = 0.0
C
      JJ = JI
      JJ1 = JJ + 1
      JJ3 = JJ1 + 1
      GO TO 40
   30 JJ3 = JJ + 1
   40 J = J + 1
      A0     =          ABFORM(XO(I),YO(I),DL(I),SA(I),CA(I),IFST(IB),
     1                         XO(J),YO(J),DL(J),SA(J),CA(J),ISEC(IB),
     2                         A1, AC, AP, B0, B1, BC, BP)
      B(JJ ) = B(JJ ) + B0
      A(JJ) = A(JJ) + A0
C
      IF (IPRINT .EQ. 2)
     1 WRITE (6,50) JJ, A0, A1, AC, AP, B0, B1, BC, BP
   50 FORMAT(1H , I3, 8(4XF12.6))
C
      IF (IFST(IB) .EQ. 0)  GO TO 80
      IF (IFST(IB) - 2)60,70,60
C
C  FIRST TERMS OF FIRST ORDER
   60 A(JJ ) = A(JJ ) -        (CF(J) + CD(J))*A1
      B(JJ ) = B(JJ ) -        (CF(J) + CD(J))*B1
      A(JJ1) = A(JJ1) +        CD(J)*A1
      B(JJ1) = B(JJ1) +        CD(J)*B1
      A(JJ3) = A(JJ3) +        CF(J)*A1
```

```
      B(JJ3) = B(JJ3) +       CF(J)*B1                                  MAF0114
C                                                                       MAF0115
   70 IF (IFST(IB) .LT. 2)  GO TO 80                                    MAF0116
C                                                                       MAF0117
C  CURVATURE TERM OF FIRST ORDER                                        MAF0118
      A(JJ ) = A(JJ ) + DL(J)*CURV(J)*AC                                MAF0119
      B(JJ ) = B(JJ ) + DL(J)*CURV(J)*BC                                MAF0120
C                                                                       MAF0121
   80 IF (ISEC(IB) .EQ. 0)  GO TO 110                                   MAF0122
      IF (ISEC(IB) - 2)90,100,90                                        MAF0123
C                                                                       MAF0124
C  FIRST TERMS OF SECOND ORDER                                          MAF0125
   90 A(JJ ) = A(JJ ) - (CG(J) + CI(J))*AP                              MAF0126
      B(JJ ) = B(JJ ) - (CG(J) + CI(J))*BP                              MAF0127
      A(JJ1) = A(JJ1) + CG(J)*AP                                        MAF0128
      B(JJ1) = B(JJ1) + CG(J)*BP                                        MAF0129
      A(JJ3) = A(JJ3) + CI(J)*AP                                        MAF0130
      B(JJ3) = B(JJ3) + CI(J)*BP                                        MAF0131
C                                                                       MAF0132
  100 IF (ISEC(IB) .LT. 2)  GO TO 110                                   MAF0133
C                                                                       MAF0134
C  CURVATURE TERM OF SECOND ORDER                                       MAF0135
      A(JJ ) = A(JJ ) + 2.0*AP*(DL(J)*CURV(J))**2                       MAF0136
      B(JJ ) = B(JJ ) + 2.0*BP*(DL(J)*CURV(J))**2                       MAF0137
C                                                                       MAF0138
  110 JJ1 = JJ                                                          MAF0139
      JJ = JJ + 1                                                       MAF0140
      IF (JJ - JF)30,120,130                                            MAF0141
C                                                                       MAF0142
  120 JJ3 = JJ1 - 1                                                     MAF0143
      GO TO 40                                                          MAF0144
  130 JJ = JJ - 1                                                       MAF0145
C                                                                       MAF0146
C                                                                       MAF0147
C  IF LIFTING BODY, SAVE FIRST AND LAST B. ALSO CALCULATE N AND T.      MAF0148
      IF (LIFT(IB) .EQ. 0)  GO TO 150                                   MAF0149
      VN = 0.0                                                          MAF0150
      VT = 0.0                                                          MAF0151
      SMDSWF(IB) = 0.0                                                  MAF0152
      IF ((IWT .EQ. 1) .AND. (I .EQ. IC))                               MAF0153
     1CALL WEIGHT(SUMDS(IB), DS, JI, JF, WF, IPVR(IB))                  MAF0154
      DO 140 J = JI,JF                                                  MAF0155
      SMDSWF(IB) = SMDSWF(IB) + DS(J) * WF(J)                           MAF0156
      VN = VN + B(J)*WF(J)                                              MAF0157
  140 VT = VT + A(J)*WF(J)                                              MAF0158
C                                                                       MAF0159
      K = K + 1                                                         MAF0160
      N(K) = VN                                                         MAF0161
      T(K) = VT                                                         MAF0162
C                                                                       MAF0163
C                                                                       MAF0164
  150 CONTINUE                                                          MAF0165
C                                                                       MAF0166
C  A COMPLETE ROW OF BOTH A AND B HAS BEEN GENERATED.                   MAF0167
C  SAVE A ON UNIT 9, A AND B ON UNIT 10.                                MAF0168
      CALL SAVE(IF10, 1, 1, JJ, A, 1, VN)                               MAF0169
      CALL SAVE(IF10, 1, 1, JJ, B, 1, VN)                               MAF0170
```

```
C                                                              MAF0171
C                                                              MAF0172
C  SET ALPHA ONSET FLOWS                                       MAF0173
C  ALPHA = 0                                                   MAF0174
      N(1) = SA(I)                                             MAF0175
      T(1) = CA(I)                                             MAF0176
C                                                              MAF0177
C  ALPHA = 90                                                  MAF0178
      N(2) =-CA(I)                                             MAF0179
      T(2) = SA(I)                                             MAF0180
C                                                              MAF0181
C                                                              MAF0182
      DO 160 K = 1,M                                           MAF0183
      JJ = JJ + 1                                              MAF0184
  160 A(JJ) = N(K)                                             MAF0185
C                                                              MAF0186
C  THE A-ARRAY IS SAVED ON IF09 IN AUGMENTED FORM              MAF0187
      CALL SAVE(IF09, 1, 1, JJ, A, 1, VN)                      MAF0188
C                                                              MAF0189
C                                                              MAF0190
C  ALSO SET UP TLU ARRAY                                       MAF0191
      IF (LIFT(IIB) .EQ. 0)  GO TO 203                         MAF0192
      IF (I .NE. IO)  GO TO 201                                MAF0193
      LB = LB + 1                                              MAF0194
      DO 200 K = 1,M                                           MAF0195
  200 TLU(LB,K) = T(K)                                         MAF0196
C                                                              MAF0197
  201 IF (I .NE. IF)  GO TO 203                                MAF0198
      DO 202 K = 1,M                                           MAF0199
  202 TLU(LB,K) = TLU(LB,K) + T(K)                             MAF0200
  203 CONTINUE                                                 MAF0201
C                                                              MAF0202
C                                                              MAF0203
C  SAVE NORMAL AND TANGENTIAL ONSET VELOCITIES.                MAF0204
C  NORMALS ON UNIT 11,  TANGENTIALS ON UNIT 12                 MAF0205
C                                                              MAF0206
      WRITE(IF11)  N                                           MAF0207
      WRITE(IF12)  T                                           MAF0208
C                                                              MAF0209
C  IF I=IO AND LIFTING BODY, SET BLU                           MAF0210
      IF (I .NE. IO)  GO TO 180                                MAF0211
      IF (LIFT(IIB) .EQ. 0)  GO TO 180                         MAF0212
      DO 170 J = 1, NELTCT                                     MAF0213
  170 BLU(J) = B(J)                                            MAF0214
C                                                              MAF0215
C                                                              MAF0216
  180 CONTINUE                                                 MAF0217
C  IF LIFTING BODY, SUM BLU                                    MAF0218
      IF (LIFT(IIB) .EQ. 0)  GO TO 210                         MAF0219
      DO 190 J = 1, NELTOT                                     MAF0220
  190 BLU(J) = BLU(J) + B(J)                                   MAF0221
C                                                              MAF0222
C                                                              MAF0223
C  STORE BLU ON UNIT 13                                        MAF0224
      CALL SAVE(IF13, 1, 1, NELTOT, BLU, 1, VN)                MAF0225
C                                                              MAF0226
  210 CONTINUF                                                 MAF0227
```

```
      C                                                                 MAF0228
      C                                                                 MAF0229
      C                                                                 MAF0230
      C                                                                 MAF0231
  240 CONTINUE                                                          MAF0232
      IF (IPRINT .NE. 2)  GO TO 340                                     MAF0233
      WRITE (6,250)                                                     MAF0234
  250 FORMAT(1H1)                                                       MAF0235
      DO 290 IB = 1,IBTOT                                               MAF0236
      IF (IBMF(IB) .LT. 0)  GO TO 290                                   MAF0237
      J1 = INL(IB)                                                      MAF0238
      J2 = IFL(IB)                                                      MAF0239
      WRITE (6,260) IDB(IB)                                             MAF0240
  260 FORMAT(1H0,'GEOMETRY COMBINATION COEFFICIENTS FOR BODY  ID = ',I2/MAF0241
     1       1H0, T4,'J', T15,'CD',T31,'CF',T47,'CG',T63,'CI')          MAF0242
      C                                                                 MAF0243
      DO 270 J = J1,J2                                                  MAF0244
  270 WRITE (6,280) J, CD(J), CF(J), CG(J), CI(J)                       MAF0245
  280 FORMAT(1H , I3, 4(4XF12.6))                                       MAF0246
  290 CONTINUE                                                          MAF0247
      C                                                                 MAF0248
      WRITE (6,250)                                                     MAF0249
      REWIND IF10                                                       MAF0250
      JF = NELTOT                                                       MAF0251
      DO 330 J = 1,JF                                                   MAF0252
      CALL GETT(IF10, 1,    JF, A, 1, VN)                               MAF0253
      CALL GETT(IF10, 1,    JF, B, 1, VN)                               MAF0254
      WRITE (6,300)  J                                                  MAF0255
  300 FORMAT(1H0, T4,'A(', I3,', J)'/)                                  MAF0256
      WRITE (6,310) (A(I),I=1,JF)                                       MAF0257
  310 FORMAT(8(4XF12.6))                                                MAF0258
      C                                                                 MAF0259
      WRITE (6,320)  J                                                  MAF0260
  320 FORMAT(1H0, T4,'B(', I3,', J)'/)                                  MAF0261
      WRITE (6,310) (B(I),I=1,JF)                                       MAF0262
      C                                                                 MAF0263
  330 CONTINUE                                                          MAF0264
      C                                                                 MAF0265
      C                                                                 MAF0266
  340 CONTINUE                                                          MAF0267
      C                                                                 MAF0268
      C                                                                 MAF0269
      C                                                                 MAF0270
      RETURN                                                            MAF0271
      END                                                               MAF0272
```

```
      SUBROUTINE MAIN1                                                  MAN1001
C                                                                       MAN1002
      DIMENSION XO(500),YO(500),DS(500),SA(500),CA(500),                MAN1003
     1          ELGC(3500), ELGD(2500)                                  MAN1004
C                                                                       MAN1005
      COMMON /BFLAG/ IDR(10), INL(10), IFL(10), NL(10), LIFT(10),       MAN1006
     1               IBMF(10), ISAV1(10), ISAV2(10), ISAV3(10),         MAN1007
     2               BTITLE(10, 7), IBT, IBST, IBTOT, NELTOT,           MAN1008
     3               ITRB(10), INNB(10), CHORDB(10), IBD(10), LIFTOT    MAN1009
     4               ,IPRB(10), IFST(10), ISEC(10), FTITLE(15), IPVR(10) MAN1010
C                                                                       MAN1011
C                                                                       MAN1012
C                                                                       MAN1013
      COMMON /COMBOD/CCL, INCLT, CLT, ALPHA, SUMDS(10), TLU(10,12),IND  MAN1014
     1               , ALPHAD, CNU(10), SMDSNF(10), NI0(10)             MAN1015
C                                                                       MAN1016
      COMMON /FILEID/ IFILE1, IFILE2, IFILE3, IFILE4, IFILE5,           MAN1017
     1                IFILE6, IFILE7, IFILE8, IFILE9, IFIL10,           MAN1018
     2                IFIL11, IFIL12, IFIL13, IFIL14, IFIL15            MAN1019
     3               ,IFIL16, IFIL17, IFIL18, IFIL19, IFIL20            MAN1020
      COMMON /MDATA/ ISOL,IOFF,MONU,NBNU,IPRINT,MORE,M                  MAN1021
      COMMON /ELDATA/ XO, YO, DS, SA, CA, CURV(500), DL(500)            MAN1022
      COMMON /GCOEFS/ CD(500), CF(500), CG(500), CI(500), WF(500)       MAN1023
      COMMON/CON/ IFLLL
      EQUIVALENCE  (ELGC(1), XO(1)), (ELGD(1), CD(1))                   MAN1024
C                                                                       MAN1025
C  FORM ELEMENTS                                                        MAN1026
      CALL TIMEV(T)                                                     MAN1027
      WRITE(6,20) T                                                     MAN1028
   20 FORMAT (1H0, 'CALL ELFORM, T = ', F9.3, 'SECONDS.')               MAN1029
      CALL ELFORM (SUMDS)                                               MAN1030
C                                                                       MAN1031
C  READ IN FLOW TITLE CARD                                              MAN1032
      ITYP = 8                                                          MAN1033
      READ (5,30) FTITLE,          ITYPE                                MAN1034
   30 FORMAT (15A4,          11XI1)                                     MAN1035
      IF (ITYPE .NE. ITYP)  CALL TYPE(ITYP, ITYPE)                      MAN1036
C                                                                       MAN1037
C  READ IN FLOW CONTROL CARD                                            MAN1038
      ITYP = 9                                                          MAN1039
      READ (5,40) INCLT, CLT, ICHORD, CCL, IND, ISOL, IOFF, MONU,       MAN1040
     1       NBNU, IPRINT, MORE, IFLLL,ITYPE                            MAN1041
   40 FORMAT (I1, 4XF10.5, 2XI1, 2XF10.5, 5(4XI1),  9XI1, 4XI1, I1,I1)  MAN1042
      IF (ITYPE .NE. ITYP)  CALL TYPE(ITYP, ITYPE)                      MAN1043
      IF (MONU .NE. 0) ISOL = 1                                         MAN1044
      IF (INCLT .EQ. 0)  ALPHA = CLT                                    MAN1045
      IF (ICHORD .EQ. 0)  CCL = 1.0                                     MAN1046
C                                                                       MAN1047
C  FORM MATRICES                                                        MAN1048
      CALL TIMEV(T)                                                     MAN1049
      WRITE(6,50) T                                                     MAN1050
   50 FORMAT (1H0, 'ELFORM COMPLETE, CALL MAFORM, T = ',F9.3,'SECONDS,')MAN1051
      CALL  MAFORM     ( M, MONU, NBNU, ISOL, IPRINT)                   MAN1052
C                                                                       MAN1053
      CALL TIMEV(T)                                                     MAN1054
      WRITE (6,60) T                                                    MAN1055
```

154

```
   60 FORMAT (1H0, 'MAFORM COMPLETE, CALL SOLVE, T = ',F9.3,'SECONDS.')  MAN1056
C                                                                       MAN1057
      CALL  ASSEMR                                                      MAN1058
C                                                                       MAN1059
C  SAVE ELDATA AND GCOEFS DATA                                          MAN1060
      REWIND  IFIL16                                                    MAN1061
      WRITE(IFIL16) ELGC, ELGD                                          MAN1062
C                                                                       MAN1063
      RETURN                                                            MAN1064
      END                                                               MAN1065
```

```
      SUBROUTINE  MAIN3                                                 MAN3001
C                                                                       MAN3002
      COMMON /BFLAG/ IDB(10), INL(10), IFL(10), NL(10), LIFT(10),       MAN3003
     1              IBMF(10), ISAV1(10), ISAV2(10), ISAV3(10),          MAN3004
     2              BTITLE(10, 7), IBT, IBST, IBTOT, NELTOT,            MAN3005
     3              ITRB(10), INMB(10), CHORDB(10), IBD(10), LIFTOT     MAN3006
     4              ,IPRB(10), IFST(10), ISEC(10), FTITLE(15), IPVR(10) MAN3007
C                                                                       MAN3008
C                                                                       MAN3009
C                                                                       MAN3010
      COMMON /COMBOD/CCL, INCLT, CLT, ALPHA, SUMDS(10), ILU(10,12),IND  MAN3011
     1             , ALPHAO, CNU(10), SMDSWF(10), MIO(10)               MAN3012
C                                                                       MAN3013
      COMMON /FILEID/ IFILE1, IFILE2, IFILE3, IFILE4, IFILE5,           MAN3014
     1                IFILE6, IFILE7, IFILE8, IFILE9, IFIL10,           MAN3015
     2                IFIL11, IFIL12, IFIL13, IFIL14, IFIL15            MAN3016
     3                ,IFIL16, IFIL17, IFIL18, IFIL19, IFIL20           MAN3017
      COMMON /MDATA/ ISOL,IOFF,NONU,NBNU,IPRINT,MORE,M                  MAN3018
      COMMON /COM/ IFLLL
C                                                                       MAN3019
C  CALCULATE COMBINATION CONSTANTS IF IFLLL= 1                          MAN3020
      IF(IFLLL.EQ.0) GO TO 81
      CALL  COMBO (NELTOT, LIFTOT, M, NONU)                             MAN3021
C                                                                       MAN3022
      CALL TIMEV(T)                                                     MAN3023
      WRITE(6,80) T                                                     MAN3024
   80 FORMAT (1H0, 'COMBO COMPLETE, CALL FLOWS, T = ',F9.3,'SECONDS.')  MAN3025
C                                                                       MAN3026
   81 CALL FLOWS(NELTOT, M, IPRINT)                                     MAN3027
      CALL TIMEV(T)                                                     MAN3028
      WRITE(6,90) T                                                     MAN3029
   90 FORMAT(1H0, 'FLOWS COMPLETE, T = ', F9.3, 'SECONDS.')             MAN3030
C                                                                       MAN3031
C  CHECK FOR OFFBODY POINTS                                             MAN3032
      IF (IOFF .NE. 1)  GO TO 110                                       MAN3033
C                                                                       MAN3034
      CALL  OFFBOD(NELTOT, M, CHORDB, IDB, IBTOT)                       MAN3035
      CALL TIMEV(T)                                                     MAN3036
      WRITE(6,100) T                                                    MAN3037
  100 FORMAT(1H0, 'OFFBODY POINTS COMPLETE, T = ', F9.3,' SECONDS.')    MAN3038
C                                                                       MAN3039
C                                                                       MAN3040
  110 CONTINUE                                                          MAN3041
C                                                                       MAN3042
C                                                                       MAN3043
      RETURN                                                            MAN3044
      END                                                               MAN3045
```

```
      SUBROUTINE MIS1 ( A, N, ND, B, MD, NERR, D )                      MIS1001
C                                                                       MIS1002
C     A REAL*8     SUBROUTINE TO                                        MIS1003
C     INVERT A MATRIX AND/OR SOLVE SIMULTANEOUS EQUATIONS               MIS1004
C                                                                       MIS1005
C     INPUT.                                                            MIS1006
C     A = NAME OF INPUT MATRIX, DIMENSIONED A(ND,ND)                    MIS1007
C     N = IS THE ORDER OF THE MATRIX A                                  MIS1008
C     ND = IS THE DIMENSION OF THE SQUARE ARRAY A                       MIS1009
C     B = NAME OF INPUT MATRIX, DIMENSIONED B(ND,MD)                    MIS1010
C         ( B(ND) IS ALLOWED IF MD=1 )                                  MIS1011
C     MD = IS THE NUMBER OF COLUMNS IN THE RECTANGULAR ARRAY B          MIS1012
C     D = SCALE FACTOR FOR VALUE OF DETERMINANT  (=1., FOR NO SCALING)  MIS1013
C                                                                       MIS1014
C     OUTPUT.                                                           MIS1015
C     A(I,J) = GARBAGE                                                  MIS1016
C     B(I,J) = A-INVERSE * B                                            MIS1017
C     NERR  =  0--OK,  1--A IS SINGULAR                                 MIS1018
C     D = SCALED VALUE OF DETERMINANT                                   MIS1019
C                                                                       MIS1020
      REAL*8     A,B,D,AIJMAX,ARAT                                      MIS1021
      DIMENSION A(ND,ND), B(ND,MD)                                      MIS1022
      EQUIVALENCE  (L,FL), (K,FK)                                       MIS1023
      DATA  EPS/1.E-25/                                                 MIS1024
C     START REDUCTION OF MATRIX A                                       MIS1025
C                                                                       MIS1026
      DO 80  I=1,N                                                      MIS1027
C                                                                       MIS1028
C     SEARCH FOR MAXIMUM ELEMENT IN ITH ROW OF A-MATRIX                 MIS1029
C                                                                       MIS1030
      AIJMAX = A(I,1)                                                   MIS1031
      JMAX = 1                                                          MIS1032
      DO 10  J=2,N                                                      MIS1033
      IF (  ABS(A(I,J)) .LE.   ABS(AIJMAX)) GO TO 10                    MIS1034
      AIJMAX = A(I,J)                                                   MIS1035
      JMAX = J                                                          MIS1036
   10 CONTINUE                                                          MIS1037
C                                                                       MIS1038
C     IF AIJMAX IS ZERO, THE MATRIX IS SINGULAR                         MIS1039
C                                                                       MIS1040
      IF (ABS(AIJMAX) .GT. EPS)  GO TO 20                               MIS1041
      D =  0.0E0                                                        MIS1042
      NERR =  1                                                         MIS1043
      RETURN                                                            MIS1044
C                                                                       MIS1045
C     NORMALIZE ITH ROW BY AIJMAX (JMAX ELEMENT OF ITH ROW)             MIS1046
C                                                                       MIS1047
   20 DO 30  J=1,N                                                      MIS1048
   30 A(I,J) = A(I,J) / AIJMAX                                          MIS1049
      D = D*AIJMAX                                                      MIS1050
C                                                                       MIS1051
C     NORMALIZE ITH ROW OF B                                            MIS1052
C                                                                       MIS1053
      DO 40  J=1,MD                                                     MIS1054
   40 B(I,J) = B(I,J) / AIJMAX                                          MIS1055
C                                                                       MIS1056
```

```
C     USE ROW TRANSFORMATIONS TO GET ZEROS ABOVE AND BELOW THE JMAX
C     ELEMENT OF THE ITH ROW OF A.  APPLY SAME ROW TRANSFORMATIONS
C     TO THE B MATRIX.
C
      DO 70  K=1,N
      IF (K .EQ. I) GO TO 70
      ARAT = -A(K,JMAX)
      DO 50  J=1,N
      IF (ABS(A(I,J)) .LT. EPS)  GO TO 50
      A(K,J) = ARAT * A(I,J) + A(K,J)
   50 CONTINUE
      A(K,JMAX) =  0.0E0
      DO 60  J=1,MD
      IF (ABS(B(I,J)) .LT. EPS)  GO TO 60
      B(K,J) = ARAT * B(I,J) + B(K,J)
   60 CONTINUE
   70 CONTINUE
C
C     STORE ROW COUNTER (I) IN TOP ELEMENT OF JMAX COLUMN. THUS,
C     THE TOP ROW OF A WILL CONTAIN THE LOC OF THE PIVOT (UNITY)
C     ELEMENT OF EACH COLUMN (AFTER REDUCTION).
C
      L = I
   80 A(1,JMAX) = FL
C                   THIS STORES INTEGER I IN TOP ROW OF A
C
C     THE REDUCTION OF A IS NOW COMPLETE.  PERFORM ROW INTERCHANGES
C     AS INDICATED IN THE FIRST ROW OF A.
C
      DO 120  I=1,N
      K = I
   90 FK = A(1,K)
C                   THIS PUTS THE INTEGER VALUE IN A INTO K
      IF (K-I)90,120,100
C
C     IF K(1,I) IS LESS THAN I, THEN THAT ROW HAS ALREADY BEEN
C     INVOLVED IN AN INTERCHANGE, AND WE USE K(1,K) UNTIL WE GET
C     A VALUE OF K GREATER THAN I  (CORRESPONDING TO A ROW STORED
C     BELOW THE ITH ROW.  (CLEAR AS MUC)
C
  100 DO 110  J=1,MD
      ARAT =  B(I,J)
      B(I,J) = B(K,J)
  110 B(K,J) = ARAT
      D = -D
  120 CONTINUE
      NERR = 0
      RETURN
      END
```

```
      SUBROUTINE MIS2 ( A, N, ND, B, MD, MX, NERR, D)                   MIS2001
C                                                                       MIS2002
C     A REAL*8     SUBROUTINE TO                                        MIS2003
C     INVERT A MATRIX AND/OR SOLVE SIMULTANEOUS EQUATIONS               MIS2004
C                                                                       MIS2005
C     INPUT,                                                            MIS2006
C     A = NAME OF INPUT MATRIX, DIMENSIONED A(ND,ND)                    MIS2007
C     N = IS THE ORDER OF THE MATRIX A                                  MIS2008
C     ND = IS THE DIMENSION OF THE SQUARE ARRAY A                       MIS2009
C     B = NAME OF INPUT MATRIX, DIMENSIONED B(ND,MD)                    MIS2010
C         ( B(ND) IS ALLOWED IF MD=1 )                                  MIS2011
C     MD = IS THE NUMBER OF COLUMNS IN THE RECTANGULAR ARRAY B          MIS2012
C     D = SCALE FACTOR FOR VALUE OF DETERMINANT  (=1., FOR NO SCALING)  MIS2013
C                                                                       MIS2014
C     OUTPUT,                                                           MIS2015
C     A(I,J) = GARBAGE                                                  MIS2016
C     B(I,J) = A-INVERSE * B                                            MIS2017
C     NERR  =  0--OK,  1--A IS SINGULAR                                 MIS2018
C     D = SCALED VALUE OF DETERMINANT                                   MIS2019
C                                                                       MIS2020
      REAL*8      A,B,D,AIJMAX,ARAT                                     MIS2021
      DIMENSION A(ND,ND), B(ND,MX)                                      MIS2022
      EQUIVALENCE  (L,FL), (K,FK)                                       MIS2023
      DATA  EPS/1.E-25/                                                 MIS2024
C     START REDUCTION OF MATRIX A                                       MIS2025
C                                                                       MIS2026
      DO 80  I=1,N                                                      MIS2027
C                                                                       MIS2028
C     SEARCH FOR MAXIMUM ELEMENT IN ITH ROW OF A-MATRIX                 MIS2029
C                                                                       MIS2030
      AIJMAX = A(I,1)                                                   MIS2031
      JMAX = 1                                                          MIS2032
      DO 10  J=2,N                                                      MIS2033
      IF (  ABS(A(I,J)) .LE.   ABS(AIJMAX)) GO TO 10                    MIS2034
      AIJMAX = A(I,J)                                                   MIS2035
      JMAX = J                                                          MIS2036
   10 CONTINUE                                                          MIS2037
C                                                                       MIS2038
C     IF AIJMAX IS ZERO, THE MATRIX IS SINGULAR                         MIS2039
C                                                                       MIS2040
      IF (ABS(AIJMAX) .GT. EPS)  GO TO 20                               MIS2041
      D =  0.0E0                                                        MIS2042
      NERR =  1                                                         MIS2043
      RETURN                                                            MIS2044
C                                                                       MIS2045
C     NORMALIZE ITH ROW BY AIJMAX (JMAX ELEMENT OF ITH ROW)             MIS2046
C                                                                       MIS2047
   20 DO 30  J=1,N                                                      MIS2048
   30 A(I,J) = A(I,J) / AIJMAX                                          MIS2049
      D = D*AIJMAX                                                      MIS2050
C                                                                       MIS2051
C     NORMALIZE ITH ROW OF B                                            MIS2052
C                                                                       MIS2053
      DO 40  J=1,MD                                                     MIS2054
   40 B(I,J) = B(I,J) / AIJMAX                                          MIS2055
C                                                                       MIS2056
```

```
C     USE ROW TRANSFORMATIONS TO GET ZEROS ABOVE AND BELOW THE JMAX     MIS2057
C     ELEMENT OF THE ITH ROW OF A.  APPLY SAME ROW TRANSFORMATIONS       MIS2058
C     TO THE B MATRIX.                                                  MIS2059
C                                                                       MIS2060
      DO 70  K=1,N                                                      MIS2061
      IF (K .EQ. I) GO TO 70                                            MIS2062
      ARAT = -A(K,JMAX)                                                 MIS2063
      DO 50  J=1,N                                                      MIS2064
      IF (ABS(A(I,J)) .LT. EPS)  GO TO 50                               MIS2065
      A(K,J) = ARAT * A(I,J) + A(K,J)                                   MIS2066
   50 CONTINUE                                                          MIS2067
      A(K,JMAX) =  0.0E0                                                MIS2068
      DO 60  J=1,MD                                                     MIS2069
      IF (ABS(B(I,J)) .LT. EPS)  GO TO 60                               MIS2070
      B(K,J) = ARAT * B(I,J) + B(K,J)                                   MIS2071
   60 CONTINUE                                                          MIS2072
   70 CONTINUE                                                          MIS2073
C                                                                       MIS2074
C     STORE ROW COUNTER (I) IN TOP ELEMENT OF JMAX COLUMN. THUS,        MIS2075
C     THE TOP ROW OF A WILL CONTAIN THE LOC OF THE PIVOT (UNITY)        MIS2076
C     ELEMENT OF EACH COLUMN (AFTER REDUCTION).                         MIS2077
C                                                                       MIS2078
      L = I                                                             MIS2079
   80 A(1,JMAX) = FL                                                    MIS2080
C                   THIS STORES INTEGER I IN TOP ROW OF A               MIS2081
C                                                                       MIS2082
C     THE REDUCTION OF A IS NOW COMPLETE.  PERFORM ROW INTERCHANGES     MIS2083
C     AS INDICATED IN THE FIRST ROW OF A.                               MIS2084
C                                                                       MIS2085
      DO 120  I=1,N                                                     MIS2086
      K = I                                                             MIS2087
   90 FK = A(1,K)                                                       MIS2088
C                   THIS PUTS THE INTEGER VALUE IN A INTO K             MIS2089
      IF (K-I)90,120,100                                                MIS2090
C                                                                       MIS2091
C     IF K(1,I) IS LESS THAN I, THEN THAT ROW HAS ALREADY BEEN          MIS2092
C     INVOLVED IN AN INTERCHANGE, AND WE USE K(1,K) UNTIL WE GET        MIS2093
C     A VALUE OF K GREATER THAN I  (CORRESPONDING TO A ROW STORED       MIS2094
C     BELOW THE ITH ROW.  (CLEAR AS MUD)                                MIS2095
C                                                                       MIS2096
  100 DO 110  J=1,MD                                                    MIS2097
      ARAT =  B(I,J)                                                    MIS2098
      B(I,J) = B(K,J)                                                   MIS2099
  110 B(K,J) = ARAT                                                     MIS2100
      D = -D                                                            MIS2101
  120 CONTINUE                                                          MIS2102
      NERR = 0                                                          MIS2103
      RETURN                                                            MIS2104
      END                                                               MIS2105
```

```
      SUBROUTINE  OFFBOD(N, M, CHORDB, IDB, IBTOT)                       OFFB001
C                                                                        OFFB002
      DIMENSION  X(100), Y(100), TITLE(7)                                OFFB003
     1           ,CHORDB(10), IDB(10)                                    OFFB004
      COMMON/COMBOD/CCL,INCLT,CLT,ALPHA,SUMDS(10),TLU(10,12),IND         OFFB005
     1                  , ALPHAO, CNU(10), SMDSWF(10), MIO(10)           OFFB006
C                                                                        OFFB007
C  READ IN OFFBODY POINTS (IN BLOCKS OF UP TO 100)                       OFFB008
   10 CALL OFFPTS(NO, X, Y, TITLE, LAST, CHORDB, IDB, IBTOT)             OFFB009
C                                                                        OFFB010
C  NOW CALCULATE VX AND VY                                               OFFB011
      CALL  VXYOFF(N, M, NO, X, Y)                                       OFFB012
C                                                                        OFFB013
C                                                                        OFFB014
C  PRINT OFFBODY POINTS AND VELOCITIES                                   OFFB015
      CALL VPROFF(N, M, NO, X, Y, TITLE, IND)                            OFFB016
C                                                                        OFFB017
C  CHECK IF MORE POINTS AND CYCLE IF SO                                  OFFB018
      IF (LAST .NE. 1)  GO TO 10                                         OFFB019
C                                                                        OFFB020
C                                                                        OFFB021
      RETURN                                                             OFFB022
      END                                                                OFFB023
```

```
      SUBROUTINE OFFPTS(NO, X, Y, TITLE, LAST, CHORDB, ICB, IBTOT)      OFFP001
C                                                                       OFFP002
      DIMENSION X(1), Y(1), TITLE( 7)                                   OFFP003
     1          ,CHORDB(10), IDB(10)                                    OFFP004
C                                                                       OFFP005
C                                                                       OFFP006
      DATA EPS/1.0E-7/, CR/1.79532925E-2/                               OFFP007
C                                                                       OFFP008
C  READ IN BODY TITLE AND CONTROL CARD                                  OFFP009
   10 ITYP = 21                                                         OFFP010
      READ(5,20) ID, TITLE, ITR, INORM, IDOLD, LAST, ITYPE              OFFP011
   20 FORMAT( I1, 9X7A4, 12X, 2(2XI1), 2(5XI1), 2XI2)                   OFFP012
      IF (ITYPE .NE. ITYP)  CALL TYPE(ITYP, ITYPE)                      OFFP013
C                                                                       OFFP014
C  READ IN COORDINATE TRANSFORMATION CARD IF REQUIRED                   OFFP015
      ITYP = 22                                                         OFFP016
      CHORD = 0.0                                                       OFFP017
      IF (ITR .EQ. 0 .OR. ITR .EQ. 2)  GO TO 40                         OFFP018
      READ (5,30) CHORD, XMULT, YMULT, DX, DY, THETA, XTO, YTO, ITYPE   OFFP019
   30 FORMAT (7(F8.0,1X), F8.0, I1)                                     OFFP020
      IF (ITYPE .NE. (ITYP-20))  CALL TYPE(ITYP, ITYPE)                 OFFP021
C                                                                       OFFP022
   40 CONTINUE                                                          OFFP023
C                                                                       OFFP024
C  CHECK IF ELLIPSE TO BE GENERATED                                     OFFP025
      IF (ITR .GT. 1)  GO TO 90                                         OFFP026
C                                                                       OFFP027
C  DATA ON UNIT 5.  X-COORDS FIRST                                      OFFP028
      L = 0                                                             OFFP029
      ITYP = 23                                                         OFFP030
   50 READ (5,60)  (X(L+I), I=1,6), INO, ISTAT, ITYPE                   OFFP031
   60 FORMAT (6F10.0, 4XI1, 2XI1, 3XI1)                                 OFFP032
      IF (ITYPE .NE. (ITYP-20))  CALL TYPE(ITYP, ITYPE)                 OFFP033
      IF (INO .LE. 0  .OR.  INO .GT. 6) INO = 6                         OFFP034
      L = L + INO                                                       OFFP035
      IF (ISTAT .EQ. 0) GO TO 50                                        OFFP036
      LX = L                                                            OFFP037
C                                                                       OFFP038
C  NOW READ IN Y-COORDS                                                 OFFP039
      L = 0                                                             OFFP040
      ITYP = 24                                                         OFFP041
   70 READ (5,60)  (Y(L+I), I=1,6), INO, ISTAT, ITYPE                   OFFP042
      IF (ITYPE .NE. (ITYP-20))  CALL TYPE(ITYP, ITYPE)                 OFFP043
      IF (INO .LE. 0  .OR.  INO .GT. 6) INO = 6                         OFFP044
      L = L + INO                                                       OFFP045
      IF (ISTAT .EQ. 0) GO TO 70                                        OFFP046
      LY = L                                                            OFFP047
C                                                                       OFFP048
C  CHECK FOR INPUT CONSISTENCY                                          OFFP049
      IF (LY .EQ. LX) GO TO 120                                         OFFP050
      WRITE (6,80) LY, LX                                               OFFP051
   80 FORMAT (1H1,'THE NUMBER OF Y-COORDINATES (',I3,') READ DOES ',    OFFP052
     1             'NOT EQUAL THE NUMBER OF X-COORDINATES READ (',I3,')')OFFP053
      GO TO 200                                                         OFFP054
C                                                                       OFFP055
C  ELLIPSE TO BE GENERATED.  READ IN DEFINITION CARD.                   OFFP056
```

```
   90 ITYP = 25                                                         OFFP057
      READ (5,100) LX, ELPSTH, ITYPE                                    OFFP058
  100 FORMAT (2XI3, 5XF10.5, 51XI1)                                     OFFP059
      IF (ITYPE .NE. (ITYP-20))  CALL TYPE(ITYP, ITYPE)                 OFFP060
      ITR = ITR - 2                                                     OFFP061
      IF (ITR .NE. 1)  ITR = 0                                          OFFP062
C                                                                       OFFP063
      DANGLE = 6.2831853072/(LX - 1)                                    OFFP064
      ANGLE = DANGLE                                                    OFFP065
      DO 110 I = 1,LX                                                   OFFP066
      ANGLE = ANGLE - DANGLE                                            OFFP067
      X(I) = COS(ANGLE)                                                 OFFP068
  110 Y(I) = SIN(ANGLE)*ELPSTH                                          OFFP069
C                                                                       OFFP070
  120 CONTINUE                                                          OFFP071
C                                                                       OFFP072
C  WRITE OUT BASIC GEOMETRY DATA                                        OFFP073
      IP = 1                                                            OFFP074
C                                                                       OFFP075
C                                                                       OFFP076
C  TRANSFORM COORDINATES IF REQUESTED                                   OFFP077
      IF (ITR .EQ. 1) GO TO 130                                         OFFP078
      IF (INORM .EQ. 0)  GO TO 220                                      OFFP079
      XMULT = 0.0                                                       OFFP080
      YMULT = 0.0                                                       OFFP081
      XTO = 0.0                                                         OFFP082
      YTO = 0.0                                                         OFFP083
      THETA = 0.0                                                       OFFP084
      DX = 0.0                                                          OFFP085
      DY = 0.0                                                          OFFP086
  130 CONTINUE                                                          OFFP087
C                                                                       OFFP088
      IF (ABS(XMULT) .LT. EPS)      XMULT = 1.0                         OFFP089
      IF (ABS(YMULT) .LT. EPS)      YMULT = 1.0                         OFFP090
      XSF = XMULT                                                       OFFP091
      YSF = YMULT                                                       OFFP092
      IF (INORM .EQ. 0)  GO TO 180                                      OFFP093
C                                                                       OFFP094
      IF (IDOLD .LE. 0)  GO TO 160                                      OFFP095
      DO 140  IB = 1,IBTOT                                              OFFP096
      IF (IDB(IB) .EQ. IDOLD)  GO TO 150                                OFFP097
  140 CONTINUE                                                          OFFP098
      GO TO 160                                                         OFFP099
  150 CHORD = CHORDB(IB)                                                OFFP100
  160 IF (ABS(CHORD) .LE. EPS)  GO TO 180                               OFFP101
  170 XSF = XSF/CHORD                                                   OFFP102
      YSF = YSF/CHORD                                                   OFFP103
C                                                                       OFFP104
  180 COST = COS(THETA*DR)                                              OFFP105
      SINT = SIN(THETA*DR)                                              OFFP106
      DO 190 I = 1,LX                                                   OFFP107
      XTOD = X(I) - XTO                                                 OFFP108
      YTOD = Y(I) - YTO                                                 OFFP109
      X(I) = (XTO + XTOD*COST - YTOD*SINT + DX)*XSF                     OFFP110
      Y(I) = (YTO + YTOD*SINT + YTOD*COST + DY)*YSF                     OFFP111
  190 CONTINUE                                                          OFFP112
      GO TO 220                                                         OFFP113
```

```
  200 WRITE (6,210)                                                      OFFP114
  210 FORMAT (1H0, 'BECAUSE OF THE ABOVE ERROR, THIS RUN IS TERMINATED')OFFP115
      STOP                                                               OFFP116
C                                                                        OFFP117
C                                                                        OFFP118
  220 CONTINUE                                                           OFFP119
      NO = LX                                                            OFFP120
      RETURN                                                             OFFP121
      END                                                                OFFP122
```

```
      SUBROUTINE    PRINIG (IP, N, ID, BTITLE)                           PRN6001
C                                                                        PRN6002
C  THIS SUBROUTINE WRITES OUT THE BODY COORDINATE DATA                   PRN6003
C                                                                        PRN6004
C                                                                        PRN6005
      DIMENSION BTITLE ( 7), A(2)                                        PRN6006
C                                                                        PRN6007
      COMMON /GEOMD/ X(500), Y(500)                                      PRN6008
C                                                                        PRN6009
      DATA A(1),A(2)/'UNTR','  TR'/                                      PRN6010
C                                                                        PRN6011
C                                                                        PRN6012
      J2 = 0                                                             PRN6013
   10 WRITE (6,30)                                                       PRN6014
      WRITE (6,20) A(IP), ID, BTITLE                                     PRN6015
   20 FORMAT(1H0,15XA4,'ANSFORMED COORDINATE DATA FOR BODY ID = ',I2,    PRN6016
     1       ',  ',  7A9//T15,'I', T27,'X(I)', T44,'Y(I)',                PRN6017
     2                     T75,'I', T87,'X(I)',T104,'Y(I)')              PRN6018
   30 FORMAT (1H1)                                                       PRN6019
C                                                                        PRN6020
      J1 = J2 + 1                                                        PRN6021
      JMX = J1 + 49                                                      PRN6022
      N2 = (N + 1 + J2)/2                                                PRN6023
      IF (JMX .GT. N2)   JMX = N2                                        PRN6024
      J2 = JMX                                                           PRN6025
      DO 50 J = J1,JMX                                                   PRN6026
      J2 = J2  + 1                                                       PRN6027
      IF (J2 .GT. N) GO TO 60                                            PRN6028
      WRITE (6,40) J, X(J), Y(J), J2, X(J2), Y(J2)                       PRN6029
   40 FORMAT (1H ,11XI3, 2(5XF12.6), 23XI3, 2(5XF12.6))                  PRN6030
   50 CONTINUE                                                           PRN6031
C                                                                        PRN6032
C                                                                        PRN6033
      IF (J2 .LT. N)  GO TO 10                                           PRN6034
      RETURN                                                             PRN6035
C                                                                        PRN6036
C                                                                        PRN6037
   60 WRITE (6,40) J, X(J), Y(J)                                         PRN6038
C                                                                        PRN6039
      RETURN                                                             PRN6040
      END                                                                PRN6041
```

```
      SUBROUTINE    PRNTEL                                              PRNL001
C                                                                       PRNL002
C                                                                       PRNL003
      DIMENSION AN(10)                                                  PRNL004
      REAL NEW,NO,NCHGE                                                 PRNL005
C                                                                       PRNL006
      COMMON /BFLAG/ IOB(10), INL(10), IFL(10), NL(10), LIFT(10),       PRNL007
     1               IBMF(10), ISAV1(10), ISAV2(10), ISAV3(10),         PRNL008
     2              BTITLE(10, 7), IBT, IBST, IBDTOT, NELTOT,           PRNL009
     3              ITRB(10), INMB(10), CHORDB(10), IBD(10),LIFTOT      PRNL010
     4              ,IPRE(10), IFST(10), ISEC(10), FTITLE(15), IPVR(10) PRNL011
C                                                                       PRNL012
C                                                                       PRNL013
      DATA NEW,OLD,YES,NO,SAVED,BLANK/' NEW',' OLD',' YES',' NO ',      PRNL014
     1     ' S ','   '/,CHGE,NCHGE/'CHGE',' NC '/,AN/' 0 ',' 1 ',       PRNL015
     2     ' 2 ',' 3 ',' 4 ',' 5 ',' 6 ',' 7 ',' 8 ',' 9 '/             PRNL016
     3     , AP, AL/ ' P ', ' L '/                                      PRNL017
C                                                                       PRNL018
C  WRITE HEADING FOR BODY GEOMETRY SUMMARY                              PRNL019
      WRITE (6,10)                                                      PRNL020
   10 FORMAT (1H1, T51,'BODY  GEOMETRY  SUMMARY'//                      PRNL021
     1          1H0,T33,'BODY    LIFT',T99,'SIGMA',                     PRNL022
     2               T114,'ELEMENT  STORAGE'/                           PRNL023
     3          T6,'BODY  DESCRIPTION',T34,'ID      TYPE',              PRNL024
     4          T50,'N/O    SID   TFORM  NORM      CHORD       TYPE',   PRNL025
     5          T97,'F  C    S  C    N/O     FIRST     NO.')            PRNL026
C                                                                       PRNL027
C                                                                       PRNL028
      IBDTOT = 0                                                        PRNL029
      NELTOT = 0                                                        PRNL030
      LIFTOT = 0                                                        PRNL031
      DO 150 IB = 1,IBT                                                 PRNL032
      IF (IBD(IB) .GE. 6)  GO TO 150                                    PRNL033
C                                                                       PRNL034
      IF (IBD(IB) - 3)20,20,30                                          PRNL035
   20 A2 = NEW                                                          PRNL036
      A3 = BLANK                                                        PRNL037
      IF (ISAV3(IB) .GE. 0)   A3 = SAVED                                PRNL038
      GO TO 40                                                          PRNL039
C                                                                       PRNL040
   30 A2 = OLD                                                          PRNL041
      IN = ISAV1(IB) + 1                                                PRNL042
      A3 = AN(IN)                                                       PRNL043
C                                                                       PRNL044
   40 A4 = NO                                                           PRNL045
      A5 = NO                                                           PRNL046
      IF (ITRB(IB) .EQ. 1)     A4 = YES                                 PRNL047
      IF(INMB(IB) .EQ. 1)      A5 = YES                                 PRNL048
C                                                                       PRNL049
      A1 = NO                                                           PRNL050
      IF (LIFT(IB) .EQ. 1)     A1 = YES                                 PRNL051
C                                                                       PRNL052
      A6 = OLD                                                          PRNL053
      IF (IBMF(IB) - 1)70,50,60                                         PRNL054
   50 A6 = NEW                                                          PRNL055
      GO TO 70                                                          PRNL056
```

```
      60 A6 = CHGE                                                      PRNL057
C                                                                       PRNL058
C                                                                       PRNL059
      70 A7 = AP                                                        PRNL060
         I8 = 1                                                         PRNL061
         I9 = 1                                                         PRNL062
         I10= 1                                                         PRNL063
         I11= 1                                                         PRNL064
C                                                                       PRNL065
         IF (IPRB(IB) .NE. 1)  A7 = AL                                  PRNL066
         IF (IFST(IB) - 2)80,90,100                                     PRNL067
      80 I9 = 0                                                         PRNL068
         IF (IFST(IB) .EQ. 1)  GO TO 100                                PRNL069
      90 I8 = 0                                                         PRNL070
     100 IF (ISEC(IB) - 2)110,120,130                                   PRNL071
     110 I11 = 0                                                        PRNL072
         IF (ISEC(IB) .EQ. 1)  GO TO 130                                PRNL073
     120 I10 = 0                                                        PRNL074
     130 CONTINUE                                                       PRNL075
C                                                                       PRNL076
         IBDTOT = IBDTOT + 1                                            PRNL077
         NELTOT = NELTOT + NL(IB)                                       PRNL078
         IF (LIFT(IB) .EQ. 1)  LIFTOT = LIFTOT + 1                      PRNL079
         WRITE (6,140)  (BTITLE(IB,I), I=1,7), IDB(IB), A1, A2, A3,     PRNL080
        1              A4, A5, CHORDB(IB), A7, I8, I9, I10, I11, A6,    PRNL081
        2              INL(IB), NL(IB)                                  PRNL082
     140 FORMAT (1H0,     7A4, 4XI2, 5XA4, 4XA4, 2XA4, 3XA4, 3XA4, 2XF12.6,PRNL083
        1       3XA4, 3XI1, 2XI1, 3XI1, 2XI1, 3XA4, 5XI4, 4XI4)         PRNL084
C                                                                       PRNL085
     150 CONTINUE                                                       PRNL086
C                                                                       PRNL087
C                                                                       PRNL088
         WRITE (6,160) IBDTOT, NELTOT                                   PRNL089
     160 FORMAT (1H0, ///T53, 'TOTAL NUMBER OF BODIES = ',I3/           PRNL090
        1         1H0, //T51,'TOTAL NUMBER OF ELEMENTS = ',I3)          PRNL091
C                                                                       PRNL092
C                                                                       PRNL093
         RETURN                                                         PRNL094
         END                                                            PRNL095
```

```
      SUBROUTINE QUASI (A,ND,MD,KD,NI,MM,NO,NAT,NW,LTAPE,RHSTAP,*)
C
C
C          ****    ***/  *******  *****  *        ***/       *
C          *   *  *  /*     *      *     *       *  /*    ******
C   ****   ****   * / *     *     ***    *       * / *  ****    *  *
C          *      */  *     *      *     *       */  *          ***
C          *      /***      *      *     *****   /***           *  *
C
C                D I R E C T   M A T R I X   S O L U T I O N
C
C*** ***LTAPE IS THE TAPE THE L(I,J) MATRIX WILL BE PUT ON
C*** ***RHSTAP IS THE TAPE THAT THE RIGHT HAND SIDES ARE INPUT ON
      INTEGER RHSTAP
C*** ***NATAPE IS A SCRATCH TAPE
C*** ***THE TRIANGULAR MATRIX EXCEPT FOR THE LAST K ROWS WILL BE KEPT ON
C*** ***TAPE MM
C*** ***THE LAST K ROWS OF THE TRIANGULAR MATRIX WILL BE PUT ON THE
C*** ***LTAPE BEHIND THE RHS MATRIX
C
C     COMPLEX A,SUM
      DIMENSION  A ( KD )
C
      LOGICAL JPASS1
      LOGICAL  LASTRS
      LOGICAL  LAST
C
      CALL TIME(AA1)
   10 CONTINUE
      REWIND LTAPE
      NATAPE = NAT
      REWIND NATAPE
      N = ND
      KORE = KD
C*** *  RHSTAP = 0 IF THERE ARE NO RHS TO BE PROCESSED THIS RUN
      IF(RHSTAP .NE. 0 )GO TO 20
      MRHS = 0
      GO TO 30
   20 REWIND RHSTAP
      READ(RHSTAP)MRHS
   30 M = KORE / N  -  1
      MMAX = MINO(MRHS,M)
      NPM = N + MMAX
      IF( (3*NPM) .GT. KORE )RETURN 1
      M = 0
      NPM = N
      MT = MM
      REWIND MT
      NIN = NI
      REWIND NIN
      NOUT = NO
      REWIND NOUT
      MP1 = M + 1
      NN = N
      NEL = NPM
      NLCNT = 0
```

```
      C                                                                 QUAS057
      C - - CALCULATE THE MAXIMUM NO. OF ROWS. 'K'                      QUAS058
      C                                                                 QUAS059
   40 K = (KORE - NEL) / NEL                                            QUAS060
      C                                                                 QUAS061
      C - - TEST TO SEE IF THE REST OF THE MATRIX WILL FIT IN CORE      QUAS062
      C                                                                 QUAS063
        LAST = K .GE. NN                                                QUAS064
        IF( .NOT. LAST ) GO TO 50                                       QUAS065
        K = NN                                                          QUAS066
        B = 3 + MMAX*2                                                  QUAS067
        C = 2 * (1 + MMAX - KORE )                                      QUAS068
        KTEMP = ( -B + SQRT(B**2  -  4 * C )  )  / 2                    QUAS069
        IF(KTEMP .GE. K)GO TO 50                                        QUAS070
      C                                                                 QUAS071
      C***   *  WE MUST REDUCE THE FINAL K                              QUAS072
      C                                                                 QUAS073
        K = KTEMP                                                       QUAS074
        LAST = .FALSE.                                                  QUAS075
      C                                                                 QUAS076
      C - - READ 'K' ROWS OF THE AUGMENTED 'A' MATRIX                   QUAS077
      C                                                                 QUAS078
   50 NT = 0                                                            QUAS079
        DO 60 IB = 1, K                                                 QUAS080
        NS = NT + 1                                                     QUAS081
        NT = NT + NEL                                                   QUAS082
   60 CALL  GETT(NIN, 1, NEL, A(NS), 1, AA2)                            QUAS083
      C                                                                 QUAS084
      C - - CHECK TO SEE IF WE WERE UNLUCKY ENOUGH TO END UP WITH ONLY ONE ROWQUAS085
      C                                                                 QUAS086
        IF (K .EQ. 1) GO TO 130                                         QUAS087
      C                                                                 QUAS088
      C - - 'K' IS GREATER THAN '1' SO WE CAN START THE TRIANGULARIZATION QUAS089
      C                                                                 QUAS090
        NELP1 = NEL + 1                                                 QUAS091
        NS = - NEL                                                      QUAS092
        NELP2 = NELP1 + 1                                               QUAS093
      C                                                                 QUAS094
      C - - FORM THE 'TRAPEZOIDAL' ARRAY  (B)                           QUAS095
      C                                                                 QUAS096
        DO 70 IB = 2, K                                                 QUAS097
        NP = NELP2 - IB                                                 QUAS098
        NS = NS + NELP1                                                 QUAS099
        NT = NS                                                         QUAS100
        DO 70 IO = IB, K                                                QUAS101
        NT = NT + NEL                                                   QUAS102
        MN = NT                                                         QUAS103
        NB = NS                                                         QUAS104
        A(NT) = A(NT) / A(NS)                                           QUAS105
        DO 70 NF = 2, NP                                                QUAS106
        MN = MN + 1                                                     QUAS107
        NB = NB + 1                                                     QUAS108
   70 A(MN) = A(MN) - A(NT) * A(NB)                                     QUAS109
      C*** ***WRITE PART OF THE LMATRIX ON LTAPE   (TRIANGULAR PART)    QUAS110
        WRITE (LTAPE)K                                                  QUAS111
        NLCNT = NLCNT + 1                                               QUAS112
        LBEG =NELP1                                                     QUAS113
```

```
      KM1 = K - 1                                                       QUAS114
      DO 80 IB = 1,KM1                                                  QUAS115
      LEND = LBEG + IB- 1                                               QUAS116
      CALL SAVE(LTAPE, 1, IB, IB, A(LBEG), 1, AA2)                      QUAS117
   80 LBEG = LBEG + NN                                                  QUAS118
C                                                                       QUAS119
C - - WRITE THE 'TRAPEZOIDAL' MATRIX ON TAPE                            QUAS120
C                                                                       QUAS121
      NT = 0                                                            QUAS122
      NP = NEL                                                          QUAS123
      NS = - NEL                                                        QUAS124
      DO 90 IO = 1, K                                                   QUAS125
      NS = NS + NELP1                                                   QUAS126
      NT = NT + NEL                                                     QUAS127
      CALL  SAVE(NT, 2, NP, NP, A(NS), 1, AA2)                          QUAS128
   90 NP = NP - 1                                                       QUAS129
      IF (LAST) GO TO 130                                               QUAS130
      NP = NP - M                                                       QUAS131
      NS = KORE - NEL + 1                                               QUAS132
C                                                                       QUAS133
C - - READ ANOTHER ROW                                                  QUAS134
C                                                                       QUAS135
      DO 120 IO = 1, NP                                                 QUAS136
      CALL  GETT(NIN, 1, NEL, A(NS), 1, AA2)                            QUAS137
C                                                                       QUAS138
C - - MODIFY THIS ROW BY THE 'TRAPEZOIDAL' ARRAY                        QUAS139
C                                                                       QUAS140
      NT = 1                                                            QUAS141
      MN = NS                                                           QUAS142
      DO 110 IB = 1, K                                                  QUAS143
      NB = NT                                                           QUAS144
      NF = MN + 1                                                       QUAS145
      A(MN) = A(MN) / A(NT)                                             QUAS146
      DO 100 NN = NF, KORE                                              QUAS147
      NB = NB + 1                                                       QUAS148
  100 A(NN) = A(NN) - A(MN) * A(NB)                                     QUAS149
      MN = NF                                                           QUAS150
  110 NT = NT + NELP1                                                   QUAS151
C                                                                       QUAS152
C - - WRITE THE MODIFIED ROW ON TAPE                                    QUAS153
C                                                                       QUAS154
C*** ***WRITE REST OF LMATRIX ON LTAPE                                  QUAS155
      MNM1 = MN - 1                                                     QUAS156
      NN1 = MNM1 - NS + 1                                               QUAS157
      CALL  SAVE(LTAPE, 1, NN1, NN1, A(NS), 1, AA2)                     QUAS158
      NN1 = KORE - MN + 1                                               QUAS159
  120 CALL  SAVE(NOUT, 1, NN1, NN1, A(MN), 1, AA2)                      QUAS160
      REWIND NOUT                                                       QUAS161
      REWIND NIN                                                        QUAS162
C                                                                       QUAS163
C - - SWITCH THE TAPES                                                  QUAS164
C                                                                       QUAS165
      NT = NIN                                                          QUAS166
      NIN = NOUT                                                        QUAS167
      NOUT = NT                                                         QUAS168
C                                                                       QUAS169
C - - RE-CALCULATE ROW LENGTH AND LOOP PACK                             QUAS170
```

```
      C                                                                 QUAS171
            NEL = NEL - K                                               QUAS172
            NN = NEL - M                                                QUAS173
            GO TO 40                                                    QUAS174
      C                                                                 QUAS175
      C - - REWIND ALL TAPES                                            QUAS176
      C                                                                 QUAS177
        130 REWIND NIN                                                  QUAS178
            REWIND NOUT                                                 QUAS179
      C                                                                 QUAS180
        140 N1 = KORE - K * M + 1                                       QUAS181
            REWIND LTAPE                                                QUAS182
            REWIND MT                                                   QUAS183
      C*** ***CALCULATE THE NUMBER OF COLUMNS TO BRING OFF OF THE RHS TAPE  QUAS184
            MTOTAL = 0                                                  QUAS185
            M = MMAX                                                    QUAS186
            IF(M .EQ. 0)GO TO 520                                       QUAS187
      C*** ***MTOTAL IS THE TOTAL NUMBER OF RHS COLUMNS ALREADY BROUGHT IN  QUAS188
        150 MTOTAL = MTOTAL + M                                         QUAS189
            LASTRS = MTOTAL.GE. MRHS                                    QUAS190
            MTOTAL = MTOTAL - M                                         QUAS191
            IF (LASTRS)M = MRHS - MTOTAL                                QUAS192
            MTOTAL = MTOTAL + M                                         QUAS193
      C*** ***BRING IN M COLUMNS OF RHS                                 QUAS194
            KINIT = KORE - (M*N)                                        QUAS195
            IINIT = KINIT                                               QUAS196
            NBEG = KINIT + 1                                            QUAS197
            NEND = KINIT+ N                                             QUAS198
            DO 160 J =1,M                                               QUAS199
            CALL  GETT(RHSTAP, 1, N, A(NBEG), 1, AA2)                   QUAS200
            NBEG = NEND + 1                                             QUAS201
        160 NEND = NEND + N                                             QUAS202
      C*** ***BRING IN L(I,J) MATRIX AND APPLY IT TO RHS                QUAS203
            NBEG = 1 + KINIT                                            QUAS204
            NEND = 1 + (M-1) * N + KINIT                                QUAS205
            KSUM = 0                                                    QUAS206
      C*** ***DO TRIANGULAR SECTION OF LMATRIX                          QUAS207
        170 READ (LTAPE)K                                               QUAS208
      C*** ***KSUM IS THE TOTAL NUMBER OF L ROWS THAT WILL              QUAS209
      C*** ***BE READ AFTER THIS TRIANGULAR SECTION IS FINISHED         QUAS210
            KSUM = KSUM + K                                             QUAS211
            KM1 = K - 1                                                 QUAS212
      C***    ***NOTE THAT KM1 CAN'T BE 0 SINCE K CAN'T BE 1 AND STILL HAVE SOMQUAS213
      C***    ***ON THE LTAPE                                           QUAS214
            DO 200  I = 1,KM1                                           QUAS215
            NBEG = NBEG + 1                                             QUAS216
            NEND = NEND + 1                                             QUAS217
      C*** ***READ 1 ROW OF L(I,J) FROM LTAPE---K-1 TIMES---EACH TIME   QUAS218
      C*** ***STARTING WITH L(1)                                        QUAS219
            CALL  GETT(LTAPE, 1, I, A, 1, AA2)                          QUAS220
            JCNT = -1                                                   QUAS221
      C*** ***REDUCE THE RHS BY GOING ACROSS A SOLUTION ROW (WHICH      QUAS222
      C*** ***ARE NOT IN CONSECUTIVE ORDER, BUT A(1), A(N+1), A(2N+1) ETC.)  QUAS223
            DO 190 NPP = NBEG,NEND,N                                    QUAS224
            JCNT = JCNT + 1                                             QUAS225
            SUM = 0.0                                                   QUAS226
            NROW = KINIT + ( JCNT * N )                                 QUAS227
```

```
      DO 180 NN=1,I                                                     QUAS228
      NROW = NROW + 1                                                   QUAS229
  180 SUM = SUM + ( A(NN)*A(NROW))                                      QUAS230
  190 A(NPP) = A(NPP) - SUM                                             QUAS231
  200 CONTINUE                                                          QUAS232
      IF (KSUM .EQ. N)GO TO 240                                         QUAS233
C*** ***KSUM = N IF YOU HAVE READ ENTIRE LMATRIX AND                    QUAS234
C*** ***THERE IS NO CONSTANT SECTION LEFT                               QUAS235
      NTBEG = NBEG                                                      QUAS236
      NTEND = NEND                                                      QUAS237
      KSUMP1 = KSUM + 1                                                 QUAS238
C*** ***READ REST OF LROWS 1 ROW AT A TIME FOR CONSTANT SECTION         QUAS239
      DO 230 I=KSUMP1,N                                                 QUAS240
      NTBEG = NTBEG + 1                                                 QUAS241
      NTEND = NTEND + 1                                                 QUAS242
      CALL  GETI(LTAPE, 1, K, A, 1, AA2)                                QUAS243
      JCNT = -1                                                         QUAS244
C*** ***PARTIALLY  REDUCE A RHS ACROSS A RHS ROW BY APPLYING K NUMBER   QUAS245
C*** ***OF L(I,J) S                                                     QUAS246
      DO 220 NPP = NTBEG,NTEND,N                                        QUAS247
      JCNT = JCNT + 1                                                   QUAS248
      SUM = 0.0                                                         QUAS249
      NROW = KINIT + ( JCNT * N )                                       QUAS250
      DO 210 NN = 1,K                                                   QUAS251
      NROW = NROW + 1                                                   QUAS252
  210 SUM = SUM + ( A(NN) * A(NROW) )                                   QUAS253
  220 A(NPP) = A(NPP) - SUM                                             QUAS254
  230 CONTINUE                                                          QUAS255
      NBEG = NBEG + 1                                                   QUAS256
      NEND = NEND + 1                                                   QUAS257
C*** ***KINIT IS HOW FAR DOWN A COLUMN OF RHS TO START MULTIPLYING BY   QUAS258
C*** ***L(I,J) AT EACH PASS THROUGH                                     QUAS259
      KINIT = KINIT + K                                                 QUAS260
C*** ***IF KSUMP1 = N THERE ARE NO MORE L(I,J)'S LEFT                   QUAS261
      IF(KSUMP1 .LT. N)GO TO 170                                        QUAS262
C*** ***WRITE OUT ALL BUT LAST K ROWS OF RHS IN ROW ORDER ON NATAPE     QUAS263
  240 B = 4*M + 3                                                       QUAS264
      C = -2 * KORE                                                     QUAS265
      K = ( -B +  SQRT( B**2  -  4*C )  )/2                             QUAS266
      IF(K .GT. ND) K = ND                                              QUAS267
      KF = K                                                            QUAS268
      KM1 = K - 1                                                       QUAS269
      KLEFT = N - KF + IINIT                                            QUAS270
      INITP1 = IINIT + 1                                                QUAS271
      NEND = (M-1)*N + IINIT                                            QUAS272
      DO 250 NPP = INITP1,KLEFT                                         QUAS273
      NEND = NEND + 1                                                   QUAS274
  250 WRITE(NATAPE) ( A(J),J=NPP,NEND,N)                                QUAS275
      REWIND NATAPE                                                     QUAS276
C*** ***JPASS1 IS TRUE ON 1ST PASS THRU BACK SOLUTION                   QUAS277
      JPASS1 = .TRUE.                                                   QUAS278
C                                                                       QUAS279
C***    ***PUT REMAINING RHS IN CONTIGUOUS LOCATIONS BY COLUMNS         QUAS280
C***  FROM KORE - (M * KF) + 1  TO  KORE                                QUAS281
C                                                                       QUAS282
      NNEW = KORE - KF + 1                                              QUAS283
      MM1 = M - 1                                                       QUAS284
```

```
C                                                                       QUAS285
C***     ***IF M = 1, THE ELTS OF THE 1 RHS COLUMN ARE ALREADY IN CONTIGUOQUAS286
C***     ***LOCATIONS                                                   QUAS287
C                                                                       QUAS288
      IF ( M .EQ. 1 ) GO TO 265                                         QUAS289
      DO 260 I = 1,MM1                                                  QUAS290
      NOLD = KORE - (I*N) + 1                                           QUAS291
      DO 260 J = 1,KF                                                   QUAS292
      NNEW = NNEW - 1                                                   QUAS293
      NOLD = NOLD - 1                                                   QUAS294
      A(NNEW) = A(NOLD)                                                 QUAS295
  260 CONTINUE                                                          QUAS296
  265 CONTINUE                                                          QUAS297
C                                                                       QUAS297
C***     ***NOW NNEW = KORE - (M*KF) + 1                                QUAS298
C***     ***NOW NOLD = KORE - (M - 1) * N + 1 - KF                      QUAS299
C   ***    SKIP 1ST PART OF TRAPEZOIDAL MATRIX + READ LAST K ROWS       QUAS300
C*** ***ATTATCH RHS TO T1 SO THAT EVERYTHING IS IN CONSECUTIVE ORDER    QUAS301
      NREMAN = ND - K                                                   QUAS302
      IF(NREMAN .EQ. 0)GO TO 280                                        QUAS303
      DO 270 I = 1,NREMAN                                               QUAS304
  270 READ(MT) IDUMMY                                                   QUAS305
  280 NEND = 0                                                          QUAS306
      KCNT = K                                                          QUAS307
      NNEW = NNEW - 1                                                   QUAS308
C***     ***NOTE THAT K = KF WHICH IS ALREADY KNOWN IN CORE             QUAS309
      DO 290 JCNT = 1,K                                                 QUAS310
      NBEG = NEND + 1                                                   QUAS311
      CALL  GETT(MT, 4, KCNT, A(NBEG), 1, AA2)                          QUAS312
      KCNT = KCNT - 1                                                   QUAS313
      NEND = NBEG + KCNT                                                QUAS314
      NNEW = NNEW + 1                                                   QUAS315
      KEND = (MM1 * KF) + NNEW                                          QUAS316
      DO 290 NPP=NNEW,KEND,KF                                           QUAS317
      NEND = NEND + 1                                                   QUAS318
  290 A(NEND) = A(NPP)                                                  QUAS319
      REWIND LTAPE                                                      QUAS320
      REWIND MT                                                         QUAS321
C                                                                       QUAS322
C - - THERE, NOW WE CAN START THE BACK-SOLUTION                          QUAS323
C * * NOTE..THE FIRST AVAILABLE LOCATION FOR THE SOLUTIONS IS A(N1)     QUAS324
C                                                                       QUAS325
C                                                                       QUAS326
C***     ***NL IS THE LAST SUBSCRIPT + 1 OF THE TRAPEZOIDAL A MATRIX THAT QUAS327
C***     ***CORE                                                        QUAS328
C                                                                       QUAS329
      NL = NEND + 1                                                     QUAS330
      NREM = N                                                          QUAS331
      NPM = N + M                                                       QUAS332
      NEL = NPM                                                         QUAS333
      MP1 = M + 1                                                       QUAS334
      LAST = K .EQ. N                                                   QUAS335
      NPASS = 0                                                         QUAS336
C                                                                       QUAS337
C - - SOLVE FOR THE ANSWERS CORRESPONDING TO 'K' ROWS                    QUAS338
C                                                                       QUAS339
  300 KM1 = K - 1                                                       QUAS340
```

171

```
      MP1 = M + 1                                                       QUAS341
      NS = NL - MP1                                                     QUAS342
      NPASS = NPASS + 1                                                 QUAS343
      DO 330 MN = 1, M                                                  QUAS344
      NF = NS + MN                                                      QUAS345
      A(NF) = A(NF) / A(NS)                                             QUAS346
      NT = NS                                                           QUAS347
      IF (KM1 .EQ. 0) GO  TO 330                                        QUAS348
      DO 320 IB = 1, KM1                                                QUAS349
      NF = NF - IB - M                                                  QUAS350
      NT = NT - MP1 - IB                                                QUAS351
      SUM = 0.0                                                         QUAS352
      NP = NF                                                           QUAS353
      N2 = MP1 + IB                                                     QUAS354
      DO 310 IO = 1, IB                                                 QUAS355
      NN = NT + IO                                                      QUAS356
      NP = NP + N2 - IO                                                 QUAS357
  310 SUM = SUM + A(NN) * A(NP)                                         QUAS358
  320 A(NF) = (A(NF) - SUM) / A(NT)                                     QUAS359
  330 CONTINUE                                                          QUAS360
C                                                                       QUAS361
C - - MOVE THE SOLUTIONS TO CONTIGUOUS LOCATIONS STARTING AT A(N1)      QUAS362
C                                                                       QUAS363
      N1 = KORE + 1                                                     QUAS364
      DO 350 NN = 1, K                                                  QUAS365
      DO 340 MN = 1, M                                                  QUAS366
      NL = NL - 1                                                       QUAS367
      N1 = N1 - 1                                                       QUAS368
  340 A(N1) = A(NL)                                                     QUAS369
  350 NL = NL - NN                                                      QUAS370
C                                                                       QUAS371
C - - WRITE THE SOLUTIONS ON TAPE                                       QUAS372
C                                                                       QUAS373
      WRITE (NIN) K                                                     QUAS374
      NS = N1 - 1                                                       QUAS375
      DO 360 MN = 1, M                                                  QUAS376
      NT = NS + MN                                                      QUAS377
  360 WRITE (NIN ) (A(IO), IO = NT, KORE, M)                            QUAS378
C                                                                       QUAS379
C - - TEST IF THIS IS THE LAST PASS                                     QUAS380
C                                                                       QUAS381
      IF (LAST) GO TO 470                                               QUAS382
C                                                                       QUAS383
C - - WE MUST NOW MODIFY THE TRIANGULAR MATRIX  TO REFLECT THE EFFECT OF QUAS384
C     THE SOLUTIONS OBTAINED SO FAR  (EQ 21)                            QUAS385
C * * NOTE..LOCATIONS A(1) TO A(N1-1) ARE NOW FREE TO USE               QUAS386
C                                                                       QUAS387
C - - CALCULATE THE NEXT VALUES OF 'NEL' AND 'NREM'                     QUAS388
C                                                                       QUAS389
      NELOLD = NEL                                                      QUAS390
      KOLD = K                                                          QUAS391
      NEL = NEL - K                                                     QUAS392
      NREM = NREM - K                                                   QUAS393
C                                                                       QUAS394
      NROW = NREM - K + 1                                               QUAS395
      IF (K .LT. NREM) GO TO 370                                        QUAS396
      LAST = .TRUE.                                                     QUAS397
```

```
            NROW = 1                                                    QUAS398
            K = NREM                                                    QUAS399
      370   NS = 1                                                      QUAS400
            NT = NELOLD + 1                                             QUAS401
      C                                                                 QUAS402
      C - - READ IN THE ROWS TO BE MODIFIED                             QUAS403
      C                                                                 QUAS404
            DO 450 IB = 1, NREM                                         QUAS405
            NT = NT - 1                                                 QUAS406
            IF (IB .LE. NROW) GO TO 380                                 QUAS407
            NS = NS + NN                                                QUAS408
            NT = NT + NN                                                QUAS409
      380   IF (.NOT. JPASS1)GO TO 390                                  QUAS410
            NBEG = NT - M + 1                                           QUAS411
      C***  ***READ RHS FROM NATAPE                                     QUAS412
            CALL  GETT(NATAPE, 1, M, A(NBEG), 1, AA2)                   QUAS413
            NT = NT - M                                                 QUAS414
      390   CALL  GETT(MT, 2, NN, A(NS), 1, AA2)                        QUAS415
            IF( .NOT. JPASS1 ) GO TO 400                                QUAS416
            NT = NT + M                                                 QUAS417
            NN = NN + M                                                 QUAS418
      400   NP = N1 - 1                                                 QUAS419
            NF = NT - M - KM1                                           QUAS420
            NN = NN - KOLD                                              QUAS421
            DO 420 MN = 1, M                                            QUAS422
            N2 = NF                                                     QUAS423
            NA = NP + MN                                                QUAS424
            NB = NA                                                     QUAS425
            SUM = 0.0                                                   QUAS426
            DO 410 IO = 1, KOLD                                         QUAS427
            SUM = SUM + A(N2) * A(NA)                                   QUAS428
            N2 = N2 + 1                                                 QUAS429
      410   NA = NA + M                                                 QUAS430
            N2 = N2 + MN - 1                                            QUAS431
      420   A(N2) = A(N2) - SUM                                         QUAS432
      C                                                                 QUAS433
      C - - WRITE THE MODIFIED ROW ON TAPE OR CONDENSE THE ROW          QUAS434
      C                                                                 QUAS435
            NL = NT - M + 1                                             QUAS436
            IF (IB .GE. NROW) GO TO 430                                 QUAS437
            NF = NL - KP1                                               QUAS438
            NN1 = NF - NS + 1                                           QUAS439
            NN2 = NT - NL + 1                                           QUAS440
            CALL  SAVE(NOUT, 4, NN, NN1,      A(NS), NN2, A(NL))        QUAS441
            GO TO 450                                                   QUAS442
      430   NF = NL - KOLD                                              QUAS443
            DO 440 MN = NL, NT                                          QUAS444
            A(NF) = A(MN)                                               QUAS445
      440   NF = NF + 1                                                 QUAS446
      450   CONTINUE                                                    QUAS447
      C***  ***IF 1ST TIME THRU BACK SOLN,SWITCH TAPES SO THAT MT WHICH HAS THEQUAS448
      C***  ***ORIGINAL TRAPEZOIDAL MATRIX ON IT BECOME NATAPE AND IS NOT TO    QUAS449
      C***  ***TAPE PART IN ALTERNATING SHRINKING MATRICES.  NATAPE BECOMES MT  QUAS450
      C***  ***AND THIS NOW DOES THE ALTERNATING WITH NOUT.             QUAS451
            IF( .NOT. JPASS1 ) GO TO 460                                QUAS452
            NTEMP = MT                                                  QUAS453
            MT = NATAPE                                                 QUAS454
```

```
      NATAPE = NTEMP                                                    QUAS455
      JPASS1= .FALSE.                                                   QUAS456
      REWIND NATAPE                                                     QUAS457
  460 REWIND MT                                                         QUAS458
      REWIND NOUT                                                       QUAS459
C                                                                       QUAS460
C - - SWITCH THE TAPES                                                   QUAS461
C                                                                       QUAS462
      NT = MT                                                           QUAS463
      MT = NOUT                                                         QUAS464
      NOUT = NT                                                         QUAS465
C                                                                       QUAS466
C - - LOOP BACK THRU THE SOLUTION                                       QUAS467
C                                                                       QUAS468
      NL = NF                                                           QUAS469
      GO TO 300                                                         QUAS470
C                                                                       QUAS471
C - - START TO WRAP IT UP                                               QUAS472
C                                                                       QUAS473
  470 REWIND NIN                                                        QUAS474
      N2 = N                                                            QUAS475
C                                                                       QUAS476
C * * NOTE.. AT THIS POINT ALL LOCATIONS A(I) THRU A(KORE) ARE FREE     QUAS477
C                                                                       QUAS478
      DO 490 IB = 1, NPASS                                              QUAS479
      READ (NIN) K                                                      QUAS480
      N1 = N2 - K + 1                                                   QUAS481
      NS = N1                                                           QUAS482
      NT = N2                                                           QUAS483
C                                                                       QUAS484
C - - READ IN THE SOLUTIONS                                             QUAS485
C                                                                       QUAS486
      DO 480 IO = 1, M                                                  QUAS487
      CALL  GETT(NIN, 1, K, A(NS), 1, AA2)                              QUAS488
      NT = NT + N                                                       QUAS489
  480 NS = NS + N                                                       QUAS490
  490 N2 = N1 - 1                                                       QUAS491
C                                                                       QUAS492
C  ---  REWIND ALL INPUT TAPES                                          QUAS493
      REWIND NIN                                                        QUAS494
      REWIND MT                                                         QUAS495
      REWIND NOUT                                                       QUAS496
C - - WRITE THE SOLUTIONS ON TAPE                                       QUAS497
C                                                                       QUAS498
      NT = 0                                                            QUAS499
      DO 500 IO = 1, M                                                  QUAS500
      NS = NT + 1                                                       QUAS501
      NT = NT + N                                                       QUAS502
  500 CALL  SAVE(NW, 1, N, N, A(NS), 1, AA2)                            QUAS503
C   ***   IF TAPE WAS NEVER SWITCHED IT WOULD BE FOOLISH TO SWITCH BACK QUAS504
      IF(JPASS1)GO TO 510                                               QUAS505
C                                                                       QUAS506
C***    ***SWITCH TAPES                                                 QUAS507
C*** ***BACK SO THAT  MT WILL CONAIN THE TRAPEZOIDAL MATRIX             QUAS508
C*** ***NATAPE WILL HAVE NOTHING USEFUL ON IT.                          QUAS509
      NTEMP = NATAPE                                                    QUAS510
      NATAPE = MT                                                       QUAS511
```

```
      MT = NTEMP                                                        QUAS512
      REWIND NAIAPE                                                     QUAS513
  510 IF( .NOT. LASTRS)GC TO 150                                        QUAS514
  520 REWIND LTAPE                                                      QUAS515
      REWIND MT                                                         QUAS516
      KRED = 0                                                          QUAS517
      DO 540 I=1,NLCNT                                                  QUAS518
      READ(LTAPE)KREAD                                                  QUAS519
  530 CONTINUE                                                          QUAS520
      KRED = KRED + KREAD                                               QUAS521
      KREAD = KREAD + ( N - KRED - 1)                                   QUAS522
      DO 540 LREAD=1,KREAD                                              QUAS523
  540 READ(LTAPE)                                                       QUAS524
      DO 550 NROW = 1,ND                                                QUAS525
      CALL  GETT(MT, 2, ICNT, A(1), 1, AA2)                             QUAS526
  550 CALL  SAVE(LTAPE, ,, ICNT, ICNT, A, 1, AA2)                       QUAS527
C                                                                       QUAS528
C   ***   REWIND ALL FILES EXCEPT THE CUTPUT FILE    NW                 QUAS529
      REWIND LTAPE                                                      QUAS530
      REWIND  NI                                                        QUAS531
      REWIND  MM                                                        QUAS532
      REWIND  NO                                                        QUAS533
      REWIND  NAT                                                       QUAS534
      IF( RHSTAP .NE. 0 ) REWIND RHSTAP                                 QUAS535
      CALL TIMEV(AA2)                                                   QUAS536
      MD = MTOTAL                                                       QUAS537
      BB = (AA2 - AA1) / 60.                                            QUAS538
      WRITE(6,560)N,N,MTOTAL,BB                                         QUAS539
  560 FORMAT (4HOTHE I5, 2H X I5, 12H MATRIX WITH I4, 35H RIGHT SIDES WAQUAS540
     1S SOLVED DIRECTLY IN F8.3, 9H MINUTES. )                          QUAS541
  570 CONTINUE                                                          QUAS542
      RETURN                                                            QUAS543
      END                                                               QUAS544
```

```
      FUNCTION   RMAX (NL, X, Y, TMAX)                                 RMAX001
C                                                                      RMAX002
      DIMENSION     X(NL), Y(NL)                                       RMAX003
C                                                                      RMAX004
      ID = 1                                                           RMAX005
      I1 = NL/2                                                        RMAX006
      I3 = I1 + ID                                                     RMAX007
      R1 = SQRT((X(I1)-X(1))**2 + (Y(I1)-Y(1))**2)                     RMAX008
      R2 = SQRT((X(I3)-X(1))**2 + (Y(I3)-Y(1))**2)                     RMAX009
      IF (R2 .GT. R1)  GO TO 20                                        RMAX010
C                                                                      RMAX011
      R3 = R1                                                          RMAX012
      I3 = I1                                                          RMAX013
      ID = -ID                                                         RMAX014
C                                                                      RMAX015
   10 R1 = R2                                                          RMAX016
      R2 = R3                                                          RMAX017
   20 I3 = I3 + ID                                                     RMAX018
      R3 = SQRT((X(I3)-X(1))**2 + (Y(I3)-Y(1))**2)                     RMAX019
      IF (R3 .GT. R2) GO TO 10                                         RMAX020
C                                                                      RMAX021
C  CALCULATE ANGLES                                                    RMAX022
      I2 = I3 - ID                                                     RMAX023
      I1 = I2 - ID                                                     RMAX024
      T1 = ARSIN((Y(I1)-Y(1))/R1)                                      RMAX025
      T2 = ARSIN((Y(I2)-Y(1))/R2)                                      RMAX026
      T3 = ARSIN((Y(I3)-Y(1))/R3)                                      RMAX027
C                                                                      RMAX028
C  CALCULATE MAXIMUM RADIUS (CHORD)                                    RMAX029
      T2 = T2 - T1                                                     RMAX030
      T3 = T3 - T1                                                     RMAX031
      A3 = T2*T3*(T3 - T2)                                             RMAX032
      R2 = R2 - R1                                                     RMAX033
      R3 = R3 - R1                                                     RMAX034
      A1 = (R3*T2 - R2*T3)/A3                                          RMAX035
      A2 = (R2*T3**2 - R3*T2**2)/A3                                    RMAX036
C                                                                      RMAX037
C                                                                      RMAX038
      RMAX = -0.25*A2**2/A1 + R1                                       RMAX039
      TMAX = -0.5*A2/A1 + T1                                           RMAX040
C                                                                      RMAX041
      RETURN                                                           RMAX042
      END                                                              RMAX043
```

```
      SUBROUTINE  SAVE(IL, II, N, N1, A1, N2, A2)                      SAVE001
C                                                                      SAVE002
      DIMENSION  A1(N1), A2(N2)                                        SAVE003
C                                                                      SAVE004
      GO TO (10,20,30,40), II                                          SAVE005
C                                                                      SAVE006
C  WRITE A1                                                            SAVE007
   10 WRITE(IU) A1                                                     SAVE008
      RETURN                                                           SAVE009
C                                                                      SAVE010
C  WRITE N AND A1                                                      SAVE011
   20 WRITE(IU)  N, A1                                                 SAVE012
      RETURN                                                           SAVE013
C                                                                      SAVE014
C  WRITE A1 AND A2                                                     SAVE015
   30 WRITE(IU) A1, A2                                                 SAVE016
      RETURN                                                           SAVE017
C                                                                      SAVE018
C  WRITE N, A1, AND A2                                                 SAVE019
   40 WRITE(IU) N, A1, A2                                              SAVE020
      RETURN                                                           SAVE021
      END                                                              SAVE022
```

```
      SUBROUTINE  SOLVE(N, M, ISIZE, ISOL)                              SOLV001
C                                                                       SOLV002
C                                                                       SOLV003
C THIS ROUTINE OBTAINS THE SIGMA SOLUTIONS                              SOLV004
C FROM EITHER QUASI (OR FUTSOL)                                         SOLV005
C                                                                       SOLV006
C WKAREA MUST BE DIMENSIOND EQUAL TO IMAX.                              SOLV007
C TO CHANGE THIS, BOTH MUST BE COMPILED.                                SOLV008
C THE CURRENT ARRAY SIZE IS GIVEN BY                                    SOLV009
C ISIZE, AN INPUT VARIABLE.                                             SOLV010
C                                                                       SOLV011
C                                                                       SOLV012
      COMMON /SPACER/ WKAREA(11913)                                     SOLV013
C                                                                       SOLV014
      DIMENSION  A(101,101), SIG(101,12)                                SOLV015
C                                                                       SOLV016
      EQUIVALENCE  (A(1,1),WKAREA(1)), (SIG(1,1), WKAREA(10202))        SOLV017
C                                                                       SOLV018
      COMMON /FILEID/ IFILE1, IFILE2, IFILE3, IFILE4, IFILE5,           SOLV019
     1                 IFILE6, IFILE7, IFILE8, IFILE9, IFIL10,          SOLV020
     2                 IFIL11, IFIL12, IFIL13, IFIL14, IFIL15           SOLV021
     3                ,IFIL16, IFIL17, IFIL18, IFIL19, IFIL20           SOLV022
C                                                                       SOLV023
C                                                                       SOLV024
      IF (ISOL .EQ. 2)  GO TO 70                                        SOLV025
      MM = M                                                            SOLV026
      CALL TIMEV(T)                                                     SOLV027
   10 IF (ISOL .NE. 0)  GO TO 30                                        SOLV028
C                                                                       SOLV029
   15 CONTINUE                                                          SOLV030
C                                                                       SOLV031
      WRITE(6,20) T                                                     SOLV032
   20 FORMAT(1H0, 'SOLVII TIME = ', F9.3, ' SECONDS.')                  SOLV033
C                                                                       SOLV034
      CALL SOLVII (WKAREA, N, MM, ISIZE, IFILE9, IFILE1,                SOLV035
     1            IFILE2, IFIL14, &50)                                  SOLV036
      CALL TIMEV(T)                                                     SOLV037
      WRITE(6,20) T                                                     SOLV038
      RETURN                                                            SOLV039
C                                                                       SOLV040
C                                                                       SOLV041
   30 WRITE(6,40) T                                                     SOLV042
      CALL  QUASI(WKAREA, N,MM, ISIZE, IFILE9, IFILE1, IFILE2,          SOLV043
     1           IFILE3, IFIL14, IFIL15, IFILE4,&50)                    SOLV044
      CALL TIMEV(T)                                                     SOLV045
      WRITE(6,40) T                                                     SOLV046
   40 FORMAT(1H0, 'QUASI TIME = ', F9.3,'SECONDS.')                     SOLV047
      RETURN                                                            SOLV048
C                                                                       SOLV049
   50 WRITE (6,60) ISIZE, IMAX                                          SOLV050
   60 FORMAT (1H1, 'ISIZE = ',I5,' TOO SMALL.  SET EQUAL TO IMAX (',    SOLV051
     1        I5, ')')                                                  SOLV052
C                                                                       SOLV053
C                                                                       SOLV054
      IF (ISIZE .GE. IMAX)  GO TO 140                                   SOLV055
      ISIZE = IMAX                                                      SOLV056
```

```
      GO TO 10                                                         SOLV057
C                                                                      SOLV058
C                                                                      SOLV059
C                                                                      SOLV060
C  THIS IS A DUMMY ROUTINE TO USE MIS1 (MIS2)                          SOLV061
C  FOR SOLVING THE VORTEX DISTRIBUTIONS                                SOLV062
C                                                                      SOLV063
C  A-ARRAY STORED ON IFIL10 (THE FIRST OF TWO RECORDS)                 SOLV064
C                                                                      SOLV065
C  RHS STORED ON IFILE4                                                SOLV066
C                                                                      SOLV067
C  SIGMAS SAVED ON IF14                                                SOLV068
C                                                                      SOLV069
C                                                                      SOLV070
   70 NMAX = 101                                                       SOLV071
      MD = 12                                                          SOLV071
      IF14=  IFIL14                                                    SOLV072
      REWIND IFIL10                                                    SOLV073
      REWIND IFILE4                                                    SOLV074
      REWIND IF14                                                      SOLV075
      IF (N .LE. NMAX)  GO TO 90                                       SOLV076
      WRITE(6,80)  N, NMAX                                             SOLV077
   80 FORMAT(1H0,'THE SIZE OF ARRAY (',I5,') EXCEEDS LIMIT OF ', I4)   SOLV078
      GO TO 15                                                         SOLV079
C                                                                      SOLV080
   90 CONTINUE                                                         SOLV081
C                                                                      SOLV082
      D = 1.0                                                          SOLV083
C                                                                      SOLV084
C  READ IN A-ARRAY                                                     SOLV085
      DO 100 I = 1,N                                                   SOLV086
      READ(IFIL10)  (A(I,J),J=1,N)                                     SOLV087
  100 READ (IFIL10)                                                    SOLV088
C                                                                      SOLV089
C  READ IN RHS IN SIG ARRAY                                            SOLV090
      READ ( IFILE4 ) MRHS                                             SOLV090
      DO 110 K = 1,M                                                   SOLV091
  110 READ(IFILE4)  (SIG(I,K),I=1,N)                                   SOLV092
C                                                                      SOLV093
C                                                                      SOLV094
      CALL MIS2(A, N, NMAX, SIG, M, MD, NERR, D)                       SOLV095
C                                                                      SOLV096
      WRITE (6,120)  NERR                                              SOLV097
  120 FORMAT(1H0,'ON RETURN FROM MIS2, NERR = ', I2)                   SOLV098
C                                                                      SOLV099
C  SAVE SIGMAS                                                         SOLV100
      DO 130 J = 1,M                                                   SOLV101
  130 WRITE(IF14)  (SIG(I,J), I=1,N)                                   SOLV102
      RETURN                                                           SOLV103
C                                                                      SOLV104
C                                                                      SOLV105
  140 STOP                                                             SOLV106
      END                                                              SOLV107
```

```
      SUBROUTINE  SOLVIT (A, KD, MD, KC, NI, MM, NO, NW, *)              SLVT001
C                                                                        SLVT002
C                                                                        SLVT003
C            ****    ***/  *******  *****  *        ***/        *        SLVT004
C            *   *  *  /*     *      *     *       *  /*     ******      SLVT005
C     ****   ****   * / *     *      ***   *       * / *  ****   *  *    SLVT006
C            *      */  *     *      *     *       */  *         ***     SLVT007
C            *      /***      *      *     *****   /***          *  *    SLVT008
C                                                                        SLVT009
C              D I R E C T   M A T R I X   S O L U T I O N               SLVT010
C                                                                        SLVT011
C     WRITTEN BY J. L. HESS * PROGRAMMED BY T. M. RIDDELL                SLVT012
C                                                                        SLVT013
      DIMENSION  A ( KD )                                                SLVT014
C                                                                        SLVT015
      LOGICAL  LAST                                                      SLVT016
C                                                                        SLVT017
      CALL TIMEV(AA1)                                                    SLVT018
       IF( AA1 .EQ. 0. )        CALL TSETV                               SLVT019
      N = ND                                                             SLVT020
      M = MD                                                             SLVT021
      KORE = KD                                                          SLVT022
      NPM = N + M                                                        SLVT023
      IF (MAX0(3 * NPM, N * N) .GT. KORE) RETURN 1                       SLVT024
      MT = MM                                                            SLVT025
      REWIND MT                                                          SLVT026
      NIN = NI                                                           SLVT027
      REWIND NIN                                                         SLVT028
      NOUT = NO                                                          SLVT029
      REWIND NOUT                                                        SLVT030
      MP1 = M + 1                                                        SLVT031
      NN = N                                                             SLVT032
      NEL = NPM                                                          SLVT033
C                                                                        SLVT034
C - - CALCULATE THE MAXIMUM NO. OF ROWS, 'K'                             SLVT035
C                                                                        SLVT036
   10 K = (KORE - NEL) / NEL                                             SLVT037
C                                                                        SLVT038
C - - TEST TO SEE IF THE REST OF THE MATRIX WILL FIT IN CORE             SLVT039
C                                                                        SLVT040
      LAST = K .GE. NN                                                   SLVT041
      IF (LAST) K = NN                                                   SLVT042
C                                                                        SLVT043
C - - READ 'K' ROWS OF THE AUGMENTED 'A' MATRIX                          SLVT044
C                                                                        SLVT045
   20 NT = 0                                                             SLVT046
      DO 30 IB = 1, K                                                    SLVT047
      NS = NT + 1                                                        SLVT048
      NT = NT + NEL                                                      SLVT049
   30 CALL  GETT(NIN, 1, NEL, A(NS), 1, AA2)                             SLVT050
C                                                                        SLVT051
C - - CHECK TO SEE IF WE WERE UNLUCKY ENOUGH TO END UP WITH ONLY ONE ROWSLVT052
C                                                                        SLVT053
      IF (K .EQ. 1) GO TO 90                                             SLVT054
C                                                                        SLVT055
C   - 'K' IS GREATER THAN '1' SO WE CAN START THE TRIANGULARIZATION      SLVT056
```

```
C                                                                       SLVT057
      NELP1 = NEL + 1                                                   SLVT058
      NS = - NEL                                                        SLVT059
      NELP2 = NELP1 + 1                                                 SLVT060
C                                                                       SLVT061
C - - FORM THE 'TRAPEZOIDAL' ARRAY  (B)                                  SLVT062
C                                                                       SLVT063
      DO 90 IB = 2, K                                                   SLVT064
      NP = NELP2 - IB                                                   SLVT065
      NS = NS + NELP1                                                   SLVT066
      NT = NS                                                           SLVT067
      DO 90 IO = IB, K                                                  SLVT068
      NT = NT + NEL                                                     SLVT069
      MN = NT                                                           SLVT070
      NB = NS                                                           SLVT071
      A(NT) = (-A(NT)) / A(NS)                                          SLVT072
      DO 90 NF = 2, NP                                                  SLVT073
      MN = MN + 1                                                       SLVT074
      NB = NB + 1                                                       SLVT075
   90 A(MN) = A(MN) + A(NT) * A(NB)                                     SLVT076
      IF (LAST) GO TO 90                                                SLVT077
C                                                                       SLVT078
C - - WRITE THE 'TRAPEZOIDAL' MATRIX ON TAPE                             SLVT079
C                                                                       SLVT080
      NT = 0                                                            SLVT081
      NP = NEL                                                          SLVT082
      NS = - NEL                                                        SLVT083
      DO 50 IO = 1, K                                                   SLVT084
      NS = NS + NELP1                                                   SLVT085
      NT = NT + NEL                                                     SLVT086
      CALL  SAVE(NT, 2, NP, NP, A(NS), 1, AA2)                          SLVT087
   50 NP = NP - 1                                                       SLVT088
      NP = NP - M                                                       SLVT089
      NS = KORE - NEL + 1                                               SLVT090
C                                                                       SLVT091
C - - READ ANOTHER ROW                                                   SLVT092
C                                                                       SLVT093
      DO 80 IO = 1, NP                                                  SLVT094
      CALL  GETT(NIN, 1, NEL, A(NS), 1, AA2)                            SLVT095
C                                                                       SLVT096
C - - MODIFY THIS ROW BY THE 'TRAPEZOICAL' ARRAY                         SLVT097
C                                                                       SLVT098
      NT = 1                                                            SLVT099
      MN = NS                                                           SLVT100
      DO 70 IB = 1, K                                                   SLVT101
      NB = NT                                                           SLVT102
      NF = MN + 1                                                       SLVT103
      A(MN) = (-A(MN)) / A(NT)                                          SLVT104
      DO 60 NN = NF, KORE                                               SLVT105
      NB = NB + 1                                                       SLVT106
   60 A(NN) = A(NN) + A(MN) * A(NB)                                     SLVT107
      MN = NF                                                           SLVT108
   70 NT = NT + NELP1                                                   SLVT109
C                                                                       SLVT110
C - - WRITE THE MODIFIED ROW ON TAPE                                     SLVT111
C                                                                       SLVT112
      NN1 = KORE - MN + 1                                               SLVT113
```

```
   80 CALL  SAVE(NOUT, 1, NN1, NN1, A(PN), 1, A12)                     SLVT114
      REWIND NOUT                                                      SLVT115
      REWIND NIN                                                       SLVT116
C                                                                      SLVT117
C - - SWITCH THE TAPES                                                 SLVT118
C                                                                      SLVT119
      NT = NIN                                                         SLVT120
      NIN = NOUT                                                       SLVT121
      NOUT = NT                                                        SLVT122
C                                                                      SLVT123
C - - RE-CALCULATE ROW LENGTH AND LOOP BACK                            SLVT124
C                                                                      SLVT125
      NEL = NEL - K                                                    SLVT126
      NN = NEL - M                                                     SLVT127
      GO TO 10                                                         SLVT128
C                                                                      SLVT129
C - - REWIND ALL TAPES                                                 SLVT130
C                                                                      SLVT131
   90 REWIND NT                                                        SLVT132
      REWIND NIN                                                       SLVT133
      REWIND NOUT                                                      SLVT134
C                                                                      SLVT135
C - - CONDENSE THE MATRIX                                              SLVT136
C                                                                      SLVT137
      NN = NEL                                                         SLVT138
      NL = NEL + 1                                                     SLVT139
      IF (K .EQ. 1) GO TO 110                                          SLVT140
      NS = 1                                                           SLVT141
      NT = NEL                                                         SLVT142
      DO 100 IB = 2, K                                                 SLVT143
      NS = NS + NELP1                                                  SLVT144
      NT = NT + NEL                                                    SLVT145
      DO 100 IO = NS, NT                                               SLVT146
      A(NL) = A(IO)                                                    SLVT147
  100 NL = NL + 1                                                      SLVT148
  110 N1 = KORE - K * M + 1                                            SLVT149
C                                                                      SLVT150
C - - THERE, NOW WE CAN START THE BACK-SOLUTION                        SLVT151
C * * NOTE..THE FIRST AVAILABLE LOCATION FOR THE SOLUTIONS IS A(N1)    SLVT152
C                                                                      SLVT153
      NREM = N                                                         SLVT154
      NEL = NPM                                                        SLVT155
      LAST = K .EQ. N                                                  SLVT156
      NPASS = 0                                                        SLVT157
C                                                                      SLVT158
C - - SOLVE FOR THE ANSWERS CORRESPONDING TO 'K' ROWS                  SLVT159
C                                                                      SLVT160
  120 KM1 = K - 1                                                      SLVT161
      KP1 = K + 1                                                      SLVT162
      NS = NL - MP1                                                    SLVT163
      NPASS = NPASS + 1                                                SLVT164
      DO 150 MN = 1, M                                                 SLVT165
      NF = NS + MN                                                     SLVT166
      A(NF) = A(NF) / A(NS)                                            SLVT167
      NT = NS                                                          SLVT168
      IF (KM1 .EQ. 0) GO  TO 150                                       SLVT169
      DO 140 IB = 1, KM1                                               SLVT170
```

```
      NF = NF - IB - M                                                  SLVT171
      NT = NT - MP1 - IB                                                SLVT172
      SUM = 0.0                                                         SLVT173
      NP = NF                                                           SLVT174
      N2 = MP1 + IB                                                     SLVT175
      DO 130 IO = 1, IB                                                 SLVT176
      NN = NT + IO                                                      SLVT177
      NP = NP + N2 - IO                                                 SLVT178
  130 SUM = SUM + A(NN) * A(NP)                                         SLVT179
  140 A(NF) = (A(NF) - SUM) / A(NT)                                     SLVT180
  150 CONTINUE                                                          SLVT181
C                                                                       SLVT182
C - - MOVE THE SOLUTIONS TO CONTIGUOUS LOCATIONS STARTING AT A(N1)      SLVT183
C                                                                       SLVT184
      N1 = KORE + 1                                                     SLVT185
      DO 170 NN = 1, K                                                  SLVT186
      DO 160 MN = 1, M                                                  SLVT187
      NL = NL - 1                                                       SLVT188
      N1 = N1 - 1                                                       SLVT189
  160 A(N1) = A(NL)                                                     SLVT190
  170 NL = NL - NN                                                      SLVT191
C                                                                       SLVT192
C - - WRITE THE SOLUTIONS ON TAPE                                       SLVT193
C                                                                       SLVT194
      WRITE (NIN) K                                                     SLVT195
      NS = N1 - 1                                                       SLVT196
      DO 180 MN = 1, M                                                  SLVT197
      NT = NS + MN                                                      SLVT198
  180 WRITE (NIN ) (A(IO), IO = NT, KORE, M)                            SLVT199
C                                                                       SLVT200
C - - TEST IF THIS IS THE LAST PASS                                     SLVT201
C                                                                       SLVT202
      IF (LAST) GO TO 260                                               SLVT203
C                                                                       SLVT204
C - - WE MUST NOW MODIFY THE TRIANGULAR MATRIX TO REFLECT THE EFFECT OF SLVT205
C     THE SOLUTIONS OBTAINED SO FAR  (EQ 21)                            SLVT206
C * * NOTE..LOCATIONS A(1) TO A(N1-1) ARE NOW FREE TO USE               SLVT207
C                                                                       SLVT208
C - - CALCULATE THE NEXT VALUES OF 'NEL' AND 'NREM'                     SLVT209
C                                                                       SLVT210
      NELOLD = NEL                                                      SLVT211
      KOLD = K                                                          SLVT212
      NEL = NEL - K                                                     SLVT213
      NREM = NREM - K                                                   SLVT214
C                                                                       SLVT215
C$$$$ CALCULATE NEW K.  B AND C (REAL) WILL ALWAYS BE INTEGERS.         SLVT216
C     K WILL BE CALCULATED REAL AND TRUNCATED - - GOOD.                 SLVT217
C                                                                       SLVT218
      B = 1 + 2*M                                                       SLVT219
      C = 2*(KOLD*(M+1) - KORE)                                         SLVT220
      K = (-B + SQRT(B**2 - 4*C))/2.0                                   SLVT221
      NROW = NREM - K + 1                                               SLVT222
      IF (K .LT. NREM) GO TO 190                                        SLVT223
      LAST = .TRUE.                                                     SLVT224
      NROW = 1                                                          SLVT225
      K = NREM                                                          SLVT226
  190 NS = 1                                                            SLVT227
```

```
      NT = NELOLD + 1                                                   SLVT228
C                                                                       SLVT229
C - - READ IN THE ROWS TO BE MODIFIED                                    SLVT230
C                                                                       SLVT231
      DO 250 IB = 1, NREP                                               SLVT232
      NT = NT - 1                                                       SLVT233
      IF (IB .LE. NROW) GO TO 200                                       SLVT234
      NS = NS + NN                                                      SLVT235
      NT = NT + NN                                                      SLVT236
  200 CALL  GETT(MT, 2, NN, A(NS), 1, AA2)                              SLVT237
      NP = N1 - 1                                                       SLVT238
      NF = NT - M - KM1                                                 SLVT239
      NN = NN - KOLD                                                    SLVT240
      DO 220 MN = 1, M                                                  SLVT241
      N2 = NF                                                           SLVT242
      NA = NP + MN                                                      SLVT243
      NB = NA                                                           SLVT244
      SUM = 0.0                                                         SLVT245
      DO 210 IO = 1, KOLD                                               SLVT246
      SUM = SUM + A(N2) * A(NA)                                         SLVT247
      N2 = N2 + 1                                                       SLVT248
  210 NA = NA + M                                                       SLVT249
      N2 = N2 + MN - 1                                                  SLVT250
  220 A(N2) = A(N2) - SUM                                               SLVT251
C                                                                       SLVT252
C - - WRITE THE MODIFIED ROW ON TAPE OR CONDENSE THE ROW                SLVT253
C                                                                       SLVT254
      NL = NT - M + 1                                                   SLVT255
      IF (IB .GE. NROW) GO TO 230                                       SLVT256
      NF = NL - KP1                                                     SLVT257
      NN1 = NF - NS + 1                                                 SLVT258
      NN2 = NT - NL + 1                                                 SLVT259
      CALL  SAVE(NOUT, 4, NN, NN1, A(NS), NN2, A(NL))                   SLVT260
      GO TO 250                                                         SLVT261
  230 NF = NL - KOLD                                                    SLVT262
      DO 240 MN = NL, NT                                                SLVT263
      A(NF) = A(MN)                                                     SLVT264
  240 NF = NF + 1                                                       SLVT265
  250 CONTINUE                                                          SLVT266
      REWIND MT                                                         SLVT267
      REWIND NOUT                                                       SLVT268
C                                                                       SLVT269
C - - SWITCH THE TAPES                                                  SLVT270
C                                                                       SLVT271
      NT = MT                                                           SLVT272
      MT = NOUT                                                         SLVT273
      NOUT = NT                                                         SLVT274
C                                                                       SLVT275
C - - LOOP BACK THRU THE SOLUTION                                       SLVT276
C                                                                       SLVT277
      NL = NF                                                           SLVT278
      GO TO 120                                                         SLVT279
C                                                                       SLVT280
C - - START TO WRAP IT UP                                               SLVT281
C                                                                       SLVT282
  260 REWIND NIN                                                        SLVT283
      N2 = N                                                            SLVT284
C                                                                       SLVT285
C * * NOTE.. AT THIS POINT ALL LOCATIONS A(1) THRU A(KORE) ARE FREE     SLVT286
C                                                                       SLVT287
```

```
      N1 = N2 - N + 1                                                   SLVT290
      NS = N1                                                           SLVT291
      NT = N2                                                           SLVT292
C                                                                       SLVT293
C - - READ IN THE SOLUTIONS                                              SLVT294
C                                                                       SLVT295
      DO 270 IQ = 1, M                                                  SLVT296
      NM = NT - NS + 1                                                  SLVT297
      CALL  GET(IN(N, 1,NM, A(NS), 1, AA2)                              SLVT298
      NT = NT + N                                                       SLVT299
  270 NS = NS + N                                                       SLVT300
  280 N2 = N1 - 1                                                       SLVT301
C                                                                       SLVT302
C - - WRITE THE SOLUTIONS ON TAPE                                       SLVT303
C                                                                       SLVT304
      NT = 0                                                            SLVT305
      DO 290 IQ = 1, M                                                  SLVT306
      NS = NT + 1                                                       SLVT307
      NT = NT + N                                                       SLVT308
  290 CALL  SAVE(NW, 1, N, N, A(NS), 1, AA2)                            SLVT309
C                                                                       SLVT310
      CALL TIMEV(AA2)                                                   SLVT311
      BB = (AA2 - AA1) / 60.                                            SLVT312
      WRITE (6,300) N, N, M, BB                                         SLVT313
  300 FORMAT (4HOTHE I5, 2H X I5, 12H MATRIX WITH I4, 35H RIGHT SIDES WASLVT314
     1S SOLVED DIRECTLY IN F8.3, 9H MINUTES. )                          SLVT315
      RETURN                                                            SLVT316
      END                                                               SLVT317
```

```
      SUBROUTINE TIMEV(T)
      T=0.0
      RETURN
      END
```

```
      SUBROUTINE TSETV
      RETURN
      END
```

```
      SUBROUTINE    TYPE(IGOOD, IBAD)                                   TYPE001
C                                                                       TYPE002
C                                                                       TYPE003
      WRITE(6,10)  IGOOD, IBAD                                          TYPE004
   10 FORMAT (1H1, 'AN ATTEMPT HAS BEEN MADE TO READ A TYPE ',I2,       TYPE005
     1                   ' CARD, HOWEVER A TYPE ',I2, ' CARD WAS FOUND.'/1HO, TYPE006
     2                   'CHECK OVER THE INPUT DATA CARD SEQUENCE FOR ',  TYPE007
     3                   'COMPATIBILITY WITH DESIRED OPTIONS.')         TYPE008
C                                                                       TYPE009
C                                                                       TYPE010
   20 WRITE (6,30)                                                      TYPE011
   30 FORMAT (1HO, 'BECAUSE OF THE ABOVE ERROR, THIS RUN IS TERMINATED')TYPE012
C                                                                       TYPE013
      STOP                                                              TYPE014
      END                                                               TYPE015
```

```
      SUBROUTINE  VXYOFF(N, M, NO, X, Y)                              VXY0001
C                                                                     VXY0002
      DIMENSION  SIG(500), X(1), Y(1), A(500), B(500)                 VXY0003
     1          ,VX(12), VY(12), VXT(12), VYN(12)                     VXY0004
C                                                                     VXY0005
      COMMON/ELD/    X0(500), Y0(500), DS(500), SA(500), CA(500),     VXY0006
     1          CURV(500), DL(500)                                    VXY0007
C                                                                     VXY0008
      COMMON/COMBOD/CCL,INCLT,CLT,ALPHA,SUMDS(10),TLU(10,12),IND       VXY0009
     1              , ALPHAO, CNU(10), SMDSWF(10), MIO(10)            VXY0010
      COMMON /SIGMAS/ CSIG(500), CK(12)                               VXY0011
      COMMON /GCF/ CD(500), CF(500), CG(500), CI(500), WF(500)        VXY0012
C                                                                     VXY0013
      COMMON/MFLAG/ IDB(10), INL(10), IFL(10), NL(10), LIFT(10),      VXY0014
     1              IBMF(10), ISAV1(10), ISAV2(10), ISAV3(10),        VXY0015
     2              BTITLE(10, 7), IBT, IBST, IBTOT, NELTOT,          VXY0016
     3              IIRB(10), IMMB(10), CHORD(10), IBD(10), LIFTOT    VXY0017
     4             ,IPRB(10), IFST(10), ISEC(10), FTITLE(15), IPVR(10) VXY0018
      COMMON/COM/IFLLL
C                                                                     VXY0019
C                                                                     VXY0020
C                                                                     VXY0021
      COMMON /FILEID/  IFA1, IFA2, IF03, IF04, IF05,                  VXY0022
     1                 IF06, IF07, IF08, IF09, IF10,                  VXY0023
     2                 IF01, IF02, IF13, IF14, IF15                   VXY0024
     3                ,IF16, IF17, IF18, IF19, IF20                   VXY0025
      REWIND  IF01                                                    VXY0026
      REWIND  IF02                                                    VXY0027
C                                                                     VXY0028
C                                                                     VXY0029
C  SET SOME QUANTITIES                                                VXY0030
   20 SAI = 0.0                                                       VXY0031
      CAI = 1.0                                                       VXY0032
      DSI = 0.0                                                       VXY0033
C                                                                     VXY0034
      DO 230 I = 1,NO                                                 VXY0035
      JJ = 0                                                          VXY0036
      K = 2                                                           VXY0037
C                                                                     VXY0038
      DO 160 IB = 1,IBTOT                                             VXY0039
      IF (IBMF(IB) .LT. 0)  GO TO 150                                 VXY0040
C                                                                     VXY0041
C  COUNTER FOR ELEMENT GEOMETRY                                       VXY0042
      J = INL(IB) - 1                                                 VXY0043
C  COUNTERS FOR A,B ARRAYS                                            VXY0044
      JI = JJ + 1                                                     VXY0045
      JF = JJ + NL(IB)                                                VXY0046
C                                                                     VXY0047
C  JJ IS COUNTER FOR THE CURRENT ELEMENT                              VXY0048
C                                                                     VXY0049
C  ZERO OUT A&B ARRAYS                                                VXY0050
      DO 30 JJ1 = JI,JF                                               VXY0051
      A(JJ1) = 0.0                                                    VXY0052
   30 B(JJ1) = 0.0                                                    VXY0053
C                                                                     VXY0054
      JJ = JI                                                         VXY0055
```

```
      JJ1 = JJ + 1                                                      VXY0056
      JJ3 = JJ1 + 1                                                     VXY0057
      GO TO 50                                                          VXY0058
   40 JJ3 = JJ + 1                                                      VXY0059
   50 J = J + 1                                                         VXY0060
      A0 = XYFORM(X(I), Y(I), DSI, SAI, CAI, IFST(IB), X0(J), Y0(J),    VXY0062
     1     DL(J), SA(J), CA(J), ISEC(IB), A1, AC, AP, B0, B1, BC, BP)   VXY0063
      B(JJ ) = B(JJ ) + B0                                              VXY0064
      A(JJ) = A(JJ) + A0                                                VXY0065
C                                                                       VXY0066
C                                                                       VXY0067
      IF (IFST(IB) .EQ. 0)  GO TO 80                                    VXY0068
      IF (IFST(IB) - 2)60,70,60                                         VXY0069
C                                                                       VXY0070
C  FIRST TERMS OF FIRST ORDER                                           VXY0071
   60 A(JJ ) = A(JJ ) -        (CF(J) + CD(J))*A1                       VXY0072
      B(JJ ) = B(JJ ) -        (CF(J) + CD(J))*B1                       VXY0073
      A(JJ1) = A(JJ1) +        CD(J)*A1                                 VXY0074
      B(JJ1) = B(JJ1) +        CD(J)*B1                                 VXY0075
      A(JJ3) = A(JJ3) +        CF(J)*A1                                 VXY0076
      B(JJ3) = B(JJ3) +        CF(J)*B1                                 VXY0077
C                                                                       VXY0078
   70 IF (IFST(IB) .LT. 2)  GO TO 80                                    VXY0079
C                                                                       VXY0080
C  CURVATURE TERM OF FIRST ORDER                                        VXY0081
      A(JJ ) = A(JJ ) + DL(J)*CURV(J)*AC                                VXY0082
      B(JJ ) = B(JJ ) + DL(J)*CURV(J)*BC                                VXY0083
C                                                                       VXY0084
   80 IF (ISEC(IB) .EQ. 0)  GO TO 110                                   VXY0085
      IF (ISEC(IB) - 2)90,100,90                                        VXY0086
C                                                                       VXY0087
C  FIRST TERMS OF SECOND ORDER                                          VXY0088
   90 A(JJ ) = A(JJ ) - (CG(J) + CI(J))*AP                              VXY0089
      B(JJ ) = B(JJ ) - (CG(J) + CI(J))*BP                              VXY0090
      A(JJ1) = A(JJ1) + CG(J)*AP                                        VXY0091
      B(JJ1) = B(JJ1) + CG(J)*BP                                        VXY0092
      A(JJ3) = A(JJ3) + CI(J)*AP                                        VXY0093
      B(JJ3) = B(JJ3) + CI(J)*BP                                        VXY0094
C                                                                       VXY0095
  100 IF (ISEC(IB) .LT. 2)  GO TO 110                                   VXY0096
C                                                                       VXY0097
C  CURVATURE TERM OF SECOND ORDER                                       VXY0098
      A(JJ ) = A(JJ ) + 2.0*AP*(DL(J)*CURV(J))**2                       VXY0099
      B(JJ ) = B(JJ ) + 2.0*BP*(DL(J)*CURV(J))**2                       VXY0100
C                                                                       VXY0101
  110 JJ1 = JJ                                                          VXY0102
      JJ = JJ + 1                                                       VXY0103
      IF (JJ - JF)40,120,130                                            VXY0104
C                                                                       VXY0105
  120 JJ3 = JJ1 - 1                                                     VXY0106
      GO TO 50                                                          VXY0107
  130 JJ = JJ - 1                                                       VXY0108
C                                                                       VXY0109
C                                                                       VXY0110
      IF (LIFT(IB) .EQ. 0)  GO TO 150                                   VXY0111
      VN = 0.0                                                          VXY0112
      VT = 0.0                                                          VXY0113
```

```
      DO 140 J = JI,JF                                                  VXY0114
      VN = VN-  B(J)*WF(J)                                              VXY0115
  140 VT = VT + A(J)*WF(J)                                              VXY0116
C                                                                       VXY0117
      K = K + 1                                                         VXY0118
      VXT(K)= VT                                                        VXY0119
      VYN(K)= VN                                                        VXY0120
  150 CONTINUE                                                          VXY0121
C                                                                       VXY0122
C  SET UNIFORM ONSET FLOWS                                              VXY0123
C  ALPHA = 0                                                            VXY0124
      VXT(1)= CAI                                                       VXY0125
      VYN(1)=-SAI                                                       VXY0126
C                                                                       VXY0127
C  ALPHA = 90                                                           VXY0128
      VXT(2)= SAI                                                       VXY0129
      VYN(2)= CAI                                                       VXY0130
C                                                                       VXY0131
C  SET INPUT NON-UNIFORM ONSET FLOWS                                    VXY0132
      M1 = LIFTOT + 2                                                   VXY0133
      IF (M .EQ. M1)  GO TO 170                                         VXY0134
      M2 = M1 + 1                                                       VXY0135
      DO 160 K = M2, M                                                  VXY0136
      VXT(K) = 0.0                                                      VXY0137
  160 VYN(K) = 0.0                                                      VXY0138
  170 CONTINUE                                                          VXY0139
C                                                                       VXY0140
C  CHECK IF INDIVIDUAL FLOWS DESIRED                                    VXY0141
      IF (IND .NE. 1)  GO TO 200                                        VXY0142
      REWIND IF14                                                       VXY0143
C                                                                       VXY0144
C  CALCULATE INDIVIDUAL FLOWS                                           VXY0145
      DO 190 K = 1,M                                                    VXY0146
      CALL GETT(IF14, 1, N, SIG(1), 1, VN)                              VXY0147
      VX(K) = VXT(K)                                                    VXY0148
      VY(K) = VYN(K)                                                    VXY0149
C                                                                       VXY0150
      DO 180 J = 1,N                                                    VXY0151
      VX(K) = VX(K) + B(J)*SIG(J )                                      VXY0152
  180 VY(K) = VY(K) + A(J)*SIG(J )                                      VXY0153
  190 CONTINUE                                                          VXY0154
C                                                                       VXY0155
C  SAVE VELOCITIES                                                      VXY0156
      WRITE(IF02)  (VX(K), VY(K), K = 1,M)                              VXY0157
C                                                                       VXY0158
C  CALCULATE COMBINED FLOW                                              VXY0159
      IF(IFLLL.EQ.0) GO TO 230
  200 VXC = 0.0                                                         VXY0160
      VYC = 0.0                                                         VXY0161
      DO 210 K = 1,M                                                    VXY0162
      VXC = VXC +VXT(K)*CK(K)                                           VXY0163
  210 VYC = VYC +VYN(K)*CK(K)                                           VXY0164
C                                                                       VXY0165
      DO 220 J = 1,N                                                    VXY0166
      VXC = VXC + B(J)*CSIG(J)                                          VXY0167
  220 VYC = VYC + A(J)*CSIG(J)                                          VXY0168
C                                                                       VXY0169
C  SAVE VELOCITIES                                                      VXY0170
      WRITE(IF01)  VXC, VYC                                             VXY0171
C                                                                       VXY0172
C                                                                       VXY0173
  230 CONTINUE                                                          VXY0174
C                                                                       VXY0175
C                                                                       VXY0176
      RETURN                                                            VXY0177
      END                                                               VXY0178
```

```
      COMPILER(XM=1)
      SUBROUTINE  VPROFF(N, M, NO, X, Y, TITLE, IND)
C
      COMMON /FILEID/  IFA1, IFA2, IF03, IF04, IF05,
     1                 IF06, IF07, IF08, IF09, IF10,
     2                 IF11, IF12, IF13, IF14, IF15
     3                ,IF16, IF17, IF18, IF19, IF20
      DIMENSION X(1), Y(1), TITLE(7), V(1200),V1X(200),V2X(200),
     1  V3X(200),V4X(200),V5X(200)
      COMMON/OVER/ V6X(200),V1Y(200),V2Y(200),V3Y(200),V4Y(200),
     1             V5Y(200),V6Y(200)
      COMMON/COM/IFLLL
C
      DATA RD/57.2957797/
C
C  VX AND VY HAVE BEEN SAVED ON UNIT IF02
      REWIND  IF01
      REWIND  IF02
C
      IF (IND .NE. 1)  GO TO 40
C
C  INDIVIDUAL FLOWS
      M2 = M + M
      I1 = 1 - M2
      I2 = 0
      DO 10 J = 1,NO
      I1 = I1 + M2
      I2 = I2 + M2
C
   10 READ(IF02)  (V(K), K = I1,I2)
C
      DO 30 K = 1,M
      WRITE(6,60)  K, TITLE
C
      DO 20 I = 1,NO
      IX = (I-1)*M2 + 2*K - 1
      IY = IX + 1
      VT = SQRT(V(IX)**2 + V(IY)**2)
      TH = ATAN2(V(IY), V(IX))*RD
      IF(K.EQ.1) V1X(I)=V(IX)
      IF(K.EQ.1) V1Y(I)=V(IY)
      IF(K.EQ.2) V2X(I)=V(IX)
      IF(K.EQ.2) V2Y(I)=V(IY)
      IF(K.EQ.3) V3X(I)=V(IX)
      IF(K.EQ.3) V3Y(I)=V(IY)
      IF(K.EQ.4) V4X(I)=V(IX)
      IF(K.EQ.4) V4Y(I)=V(IY)
      IF(K.EQ.5) V5X(I)=V(IX)
      IF(K.EQ.5) V5Y(I)=V(IY)
      IF(K.EQ.6) V6X(I)=V(IX)
      IF(K.EQ.6) V6Y(I)=V(IY)
C
   20 WRITE(6,70)  I, X(I), Y(I), V(IX), V(IY), VT, TH
C
   30 CONTINUE
      WRITE(7,1500) (X(J),J=1,NO)
```

```
      WRITE(7,1500) (Y(J),J=1,NO)
      WRITE(7,1500) (V1X(J),J=1,NO)
      WRITE(7,1500) (V2X(J),J=1,NO)
      WRITE(7,1500) (V3X(J),J=1,NO)
      WRITE(7,1500) (V4X(J),J=1,NO)
      WRITE(7,1500) (V5X(J),J=1,NO)
      IF(K.GE.6) WRITE(7,1500) (V6X(J),J=1,NO)
      WRITE(7,1500) (V1Y(J),J=1,NO)
      WRITE(7,1500) (V2Y(J),J=1,NO)
      WRITE(7,1500) (V3Y(J),J=1,NO)
      WRITE(7,1500) (V4Y(J),J=1,NO)
      WRITE(7,1500) (V5Y(J),J=1,NO)
      IF(K.GE.6) WRITE(7,1500) (V6Y(J),J=1,NO)
 1500 FORMAT(0P6E13.8)
C                                                                       VPR00039
C                                                                       VPR00040
C  COMBINED FLOW                                                        VPR00041
      IF(IFLLL.EQ.0) GO TO 91
   40 WRITE(6,80)  TITLE                                                VPR00042
C                                                                       VPR00043
      DO 50 I = 1,NO                                                    VPR00044
      READ(IFD1)  V(1), V(2)                                            VPR00045
      VT = SQRT(V(1)**2 + V(2)**2)                                      VPR00046
      TH = ATAN2(V(2), V(1))*RD                                         VPR00047
   50 WRITE(6,70)  I, X(I), Y(I), V(1), V(2), VT, TH                    VPR00048
C                                                                       VPR00049
   60 FORMAT(1H1, 20X 'INDIVIDUAL FLOW NO. ',I2, 5X'OFFBODY POINTS',    VPR0005C
     1       5X,7A4//T15,'I', T27,'X(I)', T44,'Y(I)', T61,'VX',         VPR00051
     2       T78,'VY', T95,'VT', T109,'THETA(DEG)'//)                   VPR00052
   70 FORMAT(1H , 11XI3, 6(5XF12.6))                                    VPR00053
   80 FORMAT(1H1, 20X 'COMBINED FLOW FOR OFFBODY POINTS', 5X,7A4//      VPR00054
     1       T15,'I', T27,'X(I)', T44,'Y(I)', T61,'VX',                 VPR00055
     2       T78,'VY', T95,'VT', T109,'THETA(DEG)'//)                   VPR00056
C                                                                       VPR00057
   91 RETURN                                                            VPR00058
      END                                                               VPR00059
```

```
      SUBROUTINE WEIGHT ( SUMDS, DS, JI, JF, WF, IPVOR)                 WEI6001
C                                                                       WEI6002
      DIMENSION     DS (1),   WF (1)                                    WEI6003
      IF ( IPVOR .EQ. 0)  GO TO 20                                      WEI6004
C                                                                       WEI6005
      WRITE ( 6, 1000 )                                                 WEI6006
 1000 FORMAT  ( 1H1, 'VORTICITY WEIGHTING FUNCTION = S/L*(1 - S/L)'//)  WEI6007
C                                                                       WEI6008
      S1 = 0.0                                                          WEI6009
      SD1= 0.0                                                          WEI6010
      SDN= 1.0                                                          WEI6011
C                                                                       WEI6012
      DO 10 J = JI, JF                                                  WEI6013
      SD = DS (J) / (2.0 * SUMDS)                                       WEI6014
      S  = S1 + SD                                                      WEI6015
      S1 = S  + SD                                                      WEI6016
      WF (J) = (S - SD1) * (SDN - S)                                    WEI6017
   10 CONTINUE                                                          WEI6018
C                                                                       WEI6019
      WRITE ( 6, 1010 ) ( WF (J), J = JI, JF )                          WEI6020
 1010 FORMAT  ( 1H , 6F18.6 )                                           WEI6021
C                                                                       WEI6022
      RETURN                                                            WEI6023
C                                                                       WEI6024
   20 CONTINUE                                                          WEI6025
      DO 30 J = JI, JF                                                  WEI6026
   30 WF (J) = 1.0                                                      WEI6027
C                                                                       WEI6028
      WRITE ( 6, 1020 )                                                 WEI6029
 1020 FORMAT  ( 1H1, 'VORTICITY WEIGHTING FUNCTION CONSTANT = 1.0' //)  WEI6030
C                                                                       WEI6031
      RETURN                                                            WEI6032
      END                                                               WEI6033
```

```
      FUNCTION  XYFORM (XI, YI, DSI, SINI, COSI, IFST,                  XYF0001
     1                  XJ, YJ, DSJ, SINJ, COSJ, ISEC,                  XYF0002
     2                  AI, AC, AP, BO, BI, BC, BP)                     XYF0003
C                                                                       XYF0004
C                                                                       XYF0005
C  THIS ROUTINE ACTUALLY CALCULATES THE INDUCED VELOCITY                XYF0006
C  ARRAY ELEMENTS A AND B.                                              XYF0007
C                                                                       XYF0008
C                                                                       XYF0009
      DATA E1,E2/169.0,11.111/,EO,EY/0.0001,0.01/                       XYF0010
C                                                                       XYF0011
C                                                                       XYF0012
      DX = XI - XJ                                                      XYF0013
      DY = YI - YJ                                                      XYF0014
      ROSQ = DX**2 + DY**2                                              XYF0015
      DSJSQ = DSJ**2                                                    XYF0016
C                                                                       XYF0017
      IF (ROSQ .LT. DSJSQ*E1)  GO TO 10                                 XYF0018
C                                                                       XYF0019
C  USE FAR FIELD FORMULAS                                               XYF0020
      VX = 2.*DSJ/ROSQ                                                  XYF0021
      VY = VX*DY                                                        XYF0022
      VX = VX*DX                                                        XYF0023
      AO = -VX*SINI + VY*COSI                                           XYF0024
      XYFORM = AO                                                       XYF0025
      BO =  VX*COSI + VY*SINI                                           XYF0026
      IF ((IFST .EQ. 0) .AND. (ISEC .EQ. 0))  RETURN                    XYF0027
      SC = SINI*COSJ                                                    XYF0028
      SS = SINI*SINJ                                                    XYF0029
      CS = COSI*SINJ                                                    XYF0030
      CC = COSI*COSJ                                                    XYF0031
      SSCC = (SS - CC)*VX*VY/12.0                                       XYF0032
      CSSC = (CS + SC)*VX*VY/12.0                                       XYF0033
      DXR = 1.0 - 2.0*DX**2/ROSQ                                        XYF0034
      DYR = 1.0 - 2.0*DY**2/ROSQ                                        XYF0035
      DDR = DSJSQ/(ROSQ*6.0)                                            XYF0036
C                                                                       XYF0037
      AI = DDR*(SC*DXR - CS*DYR) - SSCC                                 XYF0038
      BI =-DDR*(CC*DXR + SS*DYR) + CSSC                                 XYF0039
      AC =-DDR*(SS*DXR + CC*DYR) - CSSC                                 XYF0040
      BC = DDR*(CS*DXR - SC*DYR) - SSCC                                 XYF0041
      AP = AO/12.0                                                      XYF0042
      BP = BO/12.0                                                      XYF0043
      RETURN                                                            XYF0044
C                                                                       XYF0045
C  USE NEAR FIELD FORMULAS                                              XYF0046
   10 X = DX*COSJ + DY*SINJ                                             XYF0047
      Y = DY*COSJ - DX*SINJ                                             XYF0048
      S = SINI*COSJ - COSI*SINJ                                         XYF0049
      C = COSI*COSJ + SINI*SINJ                                         XYF0050
C                                                                       XYF0051
      IF (ROSQ .GT. DSJSQ*E2)  GO TO 20                                 XYF0052
C                                                                       XYF0053
C  USE EXACT FORMULAS                                                   XYF0054
      XB = X/DSJ                                                        XYF0055
      YB = Y/DSJ                                                        XYF0056
```

```
      RO = XB**2 + YB**2                                                XYF0057
      RD = XB**2 - YB**2                                                XYF0058
      R1 = RO + XB + 0.25                                               XYF0059
      R2 = RO - XB + 0.25                                               XYF0060
C                                                                       XYF0061
      VX = +ALOG((ROSQ+DSJ*X+0.25*DSJSQ)/(ROSQ-DSJ*X+0.25*DSJSQ))       XYF0062
      Y = Y*DSJ                                                         XYF0063
      X = ROSQ - 0.25*DSJSQ                                             XYF0064
      VY = 2.0*ATAN2(Y,X)                                               XYF0065
      IF ((IFST .EQ. 0) .AND. (ISEC .EQ. 0))  GO TO 30                  XYF0066
C                                                                       XYF0067
      VXO = VX                                                          XYF0068
      VYO = VY                                                          XYF0069
      VX1 = YB*VYO + XB*VXO - 2.0                                       XYF0070
      VY1 = XB*VYO - YB*VXO                                             XYF0071
      VXC = -2.0*VY1 + XB*YB/(R1*R2)                                    XYF0072
      VYC = 2.0*(VX1 + 1.0)-2.*(RO**2 - 0.25*RD)/(R1*R2)                XYF0073
      VXP = RD*VXO +  2.0*XB*(YB*VYO - 1.0)                             XYF0074
      VYP = RD*VYO -  2.0*YB*(XB*VXO - 1.0)                             XYF0075
C                                                                       XYF0076
      GO TO 30                                                          XYF0077
C                                                                       XYF0078
C                                                                       XYF0079
C  USE MULTIPOLE FORMULAS                                               XYF0080
   20 AE = X*DSJ/ROSQ                                                   XYF0081
      BE = Y*DSJ/ROSQ                                                   XYF0082
      ASQ = X**2/ROSQ                                                   XYF0083
      ESQ =DSJSQ/ROSQ                                                   XYF0084
      VX  = 2.0*AE*(1.0 + (ASQ - 0.75)*ESQ/3.0)                         XYF0085
      VY  = 2.0*BE*(1.0 + (ASQ - 0.25)*ESQ/3.0)                         XYF0086
      IF ((IFST .EQ. 0) .AND. (ISEC .EQ. 0))  GO TO 30                  XYF0087
      A4 = 0.15*(8.0*ASQ*(ASQ - 1.0) + 1.0)*ESQ                         XYF0088
      VX1 = ESQ*(2.0*ASQ - 1.0 + A4)/6.0                                XYF0089
      VY1 = AE*BE*(1.0 + 0.3*(2.0*ASQ - 1.0)*ESQ)/3.0                   XYF0090
      VXC = AE*BE*(1.0 + 0.3*(6.0*ASQ - 1.0)*ESQ)/3.0                   XYF0091
      BSQ = Y**2/ROSQ                                                   XYF0092
      VYC = ESQ*(2.0*BSQ - 1.0 -0.5*A4)/6.0                             XYF0093
      VYP = (1.0 + (ASQ - 0.25)*ESQ*0.6)/6.0                            XYF0094
      VXP = AE*VYP                                                      XYF0095
      VYP = BE*VYP                                                      XYF0096
C                                                                       XYF0097
C                                                                       XYF0098
   30 CONTINUE                                                          XYF0099
      XYFORM = -S*VX + C*VY                                             XYF0100
      BO = C*VX + S*VY                                                  XYF0101
      IF ((IFST .EQ. 0) .AND. (ISEC .EQ. 0))  RETURN                    XYF0102
      A1 = -S*VX1 + C*VY1                                               XYF0103
      AC = -S*VXC + C*VYC                                               XYF0104
      AP = -S*VXP + C*VYP                                               XYF0105
C                                                                       XYF0106
      B1 = C*VX1 + S*VY1                                                XYF0107
      BC = C*VXC + S*VYC                                                XYF0108
      BP = C*VXP + S*VYP                                                XYF0109
C                                                                       XYF0110
C                                                                       XYF0111
      RETURN                                                            XYF0112
      END                                                               XYF0113
```

Program NOZZLEC

M NOZZLEC

```
      COMMON /COORD/ XON(700),YON(700),XOFF(200),YOFF(200),S(700),
     1               S1(700),XTEST,XTEST1,XTEST2,YCL,YCU,YCL1,YCL2,YCU1,
     2               YCU2,XR1,XR2,XRH,YR1,YR2,YRH,CUTOF1,CUTOF2,CUTOFH,
     3               ELND,ANG(700),AR(700),AROFF(200)
      COMMON /VELOC/ V1(700),V2(700),V3(700),V4(700),V5(700),V1X(200),
     1               V2X(200),V3X(200),V4X(200),V5X(200),V1Y(200),
     2               V2Y(200),V3Y(200),V4Y(200),V5Y(200)
      COMMON /CONT/ VC,VS1,VS2,XMC,XMC1,XMC2,WDOTC,WDOTC1,WDOTC2,
     1               TITLE(3),VINF,ALFA,A,B,C,D,A1C,A2C,A3C,A4C,A5C,A11,
     2               A21,A31,A41,A51,A12,A22,A32,A42,A52,VIC,VIC1,VIC2
      COMMON /COUT/ NT,NS1,NH,NP,IW,NX,KND,ICOMP,K,NXHI1,NXHI2,NXHI3,
     1               NST2,NST3,NST7,NPPR(30),IRAK(30),M1,M2,ICOMP1,IPL,
     2               IHUB
      COMMON /CONDIT/ TTOTAL,PT,PSTAT,TSTAT,PSTATC,ATOTAL,PTC,RHOST,
     1               RHOTOT,ASTAT,QCINF,RSORTC
      COMMON /WRIT/  AA1C,AA2C,AA3C,AA4C,AA5C,AA11,AA21,AA31,AA41,AA51,
     1               AA12,AA22,AA32,AA42,AA52
      COMMON /SOLUT/ VBAR(700),VBARO(200),VINC(700),VXINC(200),
     1               VYINC(200),RHOB(700),RBORT(700),RHOBC(200),
     2               VCOM(700),RBOOT(200),VRE(200),VRECOM(200),
     3               VXCOM(200),VYCOM(200),THETA(200),PSCPTC(700),
     4               PSOPT(700),CMACH(700),XMACH(700),CPI(700),CPC(700),
     5               RHOI(700)
      COMMON /SOLUTO/ PSOFPC(200),PSOFP(200),CMACO(200),XMACO(200),
     1                RHCOI(200)
      COMMON /PICT/ VPERIN,XX,XMIN,EXEF,YY,YMIN,ORD,EMSTCR,AL,AAAA
      COMMON /CLPLOT/ XPEN,YPEN,NX6,NY,IPEN,XLABEL(10),YLABEL(10)
      COMMON/FINDER/ JCHK1,JCHK2,JCHK3
C --------------------------------------------------------------------
C
C
C  THIS IS THE MAIN PROGRAM WHICH CALLS THE SUBROUTINES TO
C  COMPUTE THE 2-D COMBINATION SOLUTIONS FOR BOTH COMPRESSIBLE
C  AND INCOMPRESSIBLE VERSIONS.
C
C
      DIMENSION XPLOT(700),YPLOT(700),KKK(8),P(14)
      CALL INPTR
      AAAA=VINF
      VSAVE = VINF
      IF(ICOMP1.EQ.1) ICOMP=0
      IF(ICOMP1.EQ.0) ICOMP=1
      IF(ICOMP .EQ.1) VINF=VINF*(1.0-0.2*(VINF/ATOTAL)**2)**2.5
      CALL SEARCH
      CALL ANGLEF
      VCSAVE=VC
      VC1S=VS1
      VC2S=VS2
      VC=VIC
      VS1=VIC1
      VS2=VIC2
   10 CALL SOLVE
      VC = VCSAVE
      VS1=VC1S
      VS2=VC2S
      VINFP=VINF
```

```
      VINF=VSAVE
      IF(IHUB.EQ.0) GO TO 11
      WRITE(6,100)
      IF(M1.EQ.1) WRITE(6,105) A11,A21,A31,A41,A51,A12,A22,A32,
     1                         A42,A52
      IF(M2.EQ.1) WRITE(6,110) A1C,A2C,A3C,A4C,A5C,A12,A22,A32,
     1                         A42,A52
      IF((M1.EQ.0).AND.(M2.EQ.0)) WRITE(6,115) A1C,A2C,A3C,A4C,A5C,
     1                         A11,A21,A31,A41,A51
      WRITE(6,120) A,B,C,D,VINFP
      GO TO 12
   11 WRITE(6,101)
      WRITE(6,102) A1C,A2C,A3C,A4C
      WRITE(6,103) A,B,C,VINFP
   12 IF(ICOMP.EQ.1) CALL COMCOR
      CALL ONBODY
      CALL OFBDY
C
C  THE FOLLOWING CODE PLOTS PS/PT VS S AND MACH NUMBER VS S.
C
      IF(CUTOF1.LE.0.0) GO TO 20
      DO 15 I=1,NS1
      IF(S(I).LT.0.0) S1(I)=-S(I)/CUTOF1
      IF(S(I).GE.0.0) S1(I)= S(I)/CUTOF1
   15 CONTINUE
      J=0
      J1=0
      DO 16 I=1,NS1
      IF(S1(I).GT.1.0) GO TO 16
      IF(I.LT.JCHK1) J=J+1
      IF(J.EQ.1) IS=I
      IF(I.GE.JCHK1) J1=J1+1
   16 CONTINUE
      IPL=1
      CALL PLTER(J1,J,JCHK1,IS,1)
   20 IF(CUTOFH.LE.0.0) GO TO 30
      IN=NS1+1
      IT=NH
      DO 25 I=IN,IT
      IF(S(I).LT.0.0) S1(I)=-S(I)/CUTOFH
      IF(S(I).GE.0.0) S1(I)= S(I)/CUTOFH
   25 CONTINUE
      J=0
      J1=0
      DO 26 I=IN,IT
      IF(S1(I).GT.1.0) GO TO 26
      IF(I.LE.JCHK2) J=J+1
      IF(J.EQ.1) IS=I
      IF(I.GT.JCHK2) J1=J1+1
   26 CONTINUE
      IPL=2
      CALL PLTER(J1,J,JCHK2,IS,1)
   30 IF(CUTOF2.LE.0.0) GO TO 37
      IN=NH+1
      DO 35 I=IN,NT
      IF(S(I).LT.0.0) S1(I)=-S(I)/CUTOF2
      IF(S(I).GE.0.0) S1(I)= S(I)/CUTOF2
```

```
   35 CONTINUE
      J=0
      J1=0
      DO 36 I=IN,NT
      IF(S1(I).GT.1.0) GC TO 36
      IF(I.LE.JCHK3) J=J+1
      IF(J.EQ.1) IS=I
      IF(I.GT.JCHK3) J1=J1+1
   36 CONTINUE
      IPL=3
      CALL PLTER(J,J1,JCHK3,IS,2)
   37 IPL=10
      IF(VPERIN.EQ.0) GO TO 40
      KKK(1) = 4
      KKK(2) = 0
      KKK(3) = 3
      IF(IHUB.EQ.0) KKK(3) = 2
      KKK(4) = 1
      KKK(5) = 1
      P(1) = 3.0
      P(2) = XX
      P(3) = XMIN
      P(4) = XMIN+XX*EXEP
      P(5) = YY
      P(6) = YMIN
      P(7) = YMIN+YY*ORD
      P(8) = 10.0
      P(9) = 0.0
      P(10) = 0.0
      P(11) = 0.0
      P(12) = 0.0
      P(13) = 0.0
      P(14) = 90.0
      II=0
      III=0
      DO 200 I=1,NS1
      IF(XON(I).LT.XMIN) GO TO 200
      IF(YON(I).LT.YMIN) GO TO 200
      IF(XON(I)-XX*EXEP-XMIN) 130,130,200
  130 IF(YON(I)-YY*ORD-YMIN) 135,135,2C0
  135 II=II+1
      III=III+1
      XPLOT(III)=XON(I)
      YPLOT(III)=YON(I)
  200 CONTINUE
      KKK(6)=II
      II=0
      IF(IHUB.EQ.0) GO TC 250
      NN=NS1+1
      DO 240 I=NN,NH
      IF(XON(I).LT.XMIN) GO TO 240
      IF(YON(I).LT.YMIN) GO TO 240
      IF(XON(I)-XX*EXEP-XMIN) 230,230,240
  230 IF(YON(I)-YY*ORD-YMIN) 235,235,24C
  235 II=II+1
      III=III+1
      XPLOT(III)=XON(I)
```

```
      YPLOT(III)=YON(I)
  240 CONTINUE
      KKK(7) = II
  250 II=0
      NN=NH+1
      DO 270 I=NN,NT
      IF(XON(I).LT.XMIN) GO TO 270
      IF(YON(I).LT.YMIN) GO TO 270
      IF(XON(I)-XX*EXEP-XMIN) 260,260,270
  260 IF(YON(I)-YY*ORD-YMIN) 265,265,270
  265 II=II+1
      III=III+1
      XPLOT(III)=XON(I)
      YPLOT(III)=YON(I)
  270 CONTINUE
      IF(IHUB.EQ.0) KKK(7)=II
      IF(IHUB.NE.0) KKK(8)=II
      DO 280 I=1,3
      XLABEL(I)=TITLE(I)
  280 CONTINUE
      XPEN=0.0
      YPEN=0.0
      IPEN=-3
      NX6=-18
      NY=0
      CALL CALPLT(XPLOT,YPLOT,KKK,P)
   40 STOP
  100 FORMAT(/,16X,'V1',12X,'V2',12X,'V3',12X,'V4',12X,'V5',/)
  101 FORMAT(/,16X,'V1',12X,'V2',12X,'V3',12X,'V4',/)
  102 FORMAT(2X,'CONTROL',3X,4(1PE10.3,4X))
  103 FORMAT(/,10X,'A',13X,'B',13X,'C',11X,'VINFP',/,6X,4(1PE10.3,4X))
  105 FORMAT(2X,'LOWER',/,2X,'PASSAGE',3X,5(1PE10.3,4X),/,2X,'UPPER',/,
     I          2X,'PASSAGE',3X,5(1PE10.3,4X))
  110 FORMAT(2X,'CONTROL',3X,5(1PE10.3,4X),/,2X,'UPPER',/,2X,'PASSAGE',
     I          3X,5(1PE10.3,4X))
  115 FORMAT(2X,'CONTROL',3X,5(1PE10.3,4X),/,2X,'LOWER',/,2X,'PASSAGE',
     1          3X,5(1PE10.3,4X))
  120 FORMAT(/,10X,'A',13X,'B',13X,'C',13X,'D',11X,
     I           'VINFP',/,6X,5(1PE10.3,4X))
      END
```

```
      SUBROUTINE INPTR
      COMMON /COORD/ XON(700),YON(700),XOFF(200),YOFF(200),S(700),
     1               S1(700),XTEST,XTEST1,XTEST2,YCL,YCU,YCL1,YCL2,YCU1,
     2               YCU2,XR1,XR2,XRH,YR1,YR2,YRH,CUTOF1,CUTOF2,CUTOFH,
     3               ELND,ANG(700),AR(700),AROFF(200)
      COMMON /VELOC/ V1(700),V2(700),V3(700),V4(700),V5(7CC),V1X(200),
     1               V2X(200),V3X(200),V4X(200),V5X(200),V1Y(200),
     2               V2Y(200),V3Y(200),V4Y(200),V5Y(200)
      COMMON /CONT/ VC,VS1,VS2,XMC,XMC1,XMC2,WDOTC,WDOTC1,WDOTC2,
     1               TITLE(3),VINF,ALFA,A,B,C,D,A1C,A2C,A3C,A4C,A5C,A11,
     2               A21,A31,A41,A51,A12,A22,A32,A42,A52,VIC,VIC1,VIC2
      COMMON /COUT/ NT,NS1,NH,NP,IW,NX,KND,ICOMP,K,NXHI1,NXHI2,NXHI3,
     1               NST2,NST3,NST7,NPFR(30),IRAK(30),M1,M2,ICOMP1,IPL,
     2               IHUB
      COMMON /CONDIT/ TICTAL,PT,PSTAT,TSTAT,PSTATC,ATOTAL,PIC,RHOST,
     1               RHOTOT,ASTAT,QCINF,RSORTC
      COMMON/PICT/VPERIN,XX,XMIN,EXEP,YY,YMIN,ORD,EMSTOR,AL,AAAA
C ----------------------------------------------------------------------
C
C  THIS SUBROUTINE READS DATA FROM BOTF CARDS AND DISK FILES.
C
      READ(5,100) TITLE
      WRITE(6,101) TITLE
      READ(5,110)  NT,NS1,NH,NP,IW,NX,KND,ICOMP1,IHUB
      WRITE(6,111) NT,NS1,NH,NP,IW,NX,KND,ICOMP1,IHUB
      READ(5,120)  VC,VS1,VS2,VINF,ALFA,XMC,XMC1,XMC2,TTCTAL,PT
      WRITE(6,121) VC,VS1,VS2,VINF,ALFA,XMC,XMC1,XMC2,TTCTAL,PT
      READ(5,120)  ELND,WDOTC,WDOTC1,WCCTC2,PSTAT,TSTAT,CLTOF1,CUTOF2,
     1          CUTOFH,VPERIN
      WRITE(6,121) ELND,WDOTC,WDOTC1,WDOTC2,PSTAT,TSTAT,CUTOF1,CUTOF2,
     1          CUTOFH,VPERIN
      IF(VPERIN.NE.0) READ(5,130) XX,XMIN,EXEP,YY,YMIN,ORD
      IF(VPERIN.NE.0) WRITE(6,131) XX,XMIN,EXEP,YY,YMIN,CRD
C
C  NT = TOTAL NUMBER OF CN-BODY POINTS.
C  NS1 = TOTAL NUMBER OF ON-BODY POINTS ON BODY 1
C  NH = TOTAL NUMBER OF CN-BODY POINTS ON BODIES 1 AND 2
C  IW = 0 - WEIGHT FLOW CATA AT CONTROL STATIONS INPUT
C     = 1 - MACH NUMBERS AT CONTROL STATIONS INPUT
C     = 2 - VELOCITIES AT CONTROL STATIONS INPUT
C  NX = 1 SUPERSONIC VELCCITY CORRECTION APPLIES.
C  KND = 0 DATA NOT SCALED.
C      = 6 DATA SCALED BY CONTROL STATION PASSAGE HEIGHT.
C  ICOMP1 = 0 COMPRESSIBLE VERSION
C         = 1 INCOMPRESSIBLE VERSION
C  NOTE -- TWO OF THE THREE CONTROL STATIONS' DATA MUST BE INPUT, IF IHUB NE 0
C  IHUB = 0 NO BODY 2
C
C
      READ(5,140)  XTEST,YCL,YCU
      WRITE(6,141) XTEST,YCL,YCU
      IF(IHUB.EQ.0) GO TC 3
      READ(5,140)  XTEST1,YCL1,YCU1
      WRITE(6,141) XTEST1,YCL1,YCU1
      READ(5,140)  XTEST2,YCL2,YCU2
      WRITE(6,141) XTEST2,YCL2,YCU2
```

```
    3 READ(5,140)   XR1,XR2,XRH
      WRITE(6,141)  XR1,XR2,XRH
      READ(5,140)   YR1,YR2,YRH
      WRITE(6,141)  YR1,YR2,YRH
C  XTEST,XTEST1,XTEST2 ARE THE LOCATIONS OF THE CONTROL STATIONS
C  DOWNSTREAM OF BODY 2, BETWEEN BODIES 1 AND 2,
C  AND BETWEEN BODIES 2 AND 3, RESPECTIVELY.
C  XR1,XRH,XR2 ARE THE POINTS ON EACH OF THE BODIES WHERE
C  SURFACE DISTANCE EQUALS ZERO.
      READ(7,150) (XON(J),J=1,NT)
      READ(7,150) (YON(J),J=1,NT)
      READ(7,150) (V1(J),J=1,NT)
      READ(7,150) (V2(J),J=1,NT)
      READ(7,150) (V3(J),J=1,NT)
      READ(7,150) (V4(J),J=1,NT)
      READ(7,150) (V5(J),J=1,NT)
      IF(NP.GT.100) GO TO 4
      READ(7,150) (XOFF(J),J=1,NP)
      READ(7,150) (YOFF(J),J=1,NP)
      READ(7,150) ( V1X(J),J=1,NP)
      READ(7,150) ( V2X(J),J=1,NP)
      READ(7,150) ( V3X(J),J=1,NP)
      READ(7,150) ( V4X(J),J=1,NP)
      READ(7,150) ( V5X(J),J=1,NP)
      READ(7,150) ( V1Y(J),J=1,NP)
      READ(7,150) ( V2Y(J),J=1,NP)
      READ(7,150) ( V3Y(J),J=1,NP)
      READ(7,150) ( V4Y(J),J=1,NP)
      READ(7,150) ( V5Y(J),J=1,NP)
      GO TO 5
    4 READ(7,150) (XOFF(J),J=1,100)
      READ(7,150) (YOFF(J),J=1,100)
      READ(7,150) ( V1X(J),J=1,100)
      READ(7,150) ( V2X(J),J=1,100)
      READ(7,150) ( V3X(J),J=1,100)
      READ(7,150) ( V4X(J),J=1,100)
      READ(7,150) ( V5X(J),J=1,100)
      READ(7,150) ( V1Y(J),J=1,100)
      READ(7,150) ( V2Y(J),J=1,100)
      READ(7,150) ( V3Y(J),J=1,100)
      READ(7,150) ( V4Y(J),J=1,100)
      READ(7,150) ( V5Y(J),J=1,100)
      READ(7,150) (XOFF(J),J=101,NP)
      READ(7,150) (YOFF(J),J=101,NP)
      READ(7,150) ( V1X(J),J=101,NP)
      READ(7,150) ( V2X(J),J=101,NP)
      READ(7,150) ( V3X(J),J=101,NP)
      READ(7,150) ( V4X(J),J=101,NP)
      READ(7,150) ( V5X(J),J=101,NP)
      READ(7,150) ( V1Y(J),J=101,NP)
      READ(7,150) ( V2Y(J),J=101,NP)
      READ(7,150) ( V3Y(J),J=101,NP)
      READ(7,150) ( V4Y(J),J=101,NP)
      READ(7,150) ( V5Y(J),J=101,NP)
    5 WRITE(6,160) TITLE
      WRITE(6,170)
      IF(ICOMP1.EQ.0) WRITE(6,180)
```

```
      IF(ICOMP1.EQ.1) WRITE(6,175)
      IF(IHUB.EQ.0) WRITE(6,189)
      IF(IHUB.NE.0) WRITE(6,190)
      IF(ELND.EQ.0.0) ELND=1.0
      AL = ALFA
      CALL CONST
C  THE FOLLOWING SCALES THE DATA BY THE VALUE OF ELND.
      DO 10 I=1,NT
      XON(I)=XON(I)/ELND
      YON(I)=YON(I)/ELND
   10 CONTINUE
      DO 20 I=1,NP
      XOFF(I)=XOFF(I)/ELND
      YOFF(I)=YOFF(I)/ELND
   20 CONTINUE
      RETURN
C  *****FORMATS*****
  100 FORMAT(3A6)
  101 FORMAT(1H ,3A6)
  110 FORMAT(9I4)
  111 FORMAT(1H ,9I4)
  120 FORMAT(0P10F8.0)
  121 FORMAT(1H ,0P10F8.3)
  130 FORMAT(0P6F10.0)
  131 FORMAT(1H ,0P6F10.3)
  140 FORMAT(0P3F10.0)
  141 FORMAT(1H ,0P3F10.3)
  150 FORMAT(0P6E13.8)
  160 FORMAT(1H1,40X,3A6)
  170 FORMAT(///,2X,'2-D COMBINATION SOLUTION')
  175 FORMAT(/,6X,'INCOMPRESSIBLE VERSION')
  180 FORMAT(/,6X,  'COMPRESSIBLE VERSION')
  189 FORMAT(/,6X,'COMBINATION OF THE FOLLOWING BASIC SOLUTIONS',/,9X,
     1          '1. UNIFORM AXIAL',/,9X,'2. UNIFORM CROSSFLOW',/,9X,
     2          '3. VORTICITY ABOUT BODY 1',/,9X,'4. VORTICITY ABOUT '
     3          'BODY 3',/)
  190 FORMAT(/,6X,'COMBINATION OF THE FOLLOWING BASIC SOLUTIONS',/,9X,
     1          '1. UNIFORM AXIAL',/,9X,'2. UNIFORM CROSSFLOW',/,9X,
     2          '3. VORTICITY ABOUT BODY 1',/,9X,'4. VORTICITY ABOUT '
     3          'BODY 2',/,9X,'5. VORTICITY ABOUT BODY 3',/)
      END
```

```
      VSONCC = ATOTAL/SQRT(1.2)
      IF(IHUB.EQ.0) GO TO 16
      AS1 = YCU1-YCL1
      AS2 = YCU2-YCL2
      IF((IW.EQ.0).AND.(WDOTC.EQ.0.0)) WDOTC = WDOTC1+WDOTC2
      IF((IW.EQ.0).AND.(WDOTC1.EQ.0.0)) WDOTC1 = WDOTC-WDOTC2
      IF((IW.EQ.0).AND.(WDOTC2.EQ.0.0)) WDOTC2 = WDOTC-WDOTC1
   16 IF(IW.EQ.1) GO TO 40
      GO TO 50
   40 VC = ATOTAL*XMC/SQRT(1.0+XMC**2/5.0)
      IF(IHUB.EQ.0) GO TO 25
      VS1 = ATOTAL*XMC1/SQRT(1.0+XMC1**2/5.0)
      VS2 = ATOTAL*XMC2/SQRT(1.0+XMC2**2/5.0)
      GO TO 25
   50 IF(IW.EQ.2) GO TO 25
      VIC = WDOTC/(6*RHOTOT*AC)*12.0
      CALL VBARIT(VIC,ATOTAL,RHOTOT,RHOC)
      VC = WDOTC/(6*RHOC*AC)*12.0
      IF(IHUB.EQ.0) GO TO 25
      VIC1 = WDOTC1/(6*RHOTOT*AS1)*12.0
      VIC2 = WDOTC2/(6*RHOTOT*AS2)*12.0
      CALL VBARIT(VIC1,ATOTAL,RHOTOT,RHOC1)
      CALL VBARIT(VIC2,ATOTAL,RHOTOT,RHOC2)
      VS1 = WDOTC1/(6*RHOC1*AS1)*12.0
      VS2 = WDOTC2/(6*RHOC2*AS2)*12.0
   25 IF(VC.GT.VSONCC) WRITE(6,116) VC
      IF(VC.GT.VSONCC) VC = VSONCC
      IF(IHUB.EQ.0) GO TO 26
      IF(VS1.GT.VSONCC) WRITE(6,117) VS1
      IF(VS1.GT.VSONCC) VS1 = VSONCC
      IF(VS2.GT.VSONCC) WRITE(6,118) VS2
      IF(VS2.GT.VSONCC) VS2 = VSONCC
   26 IF(KND.GE.8) GO TO 35
   30 IF((KND.EQ.-1).OR.(KND.EQ.4)) ELND = YCU
      IF((KND.EQ. 1).OR.(KND.EQ.6)) ELND = YCU-YCL
      IF((KND.EQ. 0).OR.(KND.EQ.5)) ELND = 1.0
      AC = AC/ELND
      YCU = YCU/ELND
      YCL = YCL/ELND
      XR1 = XR1/ELND
      XR2 = XR2/ELND
      YR1 = YR1/ELND
      YR2 = YR2/ELND
      XTEST = XTEST/ELND
      IF(IHUB.EQ.0) GO TO 35
      AS1 = AS1/ELND
      AS2 = AS2/ELND
      YCU1 = YCU1/ELND
      YCU2 = YCU2/ELND
      YCL1 = YCL1/ELND
      YCL2 = YCL2/ELND
      XRH = XRH/ELND
      YRH = YRH/ELND
      XTEST1 = XTEST1/ELND
      XTEST2 = XTEST2/ELND
   35 RTTOT = R21568*TTOTAL
      GRHO = G*PT/TTOTAL
```

```
      SUBROUTINE CONST
      COMMON /COORD/ XON(700),YON(700),XOFF(200),YOFF(200),S(700),
     1               S1(700),XTEST,XTEST1,XTEST2,YCL,YCU,YCL1,YCL2,YCU1,
     2               YCU2,XR1,XR2,XRH,YR1,YR2,YRH,CUTOF1,CUTOF2,CUTOFH,
     3               ELND,ANG(700),AR(700),AROFF(200)
      COMMON /CONT/ VC,VS1,VS2,XMC,XMC1,XMC2,WDOTC,WDOTC1,WDOTC2,
     1               TITLE(3),VINF,ALFA,A,B,C,D,A1C,A2C,A3C,A4C,A5C,A11,
     2               A21,A31,A41,A51,A12,A22,A32,A42,A52,VIC,VIC1,VIC2
      COMMON /COUT/ NT,NS1,NH,NP,IW,NX,KND,ICOMP,K,NXHI1,NXHI2,NXHI3,
     1               NST2,NST3,NST7,NPFR(30),IRAK(30),M1,M2,ICOMP1,IPL,
     2               IHUB
      COMMON /CONDIT/ TTOTAL,PT,PSTAT,TSTAT,PSTATC,ATOTAL,PTC,RHOST,
     1               RHOTOT,ASTAT,QCINF,RSQRTC
C ----------------------------------------------------------------------
C
C  THIS SUBROUTINE CALCULATES THE CONSTANTS USED IN THE PROGRAM
C
      PI=3.141592654
      M1=0
      M2=0
      IF(IHUB.EQ.0) GO TO 5
      IF((IW.EQ.0).AND.(WDOTC.EQ.0.0)) M1=1
      IF((IW.EQ.0).AND.(WDOTC1.EQ.0.0)) M2=1
      IF((IW.EQ.1).AND.(XMC.EQ.0.0)) M1=1
      IF((IW.EQ.1).AND.(XMC1.EQ.0.0)) M2=1
      IF((IW.EQ.2).AND.(VC.EQ.0.0)) M1=1
      IF((IW.EQ.2).AND.(VS1.EQ.0.0)) M2=1
    5 PIO180=PI/180.0
      R21568= 1716.06
      G = 32.174
      PSTATC = PSTAT
      IF((PSTAT.NE.0.0).AND.(TSTAT.NE.0.0)) GO TO 10
      IF(PT.EQ.0.0) PT = 2116.23
      IF(TTOTAL.EQ.0.0) TTOTAL = 518.69
      ATOTAL = 49.009*SQRT(TTOTAL)
      CATOT = 1.0-0.2*(VINF/ATOTAL)**2
      PSTATC = PT*CATOT**3.5
      PTC = PT
      RHOTOT = PT/(R21568*TTOTAL)
      TSTAT = TTOTAL*CATOT
      RHOST = PSTATC/(R21568*TSTAT)
      PSTAT = PT-0.5*RHOTOT*VINF*VINF
      ASTAT = 49.009*SQRT(TSTAT)
      GO TO 15
   10 ASTAT = 49.009*SQRT(TSTAT)
      RHOST = PSTAT/(R21568*TSTAT)
      AMINF = VINF/ASTAT
      CAMINF = 1.0+0.2*AMINF**2
      PTC = PSTAT*CAMINF**3.5
      PSTAT = PTC-0.5*RHOTOT*VINF*VINF
      PT=PTC
      TTOTAL = TSTAT*CAMINF
      RHOTOT = PT/(R21568*TTOTAL)
      ATOTAL = 49.009*SQRT(TTOTAL)
   15 AC = YCU-YCL
      VSONIC = ATOTAL/1.728
```

```
      C102RT = 0.5*RHOST/PT
      VFOAT = VINF/ATOTAL
      VCOAT = VC/ATOTAL
      VS1OAT = VS1/ATOTAL
      VS2OAT = VS2/ATOTAL
      CON1 = 1.0-0.2*VCOAT**2
      CON2 = 1.0-0.2*VFOAT**2
      CON11 = 1.0-0.2*VS1OAT**2
      CON21 = 1.0-0.2*VS2OAT**2
      RSORTF = CON2**2.5
      RSORTC = CON1**2.5
      RSORT1 = CON11**2.5
      RSORT2 = CON21**2.5
      IF((IW.EQ.1).OR.(IW.EQ.2)) VIC = VC*RSORTC
      IF((IW.EQ.1).OR.(IW.EQ.2)) VIC1 = VS1*RSORT1
      IF((IW.EQ.1).OR.(IW.EQ.2)) VIC2 = VS2*RSORT2
      IF((IW.EQ.1).OR.(IW.EQ.2)) WDOTC = VIC*G*RHOTOT*AC/12.0
      IF((IW.EQ.1).OR.(IW.EQ.2)) WDOTC1 = VIC1*G*RHOTOT*AS1/12.0
      IF((IW.EQ.1).OR.(IW.EQ.2)) WDOTC2 = VIC2*G*RHOTOT*AS2/12.0
      IF(IHUB.EQ.0) GO TO 36
      F = WDOTC1+WDOTC2
      IF(WDOTC.EQ.0.0) VIC = F/(G*RHOTOT*AC/12.0)
      IF(WDOTC.EQ.0.0) CALL VBARIT(VIC,ATOTAL,RHOTOT,RHOC)
      IF(WDOTC.EQ.0.0) VC = F/(G*RHOC*AC)*12.0
      IF(WDOTC.EQ.0.0) VCOAT = VC/ATOTAL
      IF(WDOTC.EQ.0.0) CON1 = 1.0-0.2*VCOAT**2
      IF(WDOTC.EQ.0.0) RSORTC = CON1**2.5
      IF(WDOTC.EQ.0.0) WDOTC = F
      F = WDOTC-WDOTC2
      IF(WDOTC1.EQ.0.0) VIC1= F/(G*RHOTOT*AS1/12.0)
      IF(WDOTC1.EQ.0.0) CALL VBARIT(VIC1,ATOTAL,RHOTOT,RHOC)
      IF(WDOTC1.EQ.0.0) VS1 = F/(G*RHOC*AS1)*12.0
      IF(WDOTC1.EQ.0.0) VS1OAT = VS1/ATOTAL
      IF(WDOTC1.EQ.0.0) CON11 = 1.0-0.2*VS1OAT**2
      IF(WDOTC1.EQ.0.0) RSORT1 = CON11**2.5
      IF(WDOTC1.EQ.0.0) WDOTC1= F
      F = WDOTC-WDOTC1
      IF(WDOTC2.EQ.0.0) VIC2= F/(G*RHOTOT*AS2/12.0)
      IF(WDOTC2.EQ.0.0) CALL VBARIT(VIC2,ATOTAL,RHOTOT,RHOC)
      IF(WDOTC2.EQ.0.0) VS2 = F/(G*RHOC*AS2)*12.0
      IF(WDOTC2.EQ.0.0) VS2OAT = VS2/ATOTAL
      IF(WDOTC2.EQ.0.0) CON21 = 1.0-0.2*VS2OAT**2
      IF(WDOTC2.EQ.0.0) RSORT2 = CON21**2.5
      IF(WDOTC2.EQ.0.0) WDOTC2= F
      IF(ICOMP1.EQ.1) VS1 = VIC1
      IF(ICOMP1.EQ.1) VS2 = VIC2
   36 IF(ICOMP1.EQ.1) VC = VIC
      VNFOVC = VINF/VC
      IF(KND.LT.8) WDOTC = WDOTC/ELND
      IF(KND.LT.8) WDOTC1 = WDOTC1/ELNC
      IF(KND.LT.8) WDOTC2 = WDOTC2/ELNC
      PSPTC = 1.0-C102RT*VC**2
      PSPTC1 = 1.0-C102RT*VS1**2
      PSPTC2 = 1.0-C102RT*VS2**2
      PSPTIF = 1.0-C102RT*VINF**2
      PSPTCI = RSORTF*CON2
      PSPT1  = RSORT1*CON11
```

```
      PSPT2  = RSORT2*CON21
      PSPTCC = RSORTC*CON1
      XMINF = VFOAT/(CON2**0.5)
      IF(XMC.EQ.0.0) XMC = VCOAT/(CON1**0.5)
      IF(XMC1.EQ.0.0) XMC1 = VS1OAT/(CCN11**0.5)
      IF(XMC2.EQ.0.0) XMC2 = VS2OAT/(CCN21**0.5)
      QINF = PT*(1.0-PSPTIF)
      QCINF = PTC*(0.7*VFOAT**2*RSORTF)
      QC = PT*(1.0-PSPTC)
      QCC = PTC*(0.7*VCOAT**2*RSORTC)
      QC1 = PT*(1.0-PSPTC1)
      QC2 = PT*(1.0-PSPTC2)
      QC1C = PTC*(0.7*VS1OAT**2*RSORT1)
      QC2C = PTC*(0.7*VS2OAT**2*RSORT2)
      THETC = TTOTAL/518.69
      DEL = PTC/2116.23
      IF(IHUB.EQ.0) GO TO 37
      WDOT1R = WDOTC1*SQRT(THETC)/DEL
      WDOT2R = WDOTC2*SQRT(THETC)/DEL
   37 WDOTCR = WDOTC*SQRT(THETC)/DEL
      IF(ICOMP1.EQ.0) GO TO 41
      WRITE(6,101)
      WRITE(6,111) VC,XMC,QC,PSPTC
      IF(IHUB.EQ.0) GO TO 38
      WRITE(6,121) VS1,XMC1,QC1,PSPTC1
      WRITE(6,131) VS2,XMC2,QC2,PSPTC2
   38 WRITE(6,141) VINF,XMINF,QINF,PSPTIF
      GO TO 51
   41 WRITE(6,100)
      WRITE(6,110) VC,XMC,QC,QCC,PSPTC,PSPTCC,RSORTC
      IF(IHUB.EQ.0) GO TO 39
      WRITE(6,120) VS1,XMC1,QC1,QC1C,PSPTC1,PSPT1,RSORT1
      WRITE(6,130) VS2,XMC2,QC2,QC2C,PSPTC2,PSPT2,RSORT2
   39 WRITE(6,140) VINF,XMINF,QINF,QCINF,PSPTIF,PSPTCI,RSORTF
   51 WRITE(6,150)
      WRITE(6,155) ALFA,WNFOVC,VSONIC,VSONCC,WDOTCR,WDOT1R,WDOT2R
      WRITE(6,160)
      WRITE(6,165)
      WRITE(6,170) TSTAT,PSTAT,PSTATC,ASTAT,RHOST,WDOTC,WDOTC1,WDOTC2
      WRITE(6,175) VIC,VIC1,VIC2
      WRITE(6,180)
      WRITE(6,155) TTOTAL,PT,PTC,ATOTAL,RHOTOT,THETC,DEL
      WRITE(6,160)
      WRITE(6,185)
      WRITE(6,170) XR1,YR1,XR2,YR2,XTEST,YCL,YCU,ELND
      WRITE(6,190)
      WRITE(6,170) XTEST1,YCL1,YCU1,XTEST2,YCL2,YCU2
      WRITE(6,210) CUTOF1,CUTOFH,CUTOF2
      WRITE(6,160)
      WRITE(6,200) NT,NP,NS1,NH,KND,IW,NX,ICOMP1,IHUB
      WRITE(6,160)
      ALFA = ALFA*PIO180
C *****FORMATS*****
  100 FORMAT(1H0,27X,'MACH',11X,'DYNAMIC PRESSURE',13X,'PRESSURE RATIO',
     1       8X,'DENSITY RATIO',/,14X,'VELOCITY',7X,'NO.',9X,'INC',10X,
     2       'COMP',11X,'INC',8X,'COMP',/)
  101 FORMAT(1H0,27X,'MACH',11X,'DYNAMIC PRESSURE',13X,'PRESSURE RATIO',
```

```
     1 /,14X,'VELOCITY',7X,'NO.',17X,'INC',25X,'INC',/)
 110 FORMAT(3X,'CONTROL   ',1PE10.3,2X,2(1PE10.3,5X,1PE10.3,3X),1PE10.3,
     1      6X,1PE10.3,/)
 111 FORMAT(3X,'CONTROL   ',1PE10.3,2X,1PE1C.3,9X,1PE10.3,20X,1PE10.3,/)
 116 FORMAT(2X,'CONTROL STATION VELOCITY = ',1PE13.4,' EXCEEDS VSONIC.'
     1      ,/,' PROCEECING WITH VC = VSONIC.  IF THE CCNTROL STATION '
     2      'IS NOT AT THE THROAT, RESUBMIT WITH A LOWEF VC.')
 117 FORMAT(2X,'CONTROL STATION VELOCITY = ',1PE13.4,' EXCEEDS VSONIC.'
     1      ,/,' PROCEEDING WITH VS1 = VSONIC.  RESUBMIT WITH A LOWER '
     2      'VS1.')
 118 FORMAT(2X,'CONTROL STATION VELOCITY = ',1PE13.4,' EXCEEDS VSONIC.'
     1      ,/,' PROCEEDING WITH VS2 = VSONIC.  RESUBMIT WITH A LOWER '
     2      'VS2.')
 120 FORMAT(3X,'LOWER',/,3X,'PASSAGE   ',1PE10.3,2X,2(1PE10.3,5X,1PE10.3
     1      ,3X),1PE10.3,6X,1PE10.3,/)
 121 FORMAT(3X,'LOWER',/,3X,'PASSAGE   ',1PE10.3,2X,1PE1C.3,9X,1PE10.3,
     1      20X,1PE10.3,/)
 130 FORMAT(3X,'UPPER',/,3X,'PASSAGE   ',1PE10.3,2X,2(1PE1C.3,5X,1PE10.3
     1      ,3X),1PE10.3,6X,1PE10.3,/)
 131 FORMAT(3X,'UPPER',/,3X,'PASSAGE   ',1PE10.3,2X,1PE1C.3,9X,1PE10.3,
     1      20X,1PE10.3,/)
 140 FORMAT(3X,'FREE ',/,3X,'STREAM    ',1PE10.3,2X,2(1PE1C.3,5X,1PE10.3
     1      ,3X),1PE10.3,6X,1PE10.3,/)
 141 FORMAT(3X,'FREE',/,3X,'STREAM    ',1PE10.3,2X,1PE10.3,9X,1PE10.3,
     1      20X,1PE10.3,/)
 150 FORMAT(/,9X,'ALPHA',9X,'VINF/VC',7X,'VSONIC',8X,'VSONICC',7X,
     1      'WDOTCR',8X,'WDOTLCR',8X,'WDOTUCR')
 155 FORMAT(7X,7(1PE10.3,4X))
 160 FORMAT(/,1X,'----------------------------------------------------'
     1           '-----------------------------------------------',/)
 165 FORMAT(/,9X,'TSTAT',9X,'PSTAT',9X,'PSTATC',8X,'ASTAT',9X,'RHOSTAT'
     1      ,7X,'WDOTC',9X,'WDOTL',7X,'WDOTU')
 170 FORMAT(7X,8(1PE10.3,4X))
 175 FORMAT(/,9X,'VIC',11X,'VICL',10X,'VICU',/,7X,3(1PE1C.3,4X))
 180 FORMAT(/,9X,'TTOT',10X,'PTOT',10X,'PTOTC',9X,'ATOT',10X,'RHOTOT',
     1      8X,'THET',1CX,'DEL')
 185 FORMAT(/,1CX,'XRI1',11X,'YRI1',1CX,'XRI2',9X,'YRI2',9X,'XTEST',10X
     1      ,'YCL',11X,'YCU',10X,'LND')
 190 FORMAT(/,9X,'XTEST1',10X,'YCL1',10X,'YCU1',8X,'XTEST2',10X,'YCL2',
     1      10X,'YCU2')
 200 FORMAT(/,10X,'NT',5X,'NP',5X,'NS1',5X,'NH',5X,'KND',5X,'IW',5X,
     1      'NX',5X,'ICOMP1',5X,'IHUB',/,5X,I3,4X,I3,5X,I3,4X,I3,5X,
     2      I3,4X,I3,5X,I2,7X,I1,9X,I1)
 210 FORMAT(/,7X,'P-S CLTOFF 1',3X,'P-S CUTOFF 2   P-S CUTOFF 3',/,
     1      8X,1PE10.3,5X,1PE10.3,5X,1PE10.3)
      END
```

```
      SUBROUTINE SEARCH
      COMMON /COORD/ XON(700),YON(700),XOFF(200),YOFF(200),S(700),
     1               S1(700),XTEST,XTEST1,XTEST2,YCL,YCU,YCL1,YCL2,YCU1,
     2               YCU2,XR1,XR2,XRH,YR1,YR2,YRH,CUTOF1,CUTOF2,CUTOFH,
     3               ELND,ANG(700),AR(700),AROFF(200)
      COMMON /COUT/ NT,NS1,NH,NP,IW,NX,KND,ICOMP,K,NXHI1,NXHI2,NXHI3,
     1               NS12,NST3,NST7,NPFR(30),IRAK(30),M1,M2,ICOMP1,IPL,
     2               IHUB
      COMMON /CONT/ VC,VS1,VS2,XMC,XMC1,XMC2,WDOTC,WDOTC1,WDOTC2,
     1               TITLE(3),VINF,ALFA,A,B,C,D,A1C,A2C,A3C,A4C,A5C,A11,
     2               A21,A31,A41,A51,A12,A22,A32,A42,A52,VIC,VIC1,VIC2
      COMMON /CONDIT/ TTCTAL,PT,PSTAT,TSTAT,PSTATC,ATOTAL,PTC,RHOST,
     1               RHOTOT,ASTAT,QCINF
C -----------------------------------------------------------------
C
C THIS SUBROUTINE FINDS THE HIGHLIGHT ON EACH BODY AND
C CALCULATES AREAS FOR ALL THE ON- AND OFF-BODY POINTS.
C
      PI = 3.141592654
C
C ON-BODY POINT CALCULATIONS
C
      JJ = NS1+1
      JJJ = NH + 1
      NXHI1=1
      DO 10 J = 2,NS1
      IF(XON(J).GE.XON(J-1)) GO TO 10
      NXHI1 = J
   10 CONTINUE
      IF(IHUB.EQ.0) GO TO 21
      J1 = JJ + 1
      J2 = JJJ-1
      DO 20 J =J1,J2
      IF(XON(J).GE.XON(J-1)) GO TO 20
      NXHI2 = J
      XHI2=XON(NXHI2)
   20 CONTINUE
   21 J2 = JJJ+1
      IF(IHUB.EQ.0) XHI2=99999.0
      DO 30 J = J2,NT
      IF(XON(J).GE.XON(J-1)) GO TO 30
      NXHI3 = J
   30 CONTINUE
      NST3=NXHI1
      CALL SURF
C
C OFF-BODY POINT CALCULATIONS
C
      K =0
      NP1=NP-1
      DO 105 I = 1,NP1
      IF(XOFF(I).LT.XHI2) GO TO 101
      IF(I.EQ.1) GO TO 101
      L=L+1
      IF(L.EQ.L1) GO TO 101
      AA=YOFF(I+1)-YOFF(I)
```

```
      AB=YOFF(I)-YCFF(I-1)
      IF(ABS(AA-AB ).GT.0.001) GO TO 102
  101 IF(XOFF(I).EC.XOFF(I+1)) GO TO 1C5
  102 K=K+1
      NPPR(K) = I
      L=NPPR(K)
      L1=L+1
  105 CONTINUE
      K=K+1
      NPPR(K)=NP
      DO 110 I=1,NP
      IC =0
      IF(I.LE.NPPR(1)) K1=1
      IF(I.LE.NPPR(1)) GC TO 112
      DO 111 J=2,K
      IF((I.LE.NPPR(J)).AND.(I.GT.NPPR(J-1))) K1=J
  111 CONTINUE
  112 IF(XOFF(I).LT.XON(NXHI1)) GO TO 200
      NN=NS1+1
      IF(IHUB.NE.0) CALL INTER(XON,YON,NN,NXHI2,XOFF(I),Y5)
      IF(XOFF(I).GT.XHI2) GO TO 180
      IRAK(K1)= 3
      GO TO 110
  180 IF(YOFF(I).LT.Y5) GO TO 190
      IRAK(K1) = 4
      GO TO 110
  190 IRAK(K1) = 5
      GO TO 110
  200 IRAK(K1) = 1
  110 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ANGLEF
      COMMON /COORD/ XON(700),YON(700),XOFF(200),YOFF(200),S(700),
     1               S1(700),XTEST,XTEST1,XTEST2,YCL,YCH,YCL1,YCL2,YCU1,
     2               YCU2,XR1,XR2,XRH,YR1,YR2,YRH,CUTOF1,CUTOF2,CUTOFH,
     3               ELNE,ANG(700),AR(700),AROFF(200)
      COMMON /COUT/ NT,NS1,NH,NP,IW,NX,KND,ICOMP,K,NXHI1,NXHI2,NXHI3,
     1               NST2,NST3,NST7,NPPF(30),IRAK(30),M1,M2,ICOMP1,IPL,
     2               IHUB
C -----------------------------------------------------------------
C
C  THIS SUBROUTINE CALCULATES THE ANGLE OF THE BODY SURFACE
C
      NS = NS1-1
      DO 10 I = 1,NS
      IF(XON(I+1).EQ.XON(I)) ANG(I)=3.14159/2.0
      IF(XON(I+1).EQ.XON(I)) GO TO 10
      ANG(I) = ATAN((YON(I+1)-YON(I))/(XON(I+1)-XON(I)))
   10 CONTINUE
      NT2 = NH-1
      IF(IHUB.EQ.0) GO TO 21
      N = NS1+1
      DO 20 I = N,NT2
      IF(XON(I+1).EQ.XON(I)) ANG(I)=3.14159/2.0
      IF(XON(I+1).EQ.XON(I)) GO TO 20
      ANG(I) = ATAN((YON(I+1)-YON(I))/(XON(I+1)-XON(I)))
   20 CONTINUE
   21 N = NT2+2
      NT3 = NT-1
      DO 30 I =N,NT3
      IF(XON(I+1).EQ.XON(I)) ANG(I)=3.14159/2.0
      IF(XON(I+1).EQ.XON(I)) GO TO 30
      ANG(I) = ATAN((YON(I+1)-YON(I))/(XON(I+1)-XON(I)))
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SURF
      COMMON /COORD/ XON(700),YON(700),XOFF(200),YOFF(200),S(700),
     1               S1(700),XTEST,XTEST1,XTEST2,YCL,YCU,YCL1,YCL2,YCU1,
     2               YCU2,XR1,XR2,XRH,YR1,YR2,YRH,CUTOF1,CUTOF2,CUTOFH,
     3               ELAC,ANG(700),AR(700),AROFF(200)
      COMMON /COUT/ NT,NS1,NH,NP,IW,NX,KND,ICOMP,K,NXHI1,NXHI2,NXHI3,
     1               NST2,NST3,NST7,NPFF(30),IRAK(30),M1,M2,ICOMP1,IPL,
     2               IHLE
      COMMON/FINDER/ JCHK1,JCHK2,JCHK3
C ---------------------------------------------------------------------
C
C  THIS SUBROUTINE CALCULATES THE SURFACE DISTANCE ON EACH BODY
C  FROM ITS HIGHLIGHT
C
      CALL CHECK(1,NS1,IRT,XON)
      XRS=XR1
      IF(XON(IRT).GT.XR1) XR1=XON(IRT)
      DO 2 I=2,NS1
      IF((((YON(I).GE.YR1).AND.(YON(I-1).LE.YR1)).OR.((YCN(I).LE.YR1).
     1 AND.(YON(I-1).GE.YR1))).AND.(((XON(I).LE.XR1).AND.(XON(I-1).GE.
     2 XR1)).OR.((XON(I).GE.XR1).AND.(XON(I-1).LE.XR1)))) JCHK1 = I
    2 CONTINUE
      XR1=XRS
      IF(IHUB.EQ.0) GO TO 5
      KK=NS1+2
      K2=KK-1
      CALL CHECK(K2,NH,IR,XON)
      XRS=XRH
      IF(XON(IR).GT.XRH) XRH=XON(IR)
      DO 4 I=KK,NH
      IF((((YON(I).GE.YRH).AND.(YON(I-1).LE.YRH)).OR.((YCN(I).LE.YRH).
     1 AND.(YON(I-1).GE.YRH))).AND.(((XON(I).LE.XRH).AND.(XON(I-1).GE.
     2 XRH)).OR.((XON(I).GE.XRH).AND.(XON(I-1).LE.XRH)))) JCHK2 = I
    4 CONTINUE
      XRH=XRS
    5 NK=NH+1
      NK2=NK+1
      CALL CHECK(NK,NT,IR2,XON)
      XRS=XR2
      IF(IR2.EQ.0) IR2=NT
      IF(XON(IR2).GT.XR2) XR2=XON(IR2)
      DO 6 I=NK2,NT
      IF((((YON(I).GE.YR2).AND.(YON(I-1).LE.YR2)).OR.((YCN(I).LE.YR2).
     1 AND.(YON(I-1).GE.YR2))).AND.(((XON(I).LE.XR2).AND.(XON(I-1).GE.
     2 XR2)).OR.((XON(I).GE.XR2).AND.(XON(I-1).LE.XR2)))) JCHK3 = I
    6 CONTINUE
      XR2=XRS
      IF(YON(JCHK1).GE.YR1) GO TO 10
      S(JCHK1)= -SQRT((XCN(JCHK1)-XR1)**2+(YON(JCHK1)-YR1)**2)
      NN1 = JCHK1+2
      NN2 = JCHK1-1
      S(NN1-1)=  SQRT((XCN(NN1-1)-XR1)**2+(YON(NN1-1)-YR1)**2)
      GO TO 20
   10 S(JCHK1) = SQRT((XCN(JCHK1)-XR1)**2+(YON(JCHK1)-YR1)**2)
      NN1 = JCHK1+1
      NN2 = JCHK1-2
```

```
      S(NN2+1) =-SQRT((XCN(NN2+1)-XR1)**2+(YON(NN2+1)-YR1)**2)
   20 DO 30 I = NN1,NS1
      S(I) = S(I-1)+SQRT((XON(I)-XON(I-1))**2+(YON(I)-YON(I-1))**2)
   30 CONTINUE
      DO 40 II = 1,NN2
      I = NN2+1-II
      S(I) = S(I+1)-SQRT((XON(I)-XON(I+1))**2+(YON(I)-YOA(I+1))**2)
   40 CONTINUE
      IF(IHUB.EQ.0) GO TO 81
      IF(YON(JCHK2).GT.YRH) GO TO 50
      S(JCHK2) =-SQRT((XCN(JCHK2)-XRH)**2+(YON(JCHK2)-YRH)**2)
      NN1 = JCHK2+2
      NN2 = JCHK2-1
      S(NN1-1) = SQRT((XCN(NN1-1)-XRH)**2+(YON(NN1-1)-YRH)**2)
      GO TO 60
   50 S(JCHK2) = SQRT((XCN(JCHK2)-XRH)**2+(YON(JCHK2)-YRH)**2)
      NN1 = JCHK2+1
      NN2 = JCHK2-2
      S(NN2+1) =-SQRT((XCN(NN2+1)-XRH)**2+(YON(NN2+1)-YRH)**2)
   60 IS = NS1+1
      IS2 = NH
      DO 70 I = NN1,IS2
      S(I) = S(I-1)+SQRT((XON(I)-XON(I-1))**2+(YON(I)-YOA(I-1))**2)
   70 CONTINUE
      DO 80 II = IS,NN2
      I = NN2+IS-II
      S(I) = S(I+1)-SQRT((XON(I)-XON(I+1))**2+(YON(I)-YOA(I+1))**2)
   80 CONTINUE
   81 IF(YON(JCHK3).LT.YR2) GO TO 90
      S(JCHK3) =-SQRT((XCN(JCHK3)-XR2)**2+(YON(JCHK3)-YR2)**2)
      NN1 = JCHK3+1
      NN2 = JCHK3-2
      S(NN2+1) = SQRT((XCN(NN2+1)-XR2)**2+(YON(NN2+1)-YR2)**2)
      GO TO 100
   90 S(JCHK3) = SQRT((XCN(JCHK3)-XR2)**2+(YON(JCHK3)-YR2)**2)
      NN1 = JCHK3+2
      NN2 = JCHK3-1
      S(NN1-1) =-SQRT((XCN(NN1-1)-XR2)**2+(YON(NN1-1)-YR2)**2)
  100 IS = NH+1
      DO 110 I=NN1,NT
      S(I) = S(I-1)-SQRT((XON(I)-XON(I-1))**2+(YON(I)-YOA(I-1))**2)
  110 CONTINUE
      DO 120 II = IS,NN2
      I = NN2+IS-II
      S(I) = S(I+1)+SQRT((XON(I)-XON(I+1))**2+(YON(I)-YOA(I+1))**2)
  120 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CHECK(IS,IF,IRT,XO)
      DIMENSION XO(500)
      IT=0
      IST=IS+1
      DO 5 I=IST,IF
      IF(IT.EQ.1) GO TO 5
      IF(XO(I).GT.XO(I-1)) IRT=I-1
      IF(XO(I).CT.XO(I-1)) IT=1
    5 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SOLVE
      COMMON /CCORD/ XON(700),YON(700),XOFF(200),YOFF(200),S(700),
     1               S1(700),XTEST,XTEST1,XTEST2,YCL,YCU,YCL1,YCL2,YCU1,
     2               YCU2,XR1,XR2,XRH,YR1,YR2,YRH,CUTOF1,CUTOF2,CUTOFH,
     3               ELNC,ANG(700),AR(700),AROFF(200)
      COMMON /VELOC/ V1(700),V2(700),V3(700),V4(700),V5(700),V1X(200),
     1               V2X(200),V3X(200),V4X(200),V5X(200),V1Y(200),
     2               V2Y(200),V3Y(200),V4Y(200),V5Y(200)
      COMMON /CONT/ VC,VS1,VS2,XMC,XMC1,XMC2,WDOTC,WDOTC1,WDOTC2,
     1               TITLE(3),VINF,ALFA,A,B,C,D,A1C,A2C,A3C,A4C,A5C,A11,
     2               A21,A31,A41,A51,A12,A22,A32,A42,A52,VIC,VIC1,VIC2
      COMMON /COUT/ NT,NS1,NH,NP,IW,NX,NND,ICOMP,K,NXHI1,NXHI2,NXHI3,
     1               NST2,NST3,NST7,NPPR(30),IRAK(30),M1,M2,ICOMP1,IPL,
     2               IHUB
      COMMON /CONDIT/ TTOTAL,PT,PSTAT,TSTAT,PSTATC,ATOTAL,PTC,RHOST,
     1               RHOTOT,ASTAT,QCINF,RSORTC
      COMMON /WRIT/  AA1C,AA2C,AA3C,AA4C,AA5C,AA11,AA21,AA31,AA41,AA51,
     1               AA12,AA22,AA32,AA42,AA52
      COMMON /SOLUT/ VBAR(700),VBARO(200),VINC(700),VXINC(200),
     1               VYINC(200),RHOB(700),RBORT(700),RHOBO(200),
     2               VCOM(700),RBOOT(200),VRE(200),VRECOM(200),
     3               VXCOM(200),VYCOM(200),THETA(200),PSCPTC(700),
     4               PSOPT(700),CMACH(700),XMACH(700),CPI(700),CPC(700),
     5               RHOI(700)
C ----------------------------------------------------------------------
C
C  THIS SUBROUTINE SOLVES FOR THE COEFFICIENTS A,B,C, AND D AND
C  SOLVES FOR VBAR AND V INCOMPRESSIBLE FOR BOTH ON- AND OFF-BODY
C  POINTS
C
      DIMENSION YNEW(50),VNEW1(50),VNEW2(50),VNEW3(50),VNEW4(50),
     1          VNEW5(50)
      G = 32.174
      IF(M1.EQ.1) GO TO 30
C  THE FOLLOWING CALCULATIONS ARE FOR THE CONTROL STATION
C  UPSTREAM OF THE HUB
      A = XTEST
      NPL = 0
      NH1 = NH+1
      DO 5 I10=NXHI1,NS1
      IF((A.GE.XON(I10)).AND.(A.LT.XON(I10+1))) THEAL = ANG(I10)
    5 CONTINUE
      DO 6 I10=NH1,NXHI3
      IF((A.LE.XON(I10)).AND.(A.GT.XON(I10+1))) THEAH = ANG(I10)
    6 CONTINUE
      CALL INTER2(NXHI1,NS1,A,YL,V1L,V2L,V3L,V4L,V5L)
      CALL INTER2(NH1,NXHI3,A,YH,V1H,V2H,V3H,V4H,V5H)
      DO 10 II = 1,K
      IF(II.EQ.1) I=1
      IF(II.EQ.1) GO TO 12
      I=NPPR(II-1)+1
   12 IF(A.NE.XOFF(I)) GO TO 10
      IF(II.EQ.1) NPL = 1
      IF(II.EQ.1) NPH = NPPR(1)
      IF(II.EQ.1) J3 = NPPR(1)
      DO 15 J =2,K
```

```
      IF(II.EQ.J) NPL = I
      IF(II.EQ.J) NPH = NPPR(J)-NPPR(J-1)
      IF(II.EQ.J)  J3 = NPPR(J)
   15 CONTINUE
   10 CONTINUE
      IF(NPL.EQ.0) WRITE(6,1000)
      IF(NPL.EQ.0) STOP
      DO 20 JJ = NPL,J3
      YNEW(JJ+2-NPL)  = YOFF(JJ)
      VNEW1(JJ+2-NPL) = V1X(JJ)
      VNEW2(JJ+2-NPL) = V2X(JJ)
      VNEW3(JJ+2-NPL) = V3X(JJ)
      VNEW4(JJ+2-NPL) = V4X(JJ)
      IF(IHUB.NE.0) VNEW5(JJ+2-NPL) = V5X(JJ)
   20 CONTINUE
      YNEW(1) = YL
      VNEW1(1) = V1L*COS(THEAL)
      VNEW2(1) = V2L*COS(THEAL)
      VNEW3(1) = V3L*COS(THEAL)
      VNEW4(1) = V4L*COS(THEAL)
      IF(IHUB.NE.0) VNEW5(1) = V5L*COS(THEAL)
      YNEW(NPH+2) = YH
      VNEW1(NPH+2) = -V1H*COS(THEAH)
      VNEW2(NPH+2) = -V2H*COS(THEAH)
      VNEW3(NPH+2) = -V3H*COS(THEAH)
      VNEW4(NPH+2) = -V4H*COS(THEAH)
      IF(IHUB.NE.0) VNEW5(NPH+2) = -V5H*COS(THEAH)
      NPT = NPH+2
      CALL INTEG(VNEW1,YNEW,AA1C,NPT)
      CALL INTEG(VNEW2,YNEW,AA2C,NPT)
      CALL INTEG(VNEW3,YNEW,AA3C,NPT)
      CALL INTEG(VNEW4,YNEW,AA4C,NPT)
      IF(IHUB.NE.0) CALL INTEG(VNEW5,YNEW,AA5C,NPT)
      AROFF(NPL)=YCU-YCL
      A1C = AA1C/AROFF(NPL)
      A2C = AA2C/AROFF(NPL)
      A3C = AA3C/AROFF(NPL)
      A4C = AA4C/AROFF(NPL)
      IF(IHUB.NE.0) A5C = AA5C/AROFF(NPL)
      IF(IHUB.EQ.0) GO TO 215
C  THE FOLLOWING CALCULATIONS ARE FOR THE LOWER CONTROL STATION
   30 IF(M1.EQ.1) GO TO 40
      IF(M2.EQ.1) GO TO 70
   40 A = XTEST1
      NPL = 0
      NH1 = NS1+1
      CALL INTER2(NST3,NS1,A,YNEW(1),VNEW1(1),VNEW2(1),VNEW3(1),VNEW4(1)
     1            ,VNEW5(1))
      DO 41 I10=NST3,NS1
      IF((A.GE.XON(I10)).AND.(A.LT.XON(I10+1))) THEAL = ANG(I10)
   41 CONTINUE
      DO 42 I10=NH1,NXHI2
      IF((A.LE.XON(I10)).AND.(A.GT.XON(I10+1))) THEAH = ANG(I10)
   42 CONTINUE
      VNEW1(1) = VNEW1(1)*COS(THEAL)
      VNEW2(1) = VNEW2(1)*COS(THEAL)
      VNEW3(1) = VNEW3(1)*COS(THEAL)
```

```
      VNEW4(I) = VNEW4(I)*COS(THEAL)
      VNEW5(I) = VNEW5(I)*COS(THEAL)
      CALL INTER2(NH1,NXHI2,A,YH,V1H,V2H,V3H,V4H,V5H)
      DO 50 II = 1,K
      IF(II.EQ.1) I=1
      IF(II.EQ.1) GO TO 45
      I = NPPR(II-1)+1
   45 IF((A.NE.XOFF(I)).OR.(YOFF(I).GT.YCU1)) GO TO 50
      IF(II.EQ.1) NPL = 1
      IF(II.EQ.1) NPH = NPPR(1)
      IF(II.EQ.1)  J3 = NPPR(1)
      DO 46 J=2,K
      IF(II.EQ.J) NPL = NPPR(J-1)+1
      IF(II.EQ.J) NPH = NPPR(J)-NPPR(J-1)
      IF(II.EQ.J)  J3 = NPPR(J)
   46 CONTINUE
   50 CONTINUE
      IF(NPL.EQ.0) WRITE(6,1001)
      IF(NPL.EQ.0) STOP
      DO 60 JJ = NPL,J3
      YNEW(JJ+2-NPL) = YOFF(JJ)
      VNEW1(JJ+2-NPL) = V1X(JJ)
      VNEW2(JJ+2-NPL) = V2X(JJ)
      VNEW3(JJ+2-NPL) = V3X(JJ)
      VNEW4(JJ+2-NPL) = V4X(JJ)
      VNEW5(JJ+2-NPL) = V5X(JJ)
   60 CONTINUE
      YNEW(NPH+2) = YH
      VNEW1(NPH+2) = -V1H*COS(THEAH)
      VNEW2(NPH+2) = -V2H*COS(THEAH)
      VNEW3(NPH+2) = -V3H*COS(THEAH)
      VNEW4(NPH+2) = -V4H*COS(THEAH)
      VNEW5(NPH+2) = -V5H*COS(THEAH)
      NPT=NPH+2
      CALL INTEG(VNEW1,YNEW,AA11,NPT)
      CALL INTEG(VNEW2,YNEW,AA21,NPT)
      CALL INTEG(VNEW3,YNEW,AA31,NPT)
      CALL INTEG(VNEW4,YNEW,AA41,NPT)
      CALL INTEG(VNEW5,YNEW,AA51,NPT)
      AROFF(NPL)=YCU1-YCL1
      A11 = AA11/AROFF(NPL)
      A21 = AA21/AROFF(NPL)
      A31 = AA31/AROFF(NPL)
      A41 = AA41/AROFF(NPL)
      A51 = AA51/AROFF(NPL)
C   THE FOLLOWING CALCULATIONS ARE FOR THE UPPER CONTROL STATION
   70 IF((M1.EQ.0).AND.(M2.EQ.0)) GO TO 80
      A = XTEST2
      NPL = 0
      NH1 = NH+1
      NSH= NH
      CALL INTER2(NXHI2,NSH,A,YNEW(1),VNEW1(1),VNEW2(1),VNEW3(1),
     1            VNEW4(1),VNEW5(1))
      DO 71 I10 = NXHI2,NSH
      IF((A.GE.XON(I10)).AND.(A.LT.XON(I10+1))) THEAL = ANG(I10)
   71 CONTINUE
      DO 72 I10=NH1,NXHI2
```

```
      IF((A.LE.XON(I10)).AND.(A.GT.XON(I10+1))) THEAH = ANG(I10)
   72 CONTINUE
      VNEW1(1) = VNEW1(1)*COS(THEAL)
      VNEW2(1) = VNEW2(1)*COS(THEAL)
      VNEW3(1) = VNEW3(1)*COS(THEAL)
      VNEW4(1) = VNEW4(1)*COS(THEAL)
      VNEW5(1) = VNEW5(1)*COS(THEAL)
      DO 90 II = 1,K
      IF(II.EQ.1) I=1
      IF(II.EQ.1) GO TO 95
      I = NPPR(II-1)+1
   95 IF((A.NE.XOFF(I)).OR.(IRAK(II).NE.4)) GO TO 90
      IF(II.EQ.1) NPL=1
      IF(II.EQ.1) NPH = NPPR(1)
      IF(II.EQ.1)  J3 = NPPR(1)
      DO 96 J=2,K
      IF(II.EQ.J) NPL=I
      IF(II.EQ.J) NPH = NPPR(J)-NPPR(J-1)
      IF(II.EQ.J)  J3 = NPPR(J)
   96 CONTINUE
   90 CONTINUE
      IF(NPL.EQ.0) WRITE(6,1002)
      IF(NPL.EQ.0) STOP
      NPT = NPH+2
      CALL INTER2(NH1,NX+I3,A,YNEW(NPT),VNEW1(NPT),VNEW2(NPT),VNEW3(NPT)
     1            ,VNEW4(NPT),VNEW5(NPT))
      VNEW1(NPT) = -VNEW1(NPT)*COS(THEAH)
      VNEW2(NPT) = -VNEW2(NPT)*COS(THEAH)
      VNEW3(NPT) = -VNEW3(NPT)*COS(THEAH)
      VNEW4(NPT) = -VNEW4(NPT)*COS(THEAH)
      VNEW5(NPT) = -VNEW5(NPT)*COS(THEAH)
      DO 100 JJ = NPL,J3
      YNEW(JJ+2-NPL) = YOFF(JJ)
      VNEW1(JJ+2-NPL) = V1X(JJ)
      VNEW2(JJ+2-NPL) = V2X(JJ)
      VNEW3(JJ+2-NPL) = V3X(JJ)
      VNEW4(JJ+2-NPL) = V4X(JJ)
      VNEW5(JJ+2-NPL) = V5X(JJ)
  100 CONTINUE
      CALL INTEG(VNEW1,YNEW,AA12,NPT)
      CALL INTEG(VNEW2,YNEW,AA22,NPT)
      CALL INTEG(VNEW3,YNEW,AA32,NPT)
      CALL INTEG(VNEW4,YNEW,AA42,NPT)
      CALL INTEG(VNEW5,YNEW,AA52,NPT)
      AROFF(NPL)=YCU2-YCL2
      A12 = AA12/AROFF(NPL)
      A22 = AA22/AROFF(NPL)
      A32 = AA32/AROFF(NPL)
      A42 = AA42/AROFF(NPL)
      A52 = AA52/AROFF(NPL)
   80 IF(M1.EQ.1) GO TO 200
      IF(M2.EQ.1) GO TO 210
C  THE FOLLOWING IS FOR THE CASE WHERE UPSTREAM AND LOWER
C  CONTROL STATIONS ARE INPUT
      VCL = A3C-A4C
      VL1= A31-A41
      VU1 = A51-A41
```

```
      VCU = A5C-A4C
      A = VINF*COS(ALFA)
      B = VINF*SIN(ALFA)
      D= (VS1*VCL-VC*VL1+(A*A1C+B*A2C)*VL1-(A*A11+B*A21)*VCL)/(VU1*VCL-
     1   VCU*VL1)
      C = (VS1-(A*A11+B*A21)-D*VU1)/VL1
      GO TO 220
C  THE FOLLOWING IS FOR THE CASE WHERE UPPER AND LOWER
C  CONTROL STATIONS ARE INPUT
  200 A = VINF*COS(ALFA)
      B = VINF*SIN(ALFA)
      VL2 = A32-A42
      VL1 = A31-A41
      VU1 = A51-A41
      VU2 = A52-A42
      D = (VS1*VL2-VS2*VL1+(A*A12+B*A22)*VL1-(A*A11+B*A21)*VL2)/(VU1*VL2
     1    -VU2*VL1)
      C = (VS1-(A*A11+B*A21)-D*VU1)/VL1
      GO TO 220
C  THE FOLLOWING IS FOR THE CASE WHERE UPSTREAM AND UPPER
C  CONTROL STATIONS ARE INPUT
  210 A = VINF*COS(ALFA)
      B = VINF*SIN(ALFA)
      VCL = A3C-A4C
      VL2 = A32-A42
      VU2 = A52-A42
      VCU = A5C-A4C
      D = (VS2*VCL-VC*VL2+(A*A1C+B*A2C)*VL2-(A*A12+B*A22)*VCL)/(VU2*VCL-
     1    VCU*VL2)
      C = (VS2-(A*A12+B*A22)-D*VU2)/VL2
      GO TO 220
  215 A = VINF*COS(ALFA)
      B = VINF*SIN(ALFA)
      C = (VC-A*A1C-B*A2C)/(A3C-A4C)
C  THE FOLLOWING ARE THE CALCULATIONS FOR V AND VBAR
  220 DO 230 I=1,NT
      IF(IHUB.EQ.0) VINC(I)=A*V1(I)+B*V2(I)+C*(V3(I)-V4(I))
      IF(IHUB.EQ.0) GO TO 230
      VINC(I) = A*V1(I)+B*V2(I)+C*(V3(I)-V4(I))+D*(V5(I)-V4(I))
  230 CONTINUE
      DO 240 I =1,NP
      IF(IHUB.EQ.0) VXINC(I)=A*V1X(I)+B*V2X(I)+C*(V3X(I)-V4X(I))
      IF(IHUB.EQ.0) VYINC(I)=A*V1Y(I)+B*V2Y(I)+C*(V3Y(I)-V4Y(I))
      IF(IHUB.EQ.0) GO TO 235
      VXINC(I)= A*V1X(I)+B*V2X(I)+C*(V3X(I)-V4X(I))+D*(V5X(I)-V4X(I))
      VYINC(I)= A*V1Y(I)+B*V2Y(I)+C*(V3Y(I)-V4Y(I))+D*(V5Y(I)-V4Y(I))
  235 VRE(I) = SQRT(VXINC(I)**2+VYINC(I)**2)
      THETA(I) = ATAN(VYINC(I)/VXINC(I))/3.141592654*180.0
  240 CONTINUE
 1000 FORMAT(///,10X,'THERE IS NO RAKE LOCATED AT THE DOWNSTREAM '
     1       'CONTROL STATION.  ',/,10X,'CHECK INPUT LOCATION OF '
     2       'CONTROL STATION.  RUN TERMINATED.')
 1001 FORMAT(///,10X,'THERE IS NO RAKE LOCATED AT THE LOWER '
     1       'CONTROL STATION.  ',/,10X,'CHECK INPUT LOCATION OF '
     2       'CONTROL STATION.  RUN TERMINATED.')
 1002 FORMAT(///,10X,'THERE IS NO RAKE LOCATED AT THE UPPER '
     1       'CONTROL STATION.  ',/,10X,'CHECK INPUT LOCATION OF '
     2       'CONTROL STATION.  RUN TERMINATED.')
      RETURN
      END
```

```
      SUBROUTINE COMCOR
      COMMON /COUT/ NT,NS1,NH,NP,IW,NX,KND,ICOMP,K,NXHI1,NXHI2,NXHI3,
     1              NST2,NST3,NST7,NPPR(30),IRAK(30),M1,P2,ICOMP1,IPL,
     2              IHUB
      COMMON /CONDIT/ TTOTAL,PT,PSTAT,TSTAT,PSTATC,ATOTAL,PTC,RHOST,
     1              RHOTOT,ASTAT,QCINF,RSORTC
      COMMON /SOLUT/ VBAR(700),VBARO(200),VINC(700),VXINC(200),
     1              VYINC(200),RHOB(700),RBORT(700),RHOBC(200),
     2              VCOM(700),RBOOT(200),VRE(200),VRECOM(200),
     3              VXCOM(200),VYCOM(200),THETA(200),PSCPTC(700),
     4              PSOPT(700),CMACH(700),XMACH(700),CPI(700),CPC(700),
     5              RHCI(700)
      COMMON /CONT/ VC,VS1,VS2,XMC,XMC1,XMC2,WDOTC,WDOTC1,WDOTC2,
     1              TITLE(3),VINF,ALFA,A,B,C,D,A1C,A2C,A3C,A4C,A5C,A11,
     2              A21,A31,A41,A51,A12,A22,A32,A42,A52,VIC,VIC1,VIC2
C --------------------------------------------------------------------
C
C THIS SUBROUTINE APPLIES THE COMPRESSIBLITY CORRECTION TO THE SOLUTION
C
      EXCON = 1.095*ATOTAL*125.0/216.0
      PI = 3.141592654
      PI180 = 180.0/PI
      VCHK = ATOTAL/SQRT(1.2)
      DO 10 I=1,NT
      A9 = 1.0
      VCOM(I) = VINC(I)/RSORTC**(ABS(VINC(I)/VC))
      VA = 0.2*(VCOM(I)/ATOTAL)**2
      IF(VA.GT.1.0) GO TO 25
      PSO = (1.0-0.2*(VCOM(I)/ATOTAL)**2)**3.5
      GO TO 30
   25 PSO = 0.0
   30 IF((ABS(VCOM(I)).LT.VCHK).OR.(ABS(NX).NE.1)) GO TO 10
      VSAVE = ABS(VCOM(I))
      IF((VSAVE/VCHK).GT.2.0) GO TO 10
      RHORTC = VSAVE*PSO**0.715/EXCON
      IF(RHORTC.EQ.0.0) RHORTC = 1.0
      IF(VCOM(I).LT.0.0) A9=-1.0
      VCOM(I) = VCHK*(1.0+(VSAVE/VCHK-1.0)**(1.0/RHORTC))*A9
   10 CONTINUE
      DO 20 I =1,NP
      VRECOM(I) = VRE(I)/RSORTC**(VRE(I)/VC)
      VA = 0.2*(VRECOM(I)/ATOTAL)**2
      IF(VA.GT.1.0) GO TO 35
      PSI = (1.0-0.2*(VRECOM(I)/ATOTAL)**2)**3.5
      GO TO 40
   35 PSI = 0.0
   40 IF((VRECOM(I).LT.VCHK).OR.(ABS(NX).NE.1)) GO TO 15
      VSAVE = VRECOM(I)
      IF((VSAVE/VCHK).GT.2.0) GO TO 15
      RHORTC = VSAVE*PSI**0.715/EXCON
      IF(RHORTC.EQ.0.0) RHORTC =1.0
      VRECOM(I) = VCHK*(1.0+(VSAVE/VCHK-1.0)**(1.0/RHORTC))
   15 VXCOM(I) = VXINC(I)*VRECOM(I)/VRE(I)
      VYCOM(I) = VYINC(I)*VRECOM(I)/VRE(I)
      THETA(I) = ATAN(VYCOM(I)/VXCOM(I))*PI180
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ONBODY
      COMMON /COORD/ XON(700),YON(700),XOFF(200),YOFF(20E),S(700),
     1               S1(700),XTEST,XTEST1,XTEST2,YCL,YCU,YCL1,YCL2,YCU1,
     2               YCU2,XR1,XR2,XRH,YP1,YR2,YRH,CUTOF1,CUTOF2,CUTOFH,
     3               ELND,ANG(700),AR(7C0),AROFF(200)
      COMMON /CONT/ VC,VS1,VS2,XMC,XMC1,XMC2,WDOTC,WDOTC1,WDOTC2,
     1               TITLE(3),VINF,ALFA,A,B,C,D,A1C,A2C,A3C,A4C,A5C,A11,
     2               A21,A31,A41,A51,A12,A22,A32,A42,A52,VIC,VIC1,VIC2
      COMMON /COUT/ NT,NS1,NH,NP,IW,NX,KND,ICOMP,K,NXHI1,NXHI2,NXHI3,
     1               NST2,NST3,NST7,NPFR(30),IRAK(30),M1,M2,ICOMP1,IPL,
     2               IHUB
      COMMON /CONDIT/ TTCTAL,PT,PSTAT,TSTAT,PSTATC,ATOTAL,PTC,RHOST,
     1               RHOTOT,ASTAT,QCINF,RSORTC
      COMMON /SOLUT/ VBAR(700),VBARO(2CC),VINC(700),VXINC(200),
     1               VYINC(200),RHOB(7C0),RBORT(700),RHOBO(200),
     2               VCOM(700),RBOOT(2CC),VRE(200),VRECOP(200),
     3               VXCOM(200),VYCOM(200),THETA(200),PSCPTC(700),
     4               PSOPT(700),CMACH(700),XMACH(700),CPI(700),CPC(700),
     5               RHOI(700)
C -----------------------------------------------------------------
C
C  THIS SUBROUTINE CALCULATES THE ON-BODY PROPERTIES
C
      DIMENSION DIMDUM(5)
      DO 10 I =1,NT
      VCONC = 0.2*(ABS(VCOM(I)/ATOTAL))**2
      VCON  = 0.2*(ABS(VINC(I)/ATOTAL))**2
      IF(VINF.EQ.0.0) CPI(I) = 9999.J
      IF(VINF.EQ.0.0) CPC(I) = 9999.0
      IF(VINF.EQ.0.0) GO TO 4
      CPI(I) = 1.0-(ABS(VINC(I)/VINF))**2
      CPC(I) = (PT-PSTAT-(0.5*RHOB(I)*VCOM(I)**2))/QCINF
    4 IF(VCONC.GT.1.0) PSOPTC(I) = 0.0
      IF(VCONC.GT.1.0) CMACH(I) = 999.C
      IF(VCON.GT.1.0) XMACH(I) = 999.0
      IF(VCON.GT.1.0) PSCPT(I) = 0.0
      IF(VCON.GT.1.0) RHCI(I) = 2000.0
      IF((VCONC.GT.1.0).AND.(VCON.GT.1.0)) GO TO 10
      IF(VCONC.GT.1.0) GO TO 5
      PSOPTC(I) = (1.0-VCONC)**3.5
      CMACH(I) = ABS(VCCM(I))/ATOTAL/SCRT(1.0-VCONC)
      IF(VCON.GT.1.0) GO TO 10
    5 PSOPT(I)  = (1.0-0.5*RHOTOT*VINC(I)**2/PT)
      XMACH(I) = ABS(VINC(I))/ATOTAL/SCRT(1.0-VCON )
   10 CONTINUE
      WRITE(6,100)
      J=0
      DO 15 I=1,NS1
      IF(ICOMP.EQ.0) GO TO 11
      IF((CMACH(I).LT.1.C).OR.(ABS(NX).NE.1)) GO TO 12
      IF(CMACH(I-1).LT.1.0) WRITE(6,16C) I
      IF(CMACH(I-1).LT.1.0) J = 3
      GO TO 12
   11 IF((XMACH(I).LT.1.C).OR.(ABS(NX).NE.1)) GO TO 12
      IF(XMACH(I-1).LT.1.0) WRITE(6,160) I
      IF(XMACH(I-1).LT.1.0) J = 3
```

```
   12 IF(ICOMP.EQ.0) GO TO 13
      IF((CMACH(I).LT.1.0).AND.(CMACH(I-1).GE.1.0).AND.(ABS(NX).EQ.1))
     1    WRITE(6,170) I
      IF((CMACH(I).LT.1.0).AND.(CMACH(I-1).GE.1.0).AND.(ABS(NX).EQ.1))
     1      J = J+2
      GO TO 15
   13 IF((XMACH(I).LT.1.0).AND.(XMACH(I-1).GE.1.0).AND.(ABS(NX).EQ.1))
     1    WRITE(6,170) I
      IF((XMACH(I).LT.1.0).AND.(XMACH(I-1).GE.1.0).AND.(ABS(NX).EQ.1))
     1  J =J+2
   15 CONTINUE
      WRITE(6,105)
      DO 20 I = 1,NS1
      J=J+1
      IF(J.EQ.51) WRITE(6,95)
      IF(J.EQ.51) WRITE(6,105)
      IF(J.EQ.51) J=1
      IF(ICOMP.EQ.0) WRITE(6,110) I,XON(I),YON(I),S(I),VINC(I),XMACH(I),
     1               CPI(I),PSOPT(I)
      IF(ICOMP.EQ.1) WRITE(6,110) I,XON(I),YON(I),S(I),VCCM(I),
     1               CMACH(I),CPC(I),PSOPTC(I)
   20 CONTINUE
      IF(IHUB.EQ.0) GO TO 26
      WRITE(6,120)
      IS = NS1+1
      IST = NH
      J=0
      DO 25 I=IS,IST
      IF(ICOMP.EQ.0) GO TO 21
      IF((CMACH(I).LT.1.0).OR.(ABS(NX).NE.1)) GO TO 22
      IF(CMACH(I-1).LT.1.0) WRITE(6,160) I
      IF(CMACH(I-1).LT.1.0) J = 3
      GO TO 22
   21 IF((XMACH(I).LT.1.0).OR.(ABS(NX).NE.1)) GO TO 22
      IF(XMACH(I-1).LT.1.0) WRITE(6,160) I
      IF(XMACH(I-1).LT.1.0) J = 3
   22 IF(ICOMP.EQ.0) GO TO 23
      IF((CMACH(I).LT.1.0).AND.(CMACH(I-1).GE.1.0).AND.(ABS(NX).EQ.1))
     1    WRITE(6,170) I
      IF((CMACH(I).LT.1.0).AND.(CMACH(I-1).GE.1.0).AND.(ABS(NX).EQ.1))
     1      J = J+2
      GO TO 25
   23 IF((XMACH(I).LT.1.0).AND.(XMACH(I-1).GE.1.0).AND.(ABS(NX).EQ.1))
     1    WRITE(6,170) I
      IF((XMACH(I).LT.1.0).AND.(XMACH(I-1).GE.1.0).AND.(ABS(NX).EQ.1))
     1  J =J+2
   25 CONTINUE
      WRITE(6,105)
      DO 30 I = IS,IST
      J=J+1
      IF(J.EQ.51) WRITE(6,95)
      IF(J.EQ.51) WRITE(6,105)
      IF(J.EQ.51) J=1
      IF(ICOMP.EQ.0) WRITE(6,110) I,XON(I),YON(I),S(I),VINC(I),XMACH(I),
     1               CPI(I),PSOPT(I)
      IF(ICOMP.EQ.1) WRITE(6,110) I,XON(I),YON(I),S(I),VCCM(I),
     1               CMACH(I),CPC(I),PSOPTC(I)
```

```
   30 CONTINUE
   26 IS = NH+1
      IF(IHUB.EQ.0) WRITE(6,131)
      IF(IHUB.NE.0) WRITE(6,130)
      J=0
      DO 35 I=IS,NT
      IF(ICOMP.EQ.0) GO TO 31
      IF((CMACH(I).LT.1.0).OR.(ABS(NX).NE.1)) GO TO 32
      IF(CMACH(I-1).LT.1.0) WRITE(6,160) I
      IF(CMACH(I-1).LT.1.0) J = 3
      GO TO 32
   31 IF((XMACH(I).LT.1.0).OR.(ABS(NX).NE.1)) GO TO 32
      IF(XMACH(I-1).LT.1.0) WRITE(6,160) I
      IF(XMACH(I-1).LT.1.0) J = 3
   32 IF(ICOMP.EQ.0) GO TO 33
      IF((CMACH(I).LT.1.0).AND.(CMACH(I-1).GE.1.0).AND.(ABS(NX).EQ.1))
     1    WRITE(6,170) I
      IF((CMACH(I).LT.1.0).AND.(CMACH(I-1).GE.1.0).AND.(ABS(NX).EQ.1))
     1     J = J+2
      GO TO 35
   33 IF((XMACH(I).LT.1.0).AND.(XMACH(I-1).GE.1.0).AND.(ABS(NX).EQ.1))
     1    WRITE(6,170) I
      IF((XMACH(I).LT.1.0).AND.(XMACH(I-1).GE.1.0).AND.(ABS(NX).EQ.1))
     1   J =J+2
   35 CONTINUE
      WRITE(6,105)
      DO 40 I=IS,NT
      J=J+1
      IF(J.EQ.51) WRITE(6,95)
      IF(J.EQ.51) WRITE(6,105)
      IF(J.EQ.51) J=1
      IF(ICOMP.EQ.0) WRITE(6,110) I,XON(I),YON(I),S(I),VINC(I),XMACH(I),
     1                CPI(I),PSOPT(I)
      IF(ICOMP.EQ.1) WRITE(6,110) I,XON(I),YON(I),S(I),VCOM(I),
     1                CMACH(I),CPC(I),PSOPTC(I)
   40 CONTINUE
C     THE FOLLOWING WRITES DATA TAPES FOR THE BOUNDARY LAYER PROGRAM
      ISTAG = 0
      ISTAF =0
      REWIND 12
      REWIND 14
      DO 50 II = 1,NS1
      I = NS1+1-II
      IF(VINC(I).LT.0.0) GO TO 45
      IF((ISTAF.EQ.0).AND.(ICOMP.EQ.0)) WRITE(14,140) XON(I),YON(I),
     1          VINC(I),XMACH(I),PSOPT(I)
      IF((ISTAF.EQ.0).AND.(ICOMP.EQ.1)) WRITE(14,140) XON(I),YON(I),
     1          VCOM(I),CMACH(I),PSOPTC(I)
      IF(ISTAF.EQ.0) ISTAG=ISTAG+1
      GO TO 50
   45 IF((VINC(I)*VINC(I+1).LT.0.0).AND.(ICOMP.EQ.0).AND.(ISTAF.EQ.0))
     1          WRITE(14,140) XON(I),YON(I),VINC(I),XMACH(I),PSOPT(I)
      IF((VINC(I)*VINC(I+1).LT.0.0).AND.(ICOMP.EQ.1).AND.(ISTAF.EQ.0))
     1          WRITE(14,140) XON(I),YON(I),VCOM(I),CMACH(I),PSOPTC(I)
      ISTAF=1
   50 CONTINUE
      REWIND 14
```

```
      IF(ISTAF.EQ.1) ISTAG = ISTAG+1
      WRITE(12,150)  ISTAG
      DO 55 I=1,ISTAG
      READ(14,140) DIMOLP
      WRITE(12,140) DIMOLM
   55 CONTINUE
      ISTAG =0
      ISTAF =0
      NSTA = NH+1
      REWIND 12
      REWIND 13
      REWIND 14
      DO 60 I=NSTA,NT
      IF(VINC(I).GT.0.0) GO TO 56
      IF((ISTAF.EQ.0).AND.(ICOMP.EQ.0)) WRITE(14,140) XON(I),YON(I),
     1          VINC(I),XMACH(I),PSOPT(I)
      IF((ISTAF.EQ.0).AND.(ICOMP.EQ.1)) WRITE(14,140) XON(I),YON(I),
     1          VCOM(I),CMACH(I),PSOPTC(I)
      IF(ISTAF.EQ.0) ISTAG=ISTAG+1
      GO TO 60
   56 IF((VINC(I)*VINC(I-1).LT.0.0).AND.(ISTAF.EQ.0).AND.(ICOMP.EQ.0))
     1          WRITE(14,140) XON(I),YON(I),VINC(I),XMACH(I),PSOPT(I)
      IF((VINC(I)*VINC(I-1).LT.0.0).AND.(ISTAF.EQ.0).AND.(ICOMP.EQ.1))
     1          WRITE(14,140) XON(I),YON(I),VCOM(I),CMACH(I),PSOPTC(I)
      ISTAF=1
   60 CONTINUE
      REWIND 14
      IF(ISTAF.EQ.1) ISTAG=ISTAG+1
      WRITE(13,150) ISTAG
      DO 65 I=1,ISTAG
      READ(14,140) DIMOLP
      WRITE(13,140) DIMOLM
   65 CONTINUE
      REWIND 13
      RETURN
C *****FORMATS*****
   95 FORMAT(1H1)
  100 FORMAT(1H1,10X,'BODY 1',/,10X,'ON-BODY POINTS')
  105 FORMAT(/,2X,'I',7X,'X',11X,'Y',11X,'S',11X,' V  ',8X,'MACH',8X,
     1       'CP',6X,'PS/PT',/)
  110 FORMAT(I4,1P6E12.3,0PF8.4)
  120 FORMAT(1H1,10X,'BODY 2',/,10X,'ON-BODY POINTS')
  130 FORMAT(1H1,10X,'BODY 3',/,10X,'ON-BODY POINTS')
  131 FORMAT(1H1,10X,'BODY 2',/,10X,'ON-BODY POINTS')
  140 FORMAT(0P5F10.4)
  150 FORMAT(I3)
  160 FORMAT(/,2X,'ON-BODY SUPERSONIC VELOCITY CORRECTION START, '
     1       'I = ',I4,/)
  170 FORMAT(2X,'SUPERSONIC VELOCITY CORRECTION STOP, I = ',I4,/)
      END
```

```
      SUBROUTINE OFBOY
      COMMON /COORD/ XON(700),YON(700),XOFF(200),YOFF(20C),S(700),
     1               S1(700),XTEST,XTEST1,XTEST2,YCL,YCU,YCL1,YCL2,YCU1,
     2               YCL2,XR1,XR2,XRH,YR1,YR2,YRH,CUTOF1,CLTOF2,CUTOFH,
     3               ELNE,ANG(700),AR(700),AROFF(200)
      COMMON /COUT/ NT,NSI,NH,NP,IW,NX,KND,ICOMP,K,NXHI1,NXHI2,NXHI3,
     1               NST2,NST3,NST7,NPPR(30),IRAK(30),M1,M2,ICOMP1,IPL,
     2               IHUP
      COMMON /CONDIT/ TTOTAL,PT,PSTAT,TSTAT,PSTATC,ATOTAL,PTC,RHOST,
     1               RHOTOT,ASTAT,OCINF,RSORTC
      COMMON /SOLUT/ VBAR(700),VBARO(2C0),VINC(700),VXINC(200),
     1               VYINC(200),RHOB(7CC),RBORT(700),RHOBC(200),
     2               VCOM(700),RBOOT(2C0),VRE(2C0),VRECOM(200),
     3               VXCOM(200),VYCOM(200),THETA(200),PSCPTC(700),
     4               PSCPT(700),CMACH(700),XMACH(700),CPI(700),CPC(700),
     5               RHOI(700)
      COMMON /SOLUTO/ PSCFPC(200),PSOFP(20C),CMACO(200),XMACO(200),
     1                RHCOI(2C0)
      COMMON/PICT/VPERIN,XX,XMIN,EXEP,YY,YMIN,ORD,EMSTOR,AL,AAAA
C -------------------------------------------------------------------
C
C  THIS SUBROUTINE CALCLLATES OFF-BODY PROPERTIES
C
      DIMENSION WTOT(30),YINT(200),RV(2C0),WFRAC(200)
      WRITE(6,1)
      DO 10 I =1,NP
      VCONC  = 0.2*(VRECOM(I)/ATOTAL)**2
      VCON   = 0.2*(VRE(I)/ATOTAL)**2
      IF(VCONC.GT.1.0) PSOFPC(I)  = 0.C
      IF(VCONC.GT.1.0) CMACO(I) = 999.C
      IF(VCON.GT.1.0) XMACO(I) = 999.0
      IF(VCON.GT.1.0) PSCFP(I) = 0.0
      IF(VCON.GT.1.0) RHCOI(I) = 2000.C
      IF((VCONC.GT.1.0).AND.(VCON.GT.1.C)) GO TO 10
      IF(VCONC.GT.1.0) GC TO 5
      PSOFPC(I) = (1.0-VCONC)**3.5
      CMACO(I) = VRECOM(I)/ATOTAL/SQRT(1.0-VCONC)
      IF(VCON.GT.1.0) GC TO 1C
    5 PSOFP(I)  = (1.0-C.5*RHOTOT*VRE(I)**2/PT)
      XMACO(I) = VRE(I)/ATOTAL/SQRT(1.C-VCON )
   10 CONTINUE
      DO 50 I=1,K
      IF(I.EQ.1) NPH = NPPR(1)
      IF(I.EQ.1) NPL = 1
      IF(I.EQ.1) J3 = NPH
      II = NPPR(I-1)+1
   20 DO 30 J=2,K
      IF(I.NE.J) GO TO 3C
      NPL = II
      NPH = NPPR(J)-NPPR(J-1)
      J3 = NPPR(J)
   30 CONTINUE
C
C  IRAK = 1   RAKE IS UPSTREAM OF BODY
C       = 2   RAKE IS ON LOWER EXTENDED LIP
C       = 3   RAKE IS COMPLETELY IN INLET UPSTREAM OF HUB
```

```
C        = 4   RAKE IS DOWNSTREAM OF AND ABOVE HUB
C        = 5   RAKE IS DOWNSTREAM OF AND BELOW HUB
C        = 6   RAKE IS OUTSIDE AND BELOW INLET
C        = 7   RAKE IS OUTSIDE AND ABOVE INLET
      THEAL = 0.0
      THEAH = 0.0
      IF(IRAK(I).EQ.1) GO TO 40
      IF(IRAK(I).EQ.2) GO TO 60
      IF(IRAK(I).EQ.3) GO TO 70
      IF(IRAK(I).EQ.4) GO TO 80
      IF(IRAK(I).EQ.5) GO TO 90
      IF(IRAK(I).EQ.6) GO TO 220
      IF(IRAK(I).EQ.7) GO TO 230
   40 WRITE(6,500) I
      WRITE(6,521)
      DO 45 L = NPL,J3
      LL = L-NPL+1
      YINT(LL) =YOFF(L)
      IF(ICOMP.EQ.0) RV(LL) = VXINC(L)*RHOTOT
      RHO=(PSOFPC(L)**(1./1.4))
      IF(ICOMP.EQ.1) RV(LL) = VXCOM(L)*RHO*RHOTOT
   45 CONTINUE
      CALL INTEG(RV,YINT,WTOT(I),LL)
      L1 = NPL+1
      DO 46 L = L1,J3
      LL = L-NPL+1
      CALL INTEG(RV,YINT,W,LL)
      WFRAC(LL) = W/WTOT(I)
   46 CONTINUE
      WFRAC(1) = 0.0
      DO 100 N = NPL,J3
      NW = N-NPL+1
      IF(ICOMP.EQ.0) WRITE(6,550) NW,XOFF(N),YOFF(N),VXINC(N),VYINC(N),
     1           VRE(N),THETA(N),XMACO(N),PSOFP(N),WFRAC(NW)
      IF(ICOMP.EQ.1) WRITE(6,550) NW,XOFF(N),YOFF(N),VXCOM(N),VYCOM(N),
     1 VRECOM(N),THETA(N),CMACO(N),PSOFPC(N),WFRAC(NW)
  100 CONTINUE
      GO TO 50
   60 WRITE(6,500) I
      WRITE(6,521)
      DO 110 I10 = NXHI1,NS1
      IF((XOFF(NPL).GE.XON(I10)).AND.(XOFF(NPL).LT.XON(I10+1))) THEAL =
     1           ANG(I10)
  110 CONTINUE
      CALL INTER3(XON,NXHI1,NS1,XOFF(NPL),YINT(1),VLC,VLI,ROC,VRL,
     1              CML,XML,PL,PIL)
      VLIX = VLI*COS(THEAL)
      VLIY = VLI*SIN(THEAL)
      VLCX = VLC*COS(THEAL)
      VLCY = VLC*SIN(THEAL)
      THEAL = THEAL*180.0/3.141592654
      IF(ICOMP.EQ.0) RV(1) = VLIX*RHOTOT
      IF(ICOMP.EQ.1) RV(1) = VLCX*PL**(1./1.4)*RHOTOT
      DO 120 L = NPL,J3
      NW = L-NPL+2
      YINT(NW) = YOFF(L)
      IF(ICOMP.EQ.0) RV(NW) = VXINC(L)*RHOTOT
```

```
      RHO=(PSOFPC(L)**(1./1.4))
      IF(ICOMP.EQ.1) RV(NW) = RHO*VXCOM(L)*RHOTOT
  120 CONTINUE
  121 X3 = 0.0
      CALL INTEG(RV,YINT,WTOT(I),NW)
      IF(ICOMP.EQ.0) WRITE(6,530) XOFF(NPL),YINT(1),VLIX,VLIY,VLI,THEAL,
     1          XML,PIL,X3
      IF(ICOMP.EQ.1) WRITE(6,530) XOFF(NPL),YINT(1),VLCX,VLCY,VLC,THEAL,
     1          CML,PL,X3
      DO 130 L=NPL,J3
      LL = L-NPL+2
      CALL INTEG(RV,YINT,W,LL)
      L1 = L-NPL+1
      WFRAC(L1) = W/WTOT(I)
      IF(ICOMP.EQ.0) WRITE(6,550) L1,XOFF(L),YOFF(L),VXINC(L),VYINC(L),
     1        VRE(L),THETA(L),XMACO(L),PSOFP(L),WFRAC(L1)
      IF(ICOMP.EQ.1) WRITE(6,550) L1,XOFF(L),YOFF(L),VXCOM(L),VYCOM(L),
     1     VRECOM(L),THETA(L),CMACO(L),PSOFPC(L),
     2     WFRAC(L1)
  130 CONTINUE
      GO TO 50
   70 DO 140 I10 = NXHI1,NS1
      IF((XOFF(NPL).GE.XON(I10)).AND.(XOFF(NPL).LT.XON(I10+1))) THEAL =
     1         ANG(I10)
  140 CONTINUE
      NNS = NH+1
      DO 150 I10 = NNS,NXHI3
      IF((XOFF(NPL).LE.XON(I10)).AND.(XOFF(NPL).GT.XON(I10+1))) THEAH =
     1         ANG(I10)
  150 CONTINUE
      NH1 = NPH+2
      CALL INTER3(XON,NXHI1,NS1,XOFF(NPL),YINT(1),VLC,VLI,ROC,VBL,
     1              CML,XPL,PL,PIL)
      CALL INTER3(XON,NNS,NXHI3,XOFF(NPL),YINT(NH1),VUC,VUI,ROCU,
     1              VBU,CML,XMUI,PU,PIU)
  160 VUC = -VUC
      VUI = -VUI
      VLIX = VLI*COS(THEAL)
      VLIY = VLI*SIN(THEAL)
      VLCX = VLC*COS(THEAL)
      VLCY = VLC*SIN(THEAL)
      VUIX = VUI*COS(THEAH)
      VUIY = VUI*SIN(THEAH)
      VUCX = VUC*COS(THEAH)
      VUCY = VUC*SIN(THEAH)
      THEAL = THEAL*180.0/3.141592654
      THEAH = THEAH*180.0/3.141592654
      WRITE(6,500) I
      WRITE(6,521)
      IF(ICOMP.EQ.1)RV(1) = VLCX*PL**(1./1.4)*RHOTOT
      IF(ICOMP.EQ.0)RV(1) = VLIX*RHOTOT
      IF(ICOMP.EQ.0)RV(NH1) = VUIX*RHOTOT
      IF(ICOMP.EQ.1)RV(NH1) = VUCX*PU**(1./1.4)*RHOTOT
      DO 165 L = NPL,J3
      LL = L-NPL+2
      IF(ICOMP.EQ.0) RV(LL) = VXINC(L)*RHOTOT
      YINT(LL) =YOFF(L)
```

```
      RHO=(PSOFPC(L)**(1./1.4))
      IF(ICOMP.EQ.1) RV(LL) = VXCOM(L)*RHO*RHOTOT
  165 CONTINUE
      NW=LL
      IRT=NPPR(I)
      IF((IRAK(I).EQ.4).AND.(XOFF(IRT).LT.XON(NXHI3))) GC TO 121
      X3=0.0
      CALL INTEG(RV,YINT,WTOT(I),NH1)
      IF(ICOMP.EQ.0) WRITE(6,530) XOFF(NPL),YINT(1),VLIX,VLIY,VLI,THEAL,
     1                      XML,PIL,X3
      IF(ICOMP.EQ.1) WRITE(6,530) XOFF(NPL),YINT(1),VLCX,VLCY,VLC,THEAL,
     1                      CML,PL,X3
      DO 170 L = NPL,J3
      LL = L-NPL+2
      CALL INTEG(RV,YINT,W,LL)
      L1 = L-NPL+1
      WFRAC(L1) = W/WTOT(I)
      IF(ICOMP.EQ.0) WRITE(6,550) L1,XCFF(L),YOFF(L),VXIAC(L),VYINC(L),
     1     VRE(L),THETA(L),XMACO(L),PSOFP(L),WFRAC(L1)
      IF(ICOMP.EQ.1) WRITE(6,550) L1,XCFF(L),YOFF(L),VXCCM(L),VYCOM(L),
     1     VRECOM(L),THETA(L),CMACO(L),PSOFPC(L),
     2     WFRAC(L1)
  170 CONTINUE
      L1=LL+1
      WFRAC(L1) =1.0
      IF(ICOMP.EQ.0) WRITE(6,530) XOFF(NPL),YINT(NH1),VUIX,VUIY,VUI,
     1                       THEAH,XMUI,PIU,WFRAC(L1)
      IF(ICOMP.EQ.1) WRITE(6,530) XOFF(NPL),YINT(NH1),VUCX,VUCY,VUC,
     1                       THEAH,CMU,PU,WFRAC(L1)
      GO TO 50
   80 NT1 = NH
      DO 180 I10 = NXHI2,NT1
      IF((XOFF(NPL).GE.XCN(I10)).AND.(XOFF(NPL).LT.XON(I1C+1))) THEAL =
     1          ANG(I10)
  180 CONTINUE
      NNN = NH+1
      DO 190 I10 = NNN,NXHI3
      IF((XOFF(NPL).LE.XCN(I10)).AND.(XOFF(NPL).GT.XON(I1C+1))) THEAH =
     1          ANG(I10)
  190 CONTINUE
      NH1 = NPH+2
      CALL INTER3(XON,NXHI2,NT1,XOFF(NFL),YINT(1),VLC,VLI,ROC,VPL,
     1              CML,XML,PL,PIL)
      CALL INTER3(XON,NNN,NXHI3,XOFF(NFL),YINT(NH1),VUC,VUI,ROCU,
     1              VBU,CML,XMUI,PU,PIU)
      GO TO 160
   90 DO 200 I10 =NXHI1,NS1
      IF((XOFF(NPL).GE.XCN(I1C)).AND.(XOFF(NPL).LT.XON(I1C+1))) THEAL =
     1          ANG(I10)
  200 CONTINUE
      NNN = NS1+1
      DO 210 I10 = NNN,NXHI2
      IF((XOFF(NPL).LE.XCN(I10)).AND.(XOFF(NPL).GT.XON(I1C+1))) THEAH =
     1          ANG(I10)
  210 CONTINUE
      NH1 = NPH+2
      CALL INTER3(XON,NXHI1,NS1,XOFF(NFL),YINT(1),VLC,VLI,ROC,VPL,
```

224

```
     1               CML,XML,PL,PIL)
      CALL INTER3(XON,NNN,NXHI2,XOFF(NPL),YINT(NH1),VUC,VUI,ROCU,
     1               VBU,CML,XMUI,PU,PIU)
      GO TO 160
  220 DO 221 I10 = 1,NXHI1
      IF((XOFF(NPL).LE.XON(I10)).AND.(XOFF(NPL).GT.XON(I10+1))) THEAH =
     1           ANG(I10)
  221 CONTINUE
      WRITE(6,500) I
      WRITE(6,521)
      NH1 = NPH+1
      CALL INTER3(XON,1,  NXHI1,XOFF(NPL),YINT(NH1),VUC,VUI,ROCU,
     1               VBU,CML,XMUI,PU,PIU)
      IF(ICOMP.EQ.0) RV(NH1) = -VUI*COS(THEAH)*RHOTOT
      IF(ICOMP.EQ.1) RV(NH1) = -VUC*COS(THEAH)*PU**(1./1.4)*RHOTOT
      VUIX = -VUI*COS(THEAH)
      VUIY = -VUI*SIN(THEAH)
      VUCX = -VUC*COS(THEAH)
      VUCY = -VUC*SIN(THEAH)
      VUI = -VUI
      VUC = -VUC
      THEAH = THEAH*180.0/3.141592654
      DO 222 JJ = NPL,J3
      J1 =  JJ-NPL +1
      IF(ICOMP.EQ.0) RV(J1) = VXINC(JJ)*RHOTOT
      RHO=(PSOFPC(JJ)**(1./1.4))
      IF(ICOMP.EQ.1) RV(J1) = RHO*VXCOM(JJ)*RHOTOT
      YINT(J1) = YOFF(JJ)
  222 CONTINUE
      CALL INTEG(RV,YINT,WTOT(I),NH1)
      DO 223 J = NPL,J3
      J1 = J-NPL+2
      CALL INTEG(RV,YINT,W,J1)
      WFRAC(J1) = W/WTOT(I)
      WFRAC(1) = 0.0
      J2 =J1-1
      IF(ICOMP.EQ.0) WRITE(6,550) J2,XOFF(J),YOFF(J),VXINC(J),VYINC(J),
     1        VRE(J),THETA(J),XMACO(J),PSOFP(J),WFRAC(J2)
      IF(ICOMP.EQ.1) WRITE(6,550) J2,XOFF(J),YOFF(J),VXCOM(J),VYCOM(J),
     1VRECOM(J),THETA(J),CMACO(J),PSOFPC(J),WFRAC(J2)
  223 CONTINUE
      J2 =J2+1
      IF(ICOMP.EQ.0) WRITE(6,550) J2,XOFF(NPL),YINT(NH1),VUIX,VUIY,VUI,
     1        THEAH,XMUI,PIU,WFRAC(J2)
      IF(ICOMP.EQ.1) WRITE(6,550) J2,XOFF(NPL),YINT(NH1),VUCX,VUCY,VUC,
     1        THEAH,CMU,PU,WFRAC(J2)
      GO TO 50
  230 WRITE(6,500) I
      WRITE(6,521)
      DO 231 I10 = NXHI3,NT
      IF((XOFF(NPL).GE.XON(I10)).AND.(XOFF(NPL).LT.XON(I10+1))) THEAL =
     1           ANG(I10)
  231 CONTINUE
      NH1 = NPH+1
      CALL INTER3(XON,NXHI3,NT ,XOFF(NPL),YINT(1),VLC,VLI,ROC,VBL,
     1               CML,XML,PL,PIL)
      VLIX = VLI*COS(THEAL)
```

```
      VLIY = VLI*SIN(THEAL)
      VLCX = VLC*COS(THEAL)
      VLCY = VLC*SIN(THEAL)
      THEAL = THEAL*180.0/3.141592654
      IF(ICOMP.EQ.0) RV(1) = VLIX*RHOTOT
      IF(ICOMP.EQ.1) RV(1) = VLCX*PL**(1./1.4)*RHOTOT
      DO 232 J = NPL,J3
      J1 = J-NPL+2
      IF(ICOMP.EQ.0) RV(J1) = VXINC(J)*RHOTOT
      RHO=(PSOFPC(J)**(1./1.4))
      IF(ICOMP.EQ.1) RV(J1) = RHO*VXCOM(J)*RHOTOT
      YINT(J1) = YOFF(J)
  232 CONTINUE
      CALL INTEG(RV,YINT,WTOT(I),NH1)
      WFRAC(1) = 0.0
      IF(ICOMP.EQ.0) WRITE(6,530) XOFF(NPL),YINT(1),VLIX,VLIY,VLI,THEAL,
     1             XML,PIL,WFRAC(1)
      IF(ICOMP.EQ.1) WRITE(6,530) XOFF(NPL),YINT(1),VLCX,VLCY,VLC,THEAL,
     1             CML,PL,WFRAC(1)
      DO 233 J = NPL,J3
      J1 = J-NPL+2
      CALL INTEG(RV,YINT,W,J1)
      WFRAC(J1) = W/WTOT(I)
      J2 = J1-1
      IF(ICOMP.EQ.0) WRITE(6,550) J2,XOFF(J),YOFF(J),VXINC(J),VYINC(J),
     1 VRE(J),THETA(J),XMACO(J),PSOFP(J),WFRAC(J1)
      IF(ICOMP.EQ.1) WRITE(6,550) J2,XOFF(J),YOFF(J),VXCOM(J),VYCOM(J),
     1 VRECOM(J),THETA(J),CMACO(J),PSOFPC(J),WFRAC(J1)
  233 CONTINUE
   50 CONTINUE
C  THE FOLLOWING CALCULATES INTEGRATED RAKE WEIGHT FLOW DATA
      WRITE(6,570)
      DO 240 I=1,K
      WTOT(I) = WTOT(I)*32.174/12.0
      IF(I.EQ.1) NPL = 1
      IF(I.NE.1) NPL = NPPR(I-1) + 1
      I9 = NPPR(I)
      IF(IRAK(I).EQ.1) AREA = YOFF(I9)-YOFF(NPL)
      IF(IRAK(I).EQ.2) CALL INTER(XON,YON,NXHI1,NST3,XOFF(NPL),YL)
      IF(IRAK(I).EQ.2) AREA = YOFF(I9)-YL
      NN=NH+1
      NR=NS1+1
      IF((IRAK(I).EQ.3).OR.(IRAK(I).EQ.4))
     1CALL INTER(XON,YON,NN,NXHI3,XOFF(NPL),YH)
      IF(IRAK(I).EQ.5) CALL INTER(XON,YON,NR,NXHI2,XOFF(NPL),YH)
      IF((IRAK(I).EQ.3).OR.(IRAK(I).EQ.5))
     1CALL INTER(XON,YON,NXHI1,NS1,XOFF(NPL),YL)
      IF(IRAK(I).EQ.4) CALL INTER(XON,YON,NXHI2,NH,XOFF(NPL),YL)
      IF((IRAK(I).EQ.3).OR.(IRAK(I).EQ.4).OR.(IRAK(I).EQ.5)) AREA =
     1             YH-YL
      IF(IRAK(I).EQ.6) CALL INTER(XON,YON,1,    NXHI1,XOFF(NPL),YH)
      IF(IRAK(I).EQ.6) AREA = YH-YOFF(NPL)
      IF(IRAK(I).EQ.7) CALL INTER(XON,YON,NXHI3,NT,   XOFF(NPL),YL)
      IF(IRAK(I).EQ.7) AREA = YOFF(I9)-YL
      WDOTCA = WTOT(I)/AREA*12.0*SQRT(TTOTAL/518.67)/PTC*2116.23
      EM=0.1
  235 EF = 85.3848*EM/(1.0+0.2*EM*EM)**3.0-WDOTCA
```

```
      DFOM = 85.3848/(1.0+0.2*EM*EM)**3.0*(1.0-1.2*EM*EM)/(1.0+0.2*EM*EM)
     1        )
      EM = EM-EF/DFOM
      IF(ABS(EF/DFOM/(EM+EF/DFOM)).GT.0.01) GO TO 235
      WRITE(6,580) I,XOFF(NPL),IRAK(I),WTOT(I),WDOTCA,EM
      IF(XOFF(NPL).EQ.XTEST) EMSTOR=EM
  240 CONTINUE
      RETURN
C *****FORMATS*****
    1 FORMAT(1H1)
  500 FORMAT(////,10X,'RAKE NUMBER ',I2)
  521 FORMAT(/,2X,'I',6X,'X',10X,'Y',9X,'VX',9X,'VY',9X,'VRE',7X,'THETA'
     1       ,6X,'MACH',6X,'PS/PT',4X,'WFRACT',/)
  530 FORMAT(4X,1P7E11.3,0PF8.4,1PE11.3)
  550 FORMAT(I4,1P7E11.3,0PF8.4,1PE11.3)
  570 FORMAT(///,10X,'RAKE WEIGHT FLOW DATA',//,3X,'I',8X,'X',4X,'IRAK',
     1       4X,'WDOT',10X,'WDOTCA',8X,'MACH',/)
  580 FORMAT(I4,5X,0PF8.4,I4,1P3E14.5)
      END
```

```
      SUBROUTINE VBARIT(VBAR,ATOTAL,RHOTOT,RHOBAR)
C
C  THIS SUBROUTINE ITERATIVELY CALCULATES RHOBAR
C
      VCRIT = ATOTAL/SQRT(1.2)
      I=0
      VGUES = VBAR
   10 VGUESA = (VGUES/ATOTAL)**2
      A = 1.0-0.2*VGUESA
      B = A -VGUESA
      VCOMP = (VBAR-A**2.5*VGUES)/(A**1.5*B)+VGUES
      IF(ABS((VCOMP-VGUES)/VCOMP).LT.0.0001) GO TO 15
      I =I+1
      IF(VCOMP.GE.VCRIT) VCOMP = 0.5*(VGUES + VCRIT)
      VGUES = VCOMP
      IF(I.GT.20) GO TO 15
      GO TO 10
   15 RHOBAR = (1.0-0.2*(VCOMP/ATOTAL)**2)**2.5*RHOTOT
      IF(I.GT.20) WRITE(6,20) VBAR,VCOMP,RHOBAR
      IF(I.GT.20) VBAR = VCOMP*RHOBAR/RHOTOT
      RETURN
C *****FORMATS*****
   20 FORMAT(1H0,'I EXCEEDS 20 ITERATIONS FOR RHOBAR',5X,'VBAR = '
     1       1PE10.3,2X,'VCOMP = ',1PE10.3,2X,'RHOBAR = ',1PE10.3,/,
     2       ' VBAR HAS BEEN REDUCED TO VCOMP*RHOBAR/RHOTOT, WHERE '
     3       'VCOMP = VCRITICAL')
      END
```

```
      SUBROUTINE INTER(A,B,N1,N2,C,D)
C
C  THIS SUBROUTINE INTERPOLATES ARRAY  A  FOR THE VALUE  C
C  AND RETURNS IN  D  THE CORRESPONDING VALUE FROM  B .
C
      DIMENSION X(700),Y(700),A(1),B(1)
      N = N2-N1+1
      DO 10 I=1,N
      X(I)=A(N1+I-1)
      Y(I)=B(N1+I-1)
   10 CONTINUE
      CALL SORTXY(X,Y,N)
      DO 15 I=1,N
      K=I
      IF(C-X(I)) 25,20,15
   15 CONTINUE
   20 D = Y(K)
      GO TO 30
   25 IF(K.EQ.1) GO TO 35
      IF(K.EQ.N) K=N-1
      IF(X(K).EQ.X(K+1)) K=K-1
      W1 = (C-X(K))*(C-X(K+1))/(X(K-1)-X(K))/(X(K-1)-X(K+1))
      W2 = (C-X(K-1))*(C-X(K+1))/(X(K)-X(K-1))/(X(K)-X(K+1))
      W3 = (C-X(K-1))*(C-X(K))/(X(K+1)-X(K-1))/(X(K+1)-X(K))
      D = Y(K-1)*W1+Y(K)*W2+Y(K+1)*W3
   30 RETURN
   35 D = Y(1)
      RETURN
      END
```



```
      SUBROUTINE SORTXY(X,Y,NPTS)
C
C  THIS SUBROUTINE SORTS  X  INTO ASCENDING ORDER
C
      DIMENSION X(300),Y(300)
      N = NPTS
      NN = N-1
      DO 10 KT = 1,NN
      XMIN = X(KT)
      JAD = KT
      JKL = KT+1
      DO 20 JK = JKL,N
      IF(XMIN-X(JK)) 20,20,25
   25 XMIN = X(JK)
      JAD = JK
   20 CONTINUE
      YMIN = Y(JAD)
      X(JAD) = X(KT)
      Y(JAD) = Y(KT)
      X(KT) = XMIN
      Y(KT) = YMIN
   10 CONTINUE
      RETURN
      END
```

```
     2                YCU2,XR1,XR2,XRH,YR1,YR2,YRH,CUTOF1,CUTOF2,CUTOFH,
     3                ELNC,ANG(700),AR(700),AROFF(200)
      COMMON /VELOC/ V1(700),V2(700),V3(700),V4(700),V5(700),V1X(200),
     1                V2X(200),V3X(200),V4X(200),V5X(200),V1Y(200),
     2                V2Y(200),V3Y(200),V4Y(200),V5Y(200)
      COMMON /COUT/ NT,NS1,NH,NP,IW,NX,KND,ICOMP,K,NXHI1,NXHI2,NXHI3,
     1                NS12,NST3,NST7,NPFP(30),IRAK(30),M1,M2,ICOMP1,IPL,
     2                IHLE
C -----------------------------------------------------------------
      CALL INTER(XON,YON,I1,I2,A,B)
      CALL INTER(XON,V1,I1,I2,A,C)
      CALL INTER(XON,V2,I1,I2,A,D)
      CALL INTER(XON,V3,I1,I2,A,E)
      CALL INTER(XON,V4,I1,I2,A,F)
      IF(IHUB.NE.0) CALL INTER(XON,V5,I1,I2,A,G)
      IF(IHUB.EQ.0) G=0.0
      RETURN
      END
```

```
      SUBROUTINE INTER3(A,I1,I2,C,D,E,F,H,P,Q,R,S2,T)
      COMMON /COORD/ XON(700),YON(700),XOFF(200),YOFF(200),S(700),
     1               S1(700),XTEST,XTEST1,XTEST2,YCL,YCU,YCL1,YCL2,YCU1,
     2               YCU2,XR1,XR2,XRH,YR1,YR2,YRH,CUTOF1,CUTOF2,CUTOFH,
     3               ELNC,ANG(700),AR(700),AROFF(200)
      COMMON /SOLUT/ VBAR(700),VBARO(200),VINC(700),VXINC(200),
     1               VYINC(200),RHOB(700),RBORT(700),RHOBO(200),
     2               VCOM(700),RBOOT(200),VRE(200),VRECOM(200),
     3               VXCOM(200),VYCOM(200),THETA(200),PSOPTC(700),
     4               PSOPT(700),CMACH(700),XMACH(700),CPI(700),CPC(700),
     5               RHOI(700)
C -----------------------------------------------------------------
      DIMENSION A(700)
      CALL INTER(A,YON,  I1,I2,C,D)
      CALL INTER(A,VCOM, I1,I2,C,E)
      CALL INTER(A,VINC, I1,I2,C,F)
      CALL INTER(A,RBORT,I1,I2,C,H)
      CALL INTER(A,VBAR, I1,I2,C,P)
      CALL INTER(A,CMACH,I1,I2,C,Q)
      CALL INTER(A,XMACH,I1,I2,C,R)
      CALL INTER(A,PSOPTC,I1,I2,C,S2)
      CALL INTER(A,PSOPT, I1,I2,C,T)
      RETURN
      END
```

```
      SUBROUTINE INTEG(B,A,C,I1)
C
C  THIS SUBROUTINE IS A TRAPEZOIDAL INTEGRATION ROUTINE
C
      DIMENSION A(700),B(700)
      SUM = 0.0
      DO 10 I=2,I1
      SUM =SUM+0.5*(B(I)+B(I-1))*(A(I)-A(I-1))
   10 CONTINUE
      C = SUM
      RETURN
      END
```

```
      SUBROUTINF CALTIT
      COMMON/VELOC/ V1(7C0),V2(700),V3(700),V4(700),V5(7CC),V1X(200),
     1              V2X(200),V3X(200),V4X(2C0),V5X(200),V1Y(200),
     2              V2Y(200),V3Y(200),V4Y(200),V5Y(200)
      COMMON/PICT/VPERIN,XX,XMIN,EXEP,YY,YMIN,ORD,EMSTOR,AL,AAAA
      COMMON /COORD/ XON(700),YON(700),XOFF(200),YOFF(20C),S(700),
     1              S1(700),XTEST,XTEST1,XTEST2,YCL,YCU,YCL1,YCL2,YCU1,
     2              YCU2,XR1,XR2,XRH,YR1,YR2,YRH,CUTOF1,CUTOF2,CUTOFH,
     3              ELNC,ANG(700),AR(700),AROFF(200)
      COMMON /COUT/ NT,NS1,NH,NP,IW,NX,KND,ICOMP,K,NXHI1,NXHI2,NXHI3,
     1              NST2,NST3,NST7,NPPR(30),IRAN(30),M1,M2,ICOMP1,IPL,
     2              IHLB
      COMMON/SOLUT/ VBAR(700),VBARO(20C),VINC(7C0),VXINC(2C0),
     1              VYINC(200),RHOB(70C),RBORT(700),RHOBC(200),
     2              VCOM(700),RROOT(20C),VRE(200),VRECOM(2C0),
     3              VXCOM(200),VYCOM(2C0),THETA(200),PSOFTC(700),
     4              PSOPT(700),CMACH(7C0),XMACH(700),CPI(7C0),CPC(700),
     5              RHOI(700)
C
C -----------------------------------------------------------------
C  THIS SUBROUTINE TITLES THE PLOTS OF PS/PT AND MACH NUMBER VS S
C
      DIMENSION BD(2),BC1(1),BD2(2),BT2(1),BTIT(1),X(30),Y(30)
      DATA AA,AB,AC,AD,AE/'ANGLE=','VINF =','MACH =','AT X =',
     1                    'SCALE='/
      DATA BD/'OUTER ','WALL  '/
      DATA BD1/' HUB  '/
      DATA BD2/'INNER ','WALL  '/
      DATA BT2/'SREF ='/
      DATA BTIT/'S/SREF'/
      IF(IPL.EQ.10) GO TC 60
      TH = 0.0
      NS = 12
      HE = 0.25
      DO 10 I = 1,19
      X(I) = 0.0
   10 CONTINUE
      Y(1) = 1.0
      DO 20 I = 1,19,4
      X(I) = 10.0
      X(I+1) = 10.0
   20 CONTINUE
      K = 1
      DO 30 I=2,19,2
      K=K+1
      Y(I) = 1.0*K
      Y(I+1) = 1.0*K
   30 CONTINUE
      CALL PLOT(0.C,1.0,3)
      DO 40 I= 1,19,2
      CALL PLOT(X(I),Y(I),2)
      CALL PLOT(X(I+1),Y(I+1),3)
   40 CONTINUE
      CALL PLOT(0.C,10.C,2)
      CALL PLOT(1.C,0.0,3)
      DO 50 I=1,19,2
```

```
      CALL PLOT(Y(I),X(I),2)
      CALL PLOT(Y(I+1),X(I+1),3)
   50 CONTINUE
      CALL PLOT(10.0,0.0,2)
      CALL PLOT(0.0,0.0,3)
      A = 7.1
      B = 1.3
      IF(IPL.EQ.1) CALL SYMBOL(A,B,HE,BD2,TH,12)
      IF(IPL.EQ.2) CALL SYMBOL(A,B,HE,BD1,TH,6)
      IF(IPL.EQ.3) CALL SYMBOL(A,B,HE,BD, TH,12)
   55 A=7.1
      B=0.6
      HE=0.25
      CALL SYMBOL(A,B,HE,BT2,TH,6)
      A=8.4
      IF(IPL.EQ.1) CALL NUMBER(A,B,HE,CUTOF1,TH,3)
      IF(IPL.EQ.2) CALL NUMBER(A,B,HE,CUTOFH,TH,3)
      IF(IPL.EQ.3) CALL NUMBER(A,B,HE,CUTOF2,TH,3)
      A=4.65
      B=0.5
      HE=0.15
      CALL SYMBOL(A,B,HE,BTIT,TH,6)
      GO TO 100
   60 XST=0.0
      YST=0.0
      YST1=YST
      CALL PLOT(XST,YST,3)
      IX=INT(XX)
      IY=INT(YY)
      CALL PLOT(XX,YST,2)
      DO 70 I=1,IY,2
      YST=YST+1.0
      CALL PLOT(XX,YST,3)
      CALL PLOT(0.0,YST,2)
      YST=YST+1.0
      IF(YST.GT.YY) GO TO 70
      CALL PLOT(0.0,YST,3)
      CALL PLOT(XX,YST,2)
   70 CONTINUE
      CALL PLOT(0.0,YST1,3)
      CALL PLOT(0.0,YY,2)
      DO 75 I=1,IX,2
      XST=XST+1.0
      CALL PLOT(XST,YY,3)
      CALL PLOT(XST,0.0,2)
      XST=XST+1.0
      IF(XST.GT.XX) GO TO 75
      CALL PLOT(XST,0.0,3)
      CALL PLOT(XST,YY,2)
   75 CONTINUE
      CALL PLOT(0.0,YST1,3)
      PIO=3.141592654/180.0
      DO 80 I=1,NP
      IF(ICOMP.EQ.0) GO TO 76
      VX=VXCOM(I)
      VY=VYCOM(I)
      VRES=VRECOM(I)
```

```
      GO TO 77
   76 VX=VXINC(I)
      VY=VYINC(I)
      VRES=VRE(I)
   77 SIZE=VRES/VPERIN
      IF(VX.EQ.0) VX=.OCC0000001
      ANGLE=-SIGN(90.,VX)+ATAN(VY/VX)/FIO
      XP=(XOFF(I)-XMIN)/EXEP
      YP=(YOFF(I)-YMIN)/CPD
      IF(XP.GT.XX.OR.XP.LT.0.0.OR.YP.GT.YY.OR.YP.LT.0.0) EQ TO 80
      SIZ=AMIN1(4./21.*SIZE,.15)
      SIZ2=SIZE-0.5*SIZ
      XHEAD=XP+VX/VPERIN/SIZE*SIZ2
      YHEAD=YP+VY/VPERIN/SIZE*SIZ2
      CALL SYMBOL(XP,YP,SIZ2,16,ANGLE,-1)
      CALL SYMBOL(XHEAD,YHEAD,SIZ,2,ANELE,-1)
   80 CONTINUE
      CALL SYMBOL(XX+.15,4.,0.2,AA,0.,6)
      CALL NUMBER(XX+1.25,4.0,0.2,AL,0.,3)
      CALL SYMBOL(XX+.15,3.5,0.2,AB,0.,6)
      CALL NUMBER(XX+1.25,3.5,0.2,AAAA,C.,3)
      CALL SYMBOL(XX+.15,3.,0.2,AC,0.,6)
      CALL NUMBER(XX+1.25,3.0,0.2,EMSTCP,0.,3)
      CALL SYMBOL(XX+.15,2.7,0.2,AD,0.,6)
      CALL NUMBER(XX+1.25,2.7,0.2,XTEST,0.,3)
      CALL SYMBOL(XX+.15,2.2,0.2,AE,0.,6)
      CALL NUMBER(XX+1.25,2.2,0.2,VPERIN,0.,3)
  100 RETURN
      END
```

```
      SUBROUTINE PLTER(N1,N11,NHI,IS,K1)
      COMMON /CLPLOT/ XPEN,YPEN,NX6,NY,IPEN,XLABEL(10),YLABEL(10)
      COMMON /COORD/ XON(700),YCN(7C0),XOFF(200),YOFF(20C),S(700),
     1               S1(700),XTFST,XTEST1,XTEST2,YCL,YCU,YCL1,YCL2,YCU1,
     2               YCU2,XR1,XR2,XRH,YR1,YR2,YRH,CUTOF1,CLTOF2,CUTOFH,
     3               ELNC,ANG(700),AR(700),AROFF(200)
      COMMON /CONT/ VC,VS1,VS2,XMC,XMC1,XMC2,WDOTC,WDOTC1,WDOTC2,
     1               TITLE(3),VINF,ALFA,A,B,C,D,A1C,A2C,A3C,A4C,A5C,A11,
     2               A21,A31,A41,A51,A12,A22,A32,A42,A52,V1C,VIC1,VIC2
      COMMON /COUT/ NT,NS1,NH,NP,IW,NX,KND,ICOMP,K,NXHI1,NXHI2,NXHI3,
     1               NST2,NST3,NST7,NPFR(30),IRAK(30),M1,M2,ICOMP1,IPL,
     2              IHUB
      COMMON /SOLUT/ VBAR(700),VBARO(2C0),VINC(700),VXINC(200),
     1               VYINC(200),RHOB(7C0),RBORT(700),RHOBC(200),
     2               VCOM(700),RBOOT(2C0),VRE(200),VRECOM(200),
     3               VXCCM(200),VYCOM(200),THETA(200),PSCPTC(700),
     4               PSOPT(700),CMACH(7C0),XMACH(700),CP1(700),CPC(700),
     5               RHCI(700)
C ----------------------------------------------------------------------
C
C  THIS SUBROUTINE PLOTS PS/PT AND MACH NUMBER VS S
C
      DIMENSION YD(4),YDC(3),XPLOT(500),YPLOT(500),KKK(7),P(14)
      KKK(1) = 4
      KKK(2) = 1
      KKK(3) = 2
      KKK(4) = 1
      KKK(5) = 1
      KKK(6) = N1
      KKK(7) = N11
      P(1) =3.0
      P(2) =10.0
      P(3) =0.0
      P(4) =1.0
      P(5) =10.0
      P(6) =0.0
      P(7) =1.0
      P(8) =10.0
      P(9) =0.0
      P(10)=0.0
      P(11)=0.0
      P(12)=0.0
      P(13)=0.0
      P(14)=90.C
      DATA YD(1),YD(2),YC(3),YD(4) / 'PRESSU','RE RAT','IO, PS','/PTC  '
     1         /
      DATA YDD(1),YDD(2),YDD(3)/ 'LOCAL ','MACH N','O.    '/
      DO 10 I=1,4
      YLABEL(I) = YD(I)
   10 CONTINUE
      XLABEL(1) = TITLE(1)
      XLABEL(2) = TITLE(2)
      XLABEL(3) = TITLE(3)
      XPEN = 0.0
      YPEN = 0.0
      IPEN =-3
```

```
      NX6=-18
      NY=24
      IF(K1.EQ.2) GO TO 31
      DO 20 I=1,N1
      XPLOT(I) =S1(NHI+I-1)
      IF(ICOMP.EQ.0) YPLOT(I) = PSOPT(NHI+I-1)
      IF(ICOMP.EQ.1) YPLOT(I) = PSOPTC(NHI+I-1)
   20 CONTINUE
      IF(IPL.EQ.1) KKK(3)=1
      IF(IPL.EQ.1) GO TO 35
      DO 30 I=1,N11
      XPLOT(N1+I) = S1(IS+I-1)
      IF(ICOMP.EQ.0) YPLOT(N1+I) = PSOPT(IS+I-1)
      IF(ICOMP.EQ.1) YPLOT(N1+I) = PSOPTC(IS+I-1)
   30 CONTINUE
      GO TO 35
   31 DO 32 I=1,N1
      XPLOT(I) = S1(NHI-I+1)
      IF(ICOMP.EQ.0) YPLOT(I) = PSOPT(NHI-I+1)
      IF(ICOMP.EQ.1) YPLOT(I) = PSOPTC(NHI-I+1)
   32 CONTINUE
   35 CALL CALPLT(XPLOT,YPLOT,KKK,P)
      IF(K1.EQ.2) GO TO 61
      DO 40 I =1,N1
      IF(ICOMP.EQ.0) YPLOT(I) = XMACH(NHI+I-1)
      IF(ICOMP.EQ.1) YPLOT(I) = CMACH(NHI+I-1)
   40 CONTINUE
      IF(IPL.EQ.1) KKK(3)=1
      IF(IPL.EQ.1) GO TO 65
      DO 50 I= 1,N11
      IF(ICOMP.EQ.0) YPLOT(N1+I) = XMACH(IS+I-1)
      IF(ICOMP.EQ.1) YPLOT(N1+I) = CMACH(IS+I-1)
   50 CONTINUE
      GO TO 65
   61 DO 62 I=1,N1
      IF(ICOMP.EQ.0) YPLOT(I)= XMACH(NHI-I+1)
      IF(ICOMP.EQ.1) YPLOT(I)= CMACH(NHI-I+1)
   62 CONTINUE
   65 DO 60 I=1,3
      YLABEL(I)=YDD(I)
   60 CONTINUE
      NY = 18
      P(7) = 2.0
      CALL CALPLT(XPLOT,YPLOT,KKK,P)
      RETURN
      END
```

TEST CASE INPUT AND OUTPUT

Program SCIRCL

```
                    INPUT FILE DUMP

          NOZZLE WITH CENTERBODY
      6.        -2.       4.        6.      -10.       4.
                1331
           20Y
     3.        .4        4.        0.
    8
   -1.    .4      11.6    20
    6.    .2       4.8    10
    6.   7.2      11.8    10
   12.   -5.      -3.      2
   14.   -5.      -3.      2
   16.   -5.      -3.      2
   18.   -5.      -3.      2
   20.   -5.      -3.      2
    1.        4.
   10.
  -45.         0.
    0.          0.
    1.
    0.         6.
    0.         0.
    0.        2.        2.
    0.         6.                             9.        9.
    0.         0.                            -3.       -5.
    1.
    9.          9.
   -3.        -40.
    2.        3.        .4
    1.
   11.          6.
     6.         5.
    0.         2.        2.
   10.          6.                           3.        3.
    5.          5.                           6.        8.
   -4.
    11.         3.
    6.          6.        6.
    3.       4.         .4
    1.
   21.          21.
  -40.          -3.
    0.         2.        2.
   21.         21.                           6.        C.
   -4.          -3.                          12.       12.
    1.
    6.          0.
   12.         12.
    1.
    0.          -45.
    12.         12.
```

```
CASE                          NOZZLE WITH CENTERBODY

GEOMETRY OUTPUT IS SAVED ON UNIT 17 FOR
PROGRAM    24Y WHICH WILL GENERATE
HIGHER ORDER SOLUTION
WITH STRIP VORTEX FLOW.

SCIRCLE WILL PRODUCE
NO CROSSECTIONAL AREA PLOTS.
NO CURVATURE PLOTS.
NO REWORKING OF GEOMETRY.

      NO. OF BODIES = 3.     DELS =   .400     DELSMX =  4.000     XRI =    .000000

 BODY  1 ****************************************************************************************

   ---- STRAIGHT LINE
               ENREED
               10.000     X     -4.5000+01     0.0000
                          Y      0.0000        0.0000

         LAST POINT K=   20, X=   .00000   , Y=   .00000   ,KAPPA=   .00000   ,DY/DX=   .00000   ,ALPHA=   .00000
   ---- STRAIGHT LINE
               ENREED
                1.000     X      0.0000        6.0000+00
                          Y      0.0000        0.0000

         LAST POINT K=   35, X=  .60000+01, Y=   .00000   ,KAPPA=   .00000   ,DY/DX=   .00000   ,ALPHA=   .00000

   ---- SUPERELLIPSE, ENREED=        .00
            EXPONENTS
         P =  2.000     X     0.0000      6.0000+00     0.0000     0.0000     9.0000+00     9.0000+00
         Q =  2.000     Y     0.0000      0.0000        0.0000     0.0000    -3.0000+00    -5.0000+00

         P =      .20000000+01    A =     .30000000+01       XO =    -.60000000+01
         Q =      .20000000+01    B =     .30000000+01       YO =     .30000000+01   OMEGA =     .00000000

            3  ITERATIONS---
          DELS IN =   .40000    DELS =   .38940   DELS OUT =   .38940   DSTEST =   .00079

         LAST POINT K=   48, X=   .90000+01, Y=  -.30000+01,KAPPA=  -.33333+00,DY/DX=   .99999+05,ALPHA=   .89999+02
   ---- STRAIGHT LINE
               ENREED
                1.000     X      9.0000+00     9.0000+00
                          Y     -3.0000+00    -4.0000+01

         LAST POINT K=   66, X=   .90000+01, Y=  -.40000+02,KAPPA=   .00000   ,DY/DX=  -.99999+05,ALPHA=  -.90000+02
 BODY  2 ****************************************************************************************

   ---- STRAIGHT LINE
               ENREED
                1.000     X      1.1000+01     6.0000+00
                          Y      6.0000+00     5.0000+00
```

```
          LAST POINT K=   74, X=   .60000+01, Y=   .50000+01,KAPPA=   .00000   ,DY/DX=   .00000   ,ALPHA=   .00000
   ---- SUPERELLIPSE, ENREED=        .00
          EXPONENTS
          P =  2.000      X      1.0000+01      6.0000+00      0.0000      0.0000      3.0000+00      3.0000+0C
          Q =  2.000      Y      5.0000+00      5.0000+00      0.0000      0.0000      6.0000+00      8.0000+00

          P =     .20000000+01   A =    .30000000+01      XO =    .60000000+01
          Q =     .20000000+01   B =    .10000000+01      YO =    .60000000+01   OMEGA =    .00000000

            6  ITERATIONS---
           DELS IN =   .40000   DELS =   .35705   DELS OUT =   .32757   DSTEST =   .00276

          LAST POINT K=   84, X=   .30000+01, Y=   .60000+01,KAPPA=  -.29997+01,DY/DX=  -.99999+05,ALPHA=  -.89999+02

   ---- MIRROR IMAGE
             ENREED    THIS SEGMENT IS A MIRROR IMAGE ABOUT THE Y=          6.0000+00LINE
                           THE ORIGINAL SEGMENT LIES BETWEEN THE FOLLOWING POINTS
               ******      X      1.1000+01      3.0000+00
                           Y      6.0000+00      6.0000+00

          LAST POINT K=  101, X=   .11000+02, Y=   .60000+01,KAPPA=   .00000   ,DY/DX=  -.20000+00,ALPHA=  -.11310+02

 BODY  3 ******************************************************************************************************

   ---- STRAIGHT LINE
             ENREED
               1.000       X      2.1000+01      2.1000+01
                           Y     -4.0000+01     -3.0000+00

          LAST POINT K=  120, X=   .21000+02, Y=  -.30000+01,KAPPA=   .00008   ,DY/DX=   .00000   ,ALPHA=   .00000
   ---- SUPERELLIPSE, ENREED=        .00
          EXPONENTS
          P =  2.000      X      2.1000+01      2.1000+01      0.0000      0.0000      6.0000+00      0.0000
          Q =  2.000      Y     -4.0000+00     -3.0000+00      0.0000      0.0000      1.2000+01      1.2000+01

          P =     .20000000+01   A =    .15000000+02      XO =    .20000000+02
          Q =     .20000000+01   B =    .15000000+02      YO =    .11000000+02   OMEGA =    .00000000

            3  ITERATIONS---
           DELS IN =   .40000   DELS =   .39792   DELS OUT =   .39792   DSTEST =   .39259

          LAST POINT K=  180, X=   .60000+01, Y=   .12000+02,KAPPA=  -.66667-01,DY/DX=   .00000   ,ALPHA=   .00000

   ---- STRAIGHT LINE
             ENREED
               1.000       X      6.0000+00      0.0000
                           Y      1.2000+01      1.2000+01

          LAST POINT K=  195, X=   .28312-06, Y=   .12000+02,KAPPA=   .00000   ,DY/DX=   .00000   ,ALPHA=   .00000

   ---- STRAIGHT LINE
             ENREED
               1.000       X      0.0000        -4.5000+01
                           Y      1.2000+01      1.2000+01
```

LAST POINT K= 214, X= -.45000+02, Y= .12000+02,KAPPA= .00000 ,DY/DX= .00000 ,ALPHA= .00000

INPUT FOR THE COMBINE PROGRAM NT(1)= 65 NT(2)= 99 NT(3)= 211 NP= 50

BODY 1	CO-ORDINATES- X	Y	KAPPA	DY/DX	ALPHA	S	S-S(2)	DELTAS
1	-.45000+02	.00000	.00000	.00000	.00000	.00000	.45000+02	.00000
2	-.40733+02	.00000	.00000	.00000	.00000	.42669+01	.40733+02	.42669+01
3	-.36466+02	.00000	.00000	.00000	.00000	.85338+01	.36466+02	.42669+01
4	-.32199+02	.00000	.00000	.00000	.00000	.12801+02	.32199+02	.42669+01
5	-.27932+02	.00000	.00000	.00000	.00000	.17068+02	.27932+02	.42669+01
6	-.23666+02	.00000	.00000	.00000	.00000	.21334+02	.23666+02	.42669+01
7	-.19399+02	.00000	.00000	.00000	.00000	.25601+02	.19399+02	.42669+01
8	-.15832+02	.00000	.00000	.00000	.00000	.29168+02	.15832+02	.35664+01
9	-.12860+02	.00000	.00000	.00000	.00000	.32140+02	.12860+02	.29720+01
10	-.10383+02	.00000	.00000	.00000	.00000	.34617+02	.10383+02	.24767+01
11	-.83196+01	.00000	.00000	.00000	.00000	.36680+02	.83196+01	.20639+01
12	-.65996+01	.00000	.00000	.00000	.00000	.38400+02	.65996+01	.17199+01
13	-.51664+01	.00000	.00000	.00000	.00000	.39834+02	.51664+01	.14333+01
14	-.39720+01	.00000	.00000	.00000	.00000	.41028+02	.39720+01	.11944+01
15	-.29766+01	.00000	.00000	.00000	.00000	.42023+02	.29766+01	.99533+00
16	-.21472+01	.00000	.00000	.00000	.00000	.42853+02	.21472+01	.82944+00
17	-.14560+01	.00000	.00000	.00000	.00000	.43544+02	.14560+01	.69120+00
18	-.88000+00	.00000	.00000	.00000	.00000	.44120+02	.88000+00	.57600+00
19	-.40000+00	.00000	.00000	.00000	.00000	.44600+02	.40000+00	.48000+00
20	.00000	.00000	.00000	.00000	.00000	.45000+02	.00000	.40000+00
21	.40000+00	.00000	.00000	.00000	.00000	.45400+02	-.40000+00	.40000+00
22	.80000+00	.00000	.00000	.00000	.00000	.45800+02	-.80000+00	.40000+00
23	.12000+01	.00000	.00000	.00000	.00000	.46200+02	-.12000+01	.40000+00
24	.16000+01	.00000	.00000	.00000	.00000	.46600+02	-.16000+01	.40000+00
25	.20000+01	.00000	.00000	.00000	.00000	.47000+02	-.20000+01	.40000+00
26	.24000+01	.00000	.00000	.00000	.00000	.47400+02	-.24000+01	.40000+00
27	.28000+01	.00000	.00000	.00000	.00000	.47800+02	-.28000+01	.40000+00
28	.32000+01	.00000	.00000	.00000	.00000	.48200+02	-.32000+01	.40000+00
29	.36000+01	.00000	.00000	.00000	.00000	.48600+02	-.36000+01	.40000+00
30	.40000+01	.00000	.00000	.00000	.00000	.49000+02	-.40000+01	.40000+00
31	.44000+01	.00000	.00000	.00000	.00000	.49400+02	-.44000+01	.40000+00
32	.48000+01	.00000	.00000	.00000	.00000	.49800+02	-.48000+01	.40000+00
33	.52000+01	.00000	.00000	.00000	.00000	.50200+02	-.52000+01	.40000+00
34	.56000+01	.00000	.00000	.00000	.00000	.50600+02	-.56000+01	.40000+00
35	.60000+01	-.11905-07	-.33333+00	.00000	.00000	.51000+02	-.60000+01	.40000+00
36	.63659+01	-.22395-01	-.33333+00	-.12288+00	-.70052+01	.51367+02	-.63666+01	.36656+00
37	.67290+01	-.89927-01	-.33333+00	-.25052+00	-.14064+02	.51736+02	-.67359+01	.36937+00
38	.70839+01	-.20266+00	-.33333+00	-.38749+00	-.21181+02	.52108+02	-.71083+01	.37238+00
39	.74161+01	-.35528+00	-.33333+00	-.53546+00	-.28167+02	.52474+02	-.74739+01	.36559+00
40	.77271+01	-.54703+00	-.33333+00	-.70409+00	-.35149+02	.52839+02	-.78392+01	.36534+00
41	.80124+01	-.77489+00	-.33333+00	-.90431+00	-.42123+02	.53204+02	-.82042+01	.36495+00
42	.82671+01	-.10352+01	-.33333+00	-.11538+01	-.49085+02	.53568+02	-.85685+01	.36430+00
43	.84798+01	-.13117+01	-.33333+00	-.14688+01	-.55752+02	.53917+02	-.89174+01	.34888+00
44	.86607+01	-.16141+01	-.33333+00	-.19199+01	-.62486+02	.54270+02	-.92698+01	.35239+00
45	.88060+01	-.19366+01	-.33333+00	-.26437+01	-.69280+02	.54625+02	-.96253+01	.35553+00
46	.89125+01	-.22968+01	-.33333+00	-.40498+01	-.76130+02	.54984+02	-.99837+01	.35842+00
47	.89776+01	-.26360+01	-.33333+00	-.81817+01	-.83032+02	.55345+02	-.10345+02	.36116+00
48	.90000+01	-.30000+01	.00000	-.99999+05	-.90000+02	.55710+02	-.10710+02	.36464+00
49	.90000+01	-.33654+01	.00000	-.99999+05	-.90000+02	.56099+02	-.11099+02	.38940+00
50	.90000+01	-.38567+01	.00000	-.99999+05	-.90000+02	.56566+02	-.11566+02	.46728+00

51	.90000+01	-.44174+01	.00000	-.99999+05	-.90000+02	.57127+02	-.12127+02	.56074+00
52	.90000+01	-.50903+01	.00000	-.99999+05	-.90000+02	.57800+02	-.12800+02	.67289+00
53	.90000+01	-.58978+01	.00000	-.99999+05	-.90000+02	.58607+02	-.13607+02	.80747+00
54	.90000+01	-.68667+01	.00000	-.99999+05	-.90000+02	.59576+02	-.14576+02	.96896+00
55	.90000+01	-.80295+01	.00000	-.99999+05	-.90000+02	.60739+02	-.15739+02	.11628+01
56	.90000+01	-.94248+01	.00000	-.99999+05	-.90000+02	.62134+02	-.17134+02	.13953+01
57	.90000+01	-.11099+02	.00000	-.99999+05	-.90000+02	.63809+02	-.18809+02	.16744+01
58	.90000+01	-.13108+02	.00000	-.99999+05	-.90000+02	.65818+02	-.20818+02	.20092+01
59	.90000+01	-.15519+02	.00000	-.99999+05	-.90000+02	.68229+02	-.23229+02	.24111+01
60	.90000+01	-.18413+02	.00000	-.99999+05	-.90000+02	.71122+02	-.26122+02	.28933+01
61	.90000+01	-.21885+02	.00000	-.99999+05	-.90000+02	.74594+02	-.29594+02	.34720+01
62	.90000+01	-.25508+02	.00000	-.99999+05	-.90000+02	.78217+02	-.33217+02	.36230+01
63	.90000+01	-.29131+02	.00000	-.99999+05	-.90000+02	.81840+02	-.36840+02	.36230+01
64	.90000+01	-.32754+02	.00000	-.99999+05	-.90000+02	.85463+02	-.40463+02	.36230+01
65	.90000+01	-.36377+02	.00000	-.99999+05	-.90000+02	.89086+02	-.44086+02	.36230+01
66	.90000+01	-.40000+02	.00000	-.99999+05	-.90000+02	.92710+02	-.47710+02	.36230+01

BODY 2 CO-ORDINATES-	X	Y	KAPPA	DY/DX	ALPHA	S	S-S(2)	DELTAS
67	.11000+02	.60000+01	.00000	.20000+00	.11310+02	.00000	.84334+01	.00000
68	.98948+01	.57790+01	.00000	.20000+00	.11310+02	.11271+01	.73063+01	.11271+01
69	.89188+01	.55838+01	.00000	.20000+00	.11310+02	.21224+01	.63110+01	.99533+00
70	.81055+01	.54211+01	.00000	.20000+00	.11310+02	.29518+01	.54815+01	.82944+00
71	.74277+01	.52855+01	.00000	.20000+00	.11310+02	.36430+01	.47903+01	.69120+00
72	.68629+01	.51726+01	.00000	.20000+00	.11310+02	.42190+01	.42143+01	.57600+00
73	.63922+01	.50784+01	.00000	.20000+00	.11310+02	.46990+01	.37343+01	.48000+00
74	.60000+01	.50000+01	.11111+00	.00000	.00000	.50990+01	.33343+01	.40000+00
75	.56507+01	.50068+01	.11315+00	-.39080-01	-.22380+01	.54484+01	.29849+01	.34939+00
76	.53018+01	.50275+01	.11965+00	-.79773-01	-.45610+01	.57979+01	.26354+01	.34953+00
77	.49541+01	.50627+01	.13190+00	-.12399+00	-.70679+01	.61474+01	.22860+01	.34943+00
78	.46088+01	.51140+01	.15274+00	-.17447+00	-.98966+01	.64964+01	.19369+01	.34907+00
79	.42674+01	.51836+01	.18329+00	-.23581+00	-.13268+02	.68449+01	.15885+01	.34842+00
80	.39324+01	.52754+01	.25300+00	-.31707+00	-.17592+02	.71923+01	.12411+01	.34740+00
81	.36081+01	.53964+01	.38735+00	-.44032+00	-.23765+02	.75384+01	.89498+00	.34612+00
82	.33158+01	.55534+01	.71734+00	-.66777+00	-.33734+02	.78701+01	.56323+00	.33175+00
83	.30901+01	.57568+01	.16775+01	-.13292+01	-.53045+02	.81740+01	.25939+00	.30384+00
84	.30000+01	.60000+01	.00000	.99999+05	.89999+02	.84334+01	.00000	.25939+00
85	.30901+01	.62432+01	-.16775+01	.13292+01	.53045+02	.86927+01	-.25939+00	.25939+00
86	.33158+01	.64466+01	-.71734+00	.66777+00	.33734+02	.89966+01	-.56323+00	.30384+00
87	.36081+01	.66036+01	-.38735+00	.44032+00	.23765+02	.93283+01	-.89498+00	.33175+00
88	.39324+01	.67246+01	-.25300+00	.31707+00	.17592+02	.96745+01	-.12411+01	.34612+00
89	.42674+01	.68164+01	-.18829+00	.23581+00	.13268+02	.10022+02	-.15885+01	.34740+00
90	.46088+01	.68860+01	-.15274+00	.17447+00	.98966+01	.10370+02	-.19369+01	.34842+00
91	.49541+01	.69373+01	-.13190+00	.12399+00	.70679+01	.10719+02	-.22860+01	.34907+00
92	.53018+01	.69725+01	-.11965+00	.79773-01	.45610+01	.11069+02	-.26354+01	.34943+00
93	.56507+01	.69932+01	-.11315+00	.39080-01	.22380+01	.11418+02	-.29849+01	.34953+00
94	.60000+01	.70000+01	-.11111+00	.00000	.00000	.11768+02	-.33343+01	.34939+00
95	.63922+01	.69216+01	.00000	-.20000+00	-.11310+02	.12168+02	-.37343+01	.40000+00
96	.68629+01	.68274+01	.00000	-.20000+00	-.11310+02	.12648+02	-.42143+01	.48000+00
97	.74277+01	.67145+01	.00000	-.20000+00	-.11310+02	.13224+02	-.47903+01	.57600+00
98	.81055+01	.65789+01	.00000	-.20000+00	-.11310+02	.13915+02	-.54815+01	.69120+00
99	.89188+01	.64162+01	.00000	-.20000+00	-.11310+02	.14744+02	-.63110+01	.82944+00
100	.98948+01	.62210+01	.00000	-.20000+00	-.11310+02	.15740+02	-.73063+01	.99533+00
101	.11000+02	.60000+01	.00000	-.20000+00	-.11310+02	.16867+02	-.84334+01	.11271+01

BODY 3 CO-ORDINATES-	X	Y	KAPPA	DY/DX	ALPHA	S	S-S(2)	DELTAS

102	.21000+02	-.40000+02	.00000	-.99999+05	-.90000+02	.00000	.66561+02	.00000
103	.21000+02	-.36480+02	.00000	-.99999+05	-.90000+02	.35203+01	.63041+02	.35203+01
104	.21000+02	-.32959+02	.00000	-.99999+05	-.90000+02	.70405+01	.59521+02	.35203+01
105	.21000+02	-.29439+02	.00000	-.99999+05	-.90000+02	.10561+02	.56000+02	.35203+01
106	.21000+02	-.25919+02	.00000	-.99999+05	-.90000+02	.14081+02	.52480+02	.35203+01
107	.21000+02	-.22399+02	.00000	-.99999+05	-.90000+02	.17601+02	.48960+02	.35203+01
108	.21000+02	-.18832+02	.00000	-.99999+05	-.90000+02	.21168+02	.45393+02	.35664+01
109	.21000+02	-.15860+02	.00000	-.99999+05	-.90000+02	.24140+02	.42421+02	.29720+01
110	.21000+02	-.13383+02	.00000	-.99999+05	-.90000+02	.26617+02	.39945+02	.24767+01
111	.21000+02	-.11320+02	.00000	-.99999+05	-.90000+02	.28680+02	.37881+02	.20639+01
112	.21000+02	-.95996+01	.00000	-.99999+05	-.90000+02	.30400+02	.36161+02	.17199+01
113	.21000+02	-.81664+01	.00000	-.99999+05	-.90000+02	.31834+02	.34728+02	.14333+01
114	.21000+02	-.69720+01	.00000	-.99999+05	-.90000+02	.33028+02	.33533+02	.11944+01
115	.21000+02	-.59766+01	.00000	-.99999+05	-.90000+02	.34023+02	.32538+02	.99533+00
116	.21000+02	-.51472+01	.00000	-.99999+05	-.90000+02	.34853+02	.31708+02	.82944+00
117	.21000+02	-.44560+01	.00000	-.99999+05	-.90000+02	.35544+02	.31017+02	.69120+00
118	.21000+02	-.38800+01	.00000	-.99999+05	-.90080+02	.36120+02	.30441+02	.57600+00
119	.21000+02	-.34000+01	.00000	-.99999+05	-.90000+02	.36600+02	.29961+02	.48000+00
120	.21000+02	-.30000+01	-.66667-01	-.99900+03	.90000+02	.37000+02	.29561+02	.40000+00
121	.20995+02	-.26073+01	-.66667-01	-.38185+02	.88500+02	.37393+02	.29169+02	.39273+00
122	.20979+02	-.22148+01	-.66667-01	-.19076+02	.86999+02	.37786+02	.28776+02	.39286+00
123	.20954+02	-.18226+01	-.66667-01	-.12701+02	.85498+02	.38179+02	.28383+02	.39299+00
124	.20918+02	-.14311+01	-.66666-01	-.95085+01	.83996+02	.38572+02	.27990+02	.39313+00
125	.20872+02	-.10406+01	-.66666-01	-.75898+01	.82494+02	.38965+02	.27596+02	.39326+00
126	.20815+02	-.65125+00	-.66667-01	-.63076+01	.80991+02	.39358+02	.27203+02	.39342+00
127	.20748+02	-.26340+00	-.66667-01	-.53893+01	.79488+02	.39752+02	.26809+02	.39354+00
128	.20671+02	.12270+00	-.66667-01	-.46983+01	.77984+02	.40146+02	.26416+02	.39368+00
129	.20584+02	.50678+00	-.66667-01	-.41589+01	.76480+02	.40539+02	.26022+02	.39383+00
130	.20487+02	.88859+00	-.66667-01	-.37256+01	.74975+02	.40933+02	.25628+02	.39397+00
131	.20380+02	.12679+01	-.66667-01	-.33694+01	.73470+02	.41328+02	.25234+02	.39411+00
132	.20263+02	.16443+01	-.66667-01	-.30710+01	.71964+02	.41722+02	.24839+02	.39426+00
133	.20136+02	.20177+01	-.66667-01	-.28172+01	.70457+02	.42116+02	.24445+02	.39441+00
134	.19999+02	.23878+01	-.66667-01	-.25983+01	.68950+02	.42511+02	.24050+02	.39457+00
135	.19852+02	.27543+01	-.66667-01	-.24073+01	.67442+02	.42905+02	.23656+02	.39472+00
136	.19696+02	.31169+01	-.66667-01	-.22391+01	.65934+02	.43300+02	.23261+02	.39489+00
137	.19530+02	.34755+01	-.66667-01	-.20895+01	.64425+02	.43695+02	.22866+02	.39505+00
138	.19355+02	.38297+01	-.66667-01	-.19554+01	.62915+02	.44091+02	.22471+02	.39522+00
139	.19170+02	.41793+01	-.66667-01	-.18345+01	.61405+02	.44486+02	.22076+02	.39539+00
140	.18976+02	.45241+01	-.66667-01	-.17246+01	.59894+02	.44882+02	.21680+02	.39557+00
141	.18773+02	.48638+01	-.66667-01	-.16243+01	.58382+02	.45277+02	.21284+02	.39575+00
142	.18561+02	.51982+01	-.66667-01	-.15322+01	.56869+02	.45673+02	.20888+02	.39594+00
143	.18341+02	.55271+01	-.66667-01	-.14472+01	.55356+02	.46069+02	.20492+02	.39614+00
144	.18111+02	.58501+01	-.66667-01	-.13684+01	.53842+02	.46466+02	.20095+02	.39634+00
145	.17873+02	.61672+01	-.66667-01	-.12951+01	.52328+02	.46862+02	.19699+02	.39656+00
146	.17626+02	.64780+01	-.66667-01	-.12266+01	.50812+02	.47259+02	.19302+02	.39678+00
147	.17371+02	.67824+01	-.66667-01	-.11624+01	.49295+02	.47656+02	.18905+02	.39701+00
148	.17108+02	.70801+01	-.66667-01	-.11020+01	.47778+02	.48053+02	.18508+02	.39725+00
149	.16837+02	.73709+01	-.66667-01	-.10450+01	.46260+02	.48451+02	.18110+02	.39751+00
150	.16558+02	.76546+01	-.66667-01	-.99097+00	.44740+02	.48849+02	.17713+02	.39777+00
151	.16279+02	.79240+01	-.66667-01	-.94100+00	.43259+02	.49236+02	.17325+02	.38776+00
152	.15993+02	.81862+01	-.66667-01	-.89338+00	.41777+02	.49624+02	.16937+02	.38800+00
153	.15701+02	.84411+01	-.66667-01	-.84788+00	.40294+02	.50013+02	.16549+02	.38824+00
154	.15401+02	.86884+01	-.66667-01	-.80431+00	.38810+02	.50401+02	.16160+02	.38846+00
155	.15095+02	.89281+01	-.66667-01	-.76249+00	.37325+02	.50790+02	.15771+02	.38867+00
156	.14783+02	.91598+01	-.66667-01	-.72228+00	.35840+02	.51179+02	.15383+02	.38888+00
157	.14464+02	.93836+01	-.66667-01	-.68353+00	.34354+02	.51568+02	.14994+02	.38908+00
158	.14140+02	.95990+01	-.66667-01	-.64611+00	.32867+02	.51957+02	.14604+02	.38927+00

159	.13810+02	.98061+01	-.66667-01	-.60990+00	.31379+02	.52346+02	.14215+02	.38946+00
160	.13475+02	.10005+02	-.66667-01	-.57481+00	.29891+02	.52736+02	.13825+02	.38964+00
161	.13135+02	.10195+02	-.66667-01	-.54074+00	.28402+02	.53126+02	.13435+02	.38981+00
162	.12789+02	.10376+02	-.66667-01	-.50759+00	.26912+02	.53516+02	.13045+02	.38998+00
163	.12439+02	.10548+02	-.66667-01	-.47530+00	.25422+02	.53906+02	.12655+02	.39015+00
164	.12084+02	.10711+02	-.66667-01	-.44378+00	.23931+02	.54296+02	.12265+02	.39031+00
165	.11726+02	.10864+02	-.66667-01	-.41297+00	.22439+02	.54687+02	.11874+02	.39047+00
166	.11363+02	.11009+02	-.66667-01	-.38281+00	.20947+02	.55077+02	.11484+02	.39062+00
167	.10996+02	.11144+02	-.66667-01	-.35322+00	.19454+02	.55468+02	.11093+02	.39078+00
168	.10626+02	.11269+02	-.66667-01	-.32417+00	.17961+02	.55859+02	.10702+02	.39093+00
169	.10252+02	.11385+02	-.66667-01	-.29559+00	.16467+02	.56250+02	.10311+02	.39107+00
170	.98754+01	.11491+02	-.66667-01	-.26744+00	.14973+02	.56641+02	.99198+01	.39122+00
171	.94961+01	.11587+02	-.66667-01	-.23967+00	.13478+02	.57033+02	.95284+01	.39136+00
172	.91142+01	.11673+02	-.66667-01	-.21224+00	.11983+02	.57424+02	.91369+01	.39150+00
173	.87301+01	.11749+02	-.66667-01	-.18510+00	.10487+02	.57816+02	.87453+01	.39164+00
174	.83439+01	.11816+02	-.66667-01	-.15821+00	.89901+01	.58208+02	.83535+01	.39178+00
175	.79561+01	.11872+02	-.66667-01	-.13153+00	.74930+01	.58600+02	.79616+01	.39192+00
176	.75667+01	.11918+02	-.66667-01	-.10502+00	.59954+01	.58992+02	.75695+01	.39205+00
177	.71762+01	.11954+02	-.66667-01	-.78655-01	.44973+01	.59384+02	.71774+01	.39219+00
178	.67847+01	.11979+02	-.66667-01	-.52386-01	.29987+01	.59776+02	.67850+01	.39232+00
179	.63926+01	.11995+02	-.66667-01	-.26179-01	.14996+01	.60169+02	.63926+01	.39246+00
180	.60000+01	.12000+02	.00000	.00000	.00000	.60561+02	.60000+01	.39259+00
181	.56000+01	.12000+02	.00000	.00000	.00000	.60961+02	.56000+01	.40000+00
182	.52000+01	.12000+02	.00000	.00000	.00000	.61361+02	.52000+01	.40000+00
183	.48000+01	.12000+02	.00000	.00000	.00000	.61761+02	.48000+01	.40000+00
184	.44000+01	.12000+02	.00000	.00000	.00000	.62161+02	.44000+01	.40000+00
185	.40000+01	.12000+02	.00000	.00000	.00000	.62561+02	.40000+01	.40000+00
186	.36000+01	.12000+02	.00000	.00000	.00000	.62961+02	.36000+01	.40000+00
187	.32000+01	.12000+02	.00000	.00000	.00000	.63361+02	.32000+01	.40000+00
188	.28000+01	.12000+02	.00000	.00000	.00000	.63761+02	.28000+01	.40000+00
189	.24000+01	.12000+02	.00000	.00000	.00000	.64161+02	.24000+01	.40000+00
190	.20000+01	.12000+02	.00000	.00000	.00000	.64561+02	.20000+01	.40000+00
191	.16000+01	.12000+02	.00000	.00000	.00000	.64961+02	.16000+01	.40000+00
192	.12000+01	.12000+02	.00000	.00000	.00000	.65361+02	.12000+01	.40000+00
193	.80000+00	.12000+02	.00000	.00000	.00000	.65761+02	.80000+00	.40000+00
194	.40000+00	.12000+02	.00000	.00000	.00000	.66161+02	.40000+00	.40000+00
195	.00000	.12000+02	.00000	.00000	.00000	.66561+02	.00000	.40000+00
196	-.40000+00	.12000+02	.00000	.00000	.00000	.66961+02	-.40000+00	.40000+00
197	-.88000+00	.12000+02	.00000	.00000	.00000	.67441+02	-.88000+00	.48000+00
198	-.14560+01	.12000+02	.00000	.00000	.00000	.68017+02	-.14560+01	.57600+00
199	-.21472+01	.12000+02	.00000	.00000	.00000	.68708+02	-.21472+01	.69120+00
200	-.29766+01	.12000+02	.00000	.00000	.00000	.69538+02	-.29766+01	.82944+00
201	-.39720+01	.12000+02	.00000	.00000	.00000	.70533+02	-.39720+01	.99533+00
202	-.51664+01	.12000+02	.00000	.00000	.00000	.71728+02	-.51664+01	.11944+01
203	-.65996+01	.12000+02	.00000	.00000	.00000	.73161+02	-.65996+01	.14333+01
204	-.83196+01	.12000+02	.00000	.00000	.00000	.74881+02	-.83196+01	.17199+01
205	-.10383+02	.12000+02	.00000	.00000	.00000	.76945+02	-.10383+02	.20639+01
206	-.12860+02	.12000+02	.00000	.00000	.00000	.79421+02	-.12860+02	.24767+01
207	-.15832+02	.12000+02	.00000	.00000	.00000	.82393+02	-.15832+02	.29720+01
208	-.19399+02	.12000+02	.00000	.00000	.00000	.85960+02	-.19399+02	.35664+01
209	-.23666+02	.12000+02	.00000	.00000	.00000	.90227+02	-.23666+02	.42669+01
210	-.27932+02	.12000+02	.00000	.00000	.00000	.94494+02	-.27932+02	.42669+01
211	-.32199+02	.12000+02	.00000	.00000	.00000	.98761+02	-.32199+02	.42669+01
212	-.36466+02	.12000+02	.00000	.00000	.00000	.10303+03	-.36466+02	.42669+01
213	-.40733+02	.12000+02	.00000	.00000	.00000	.10729+03	-.40733+02	.42669+01
214	-.45000+02	.12000+02	.00000	.00000	.00000	.11156+03	-.45000+02	.42669+01

XRAK YLO YHI NDY

```
 -.10000+01     .40000+00     .11600+02     20
  .60000+01     .20000+00     .48000+01     10
  .60000+01     .72000+01     .11800+02     10
  .12000+02    -.50000+01    -.30000+01      2
  .14000+02    -.50000+01    -.30000+01      2
  .16000+02    -.50000+01    -.30000+01      2
  .18000+02    -.50000+01    -.30000+01      2
  .20000+02    -.50000+01    -.30000+01      2
```

```
      I            XON               YON              YCNN         AREA           DISC AREA       ENSUBK
   XMIN =    -4.50000+01   XMAX =   9.00000+00   JMIN =     1   JMAX =    48
   XMIN =     3.00000+00   XMAX =   1.10000+01   JMIN =    84   JMAX =    67
                                                                      0.0000
    102        2.1000+01         -4.0000+01       -4.0000+01     0.0000       -4.0000+01      0.0000
```

@PRD,E

@BRKPT PRINTS

UNTRANSFORMED COORDINATE DATA FOR BODY ID = 1,   NOZZLE WITH CENTERBD

I	X(I)	Y(I)	I	X(I)	Y(I)
1	-45.000000	.000000	34	5.600000	.000000
2	-40.733110	.000000	35	6.000000	.000000
3	-36.466210	.000000	36	6.365880	-.022390
4	-32.199320	.000000	37	6.729020	-.089930
5	-27.932430	.000000	38	7.083930	-.202660
6	-23.665530	.000000	39	7.416150	-.355280
7	-19.398640	.000000	40	7.727130	-.547030
8	-15.832200	.000000	41	8.012190	-.774890
9	-12.860170	.000000	42	8.267050	-1.035190
10	-10.383470	.000000	43	8.479830	-1.311680
11	-8.319560	.000000	44	8.660700	-1.618120
12	-6.599630	.000000	45	8.805970	-1.938620
13	-5.166360	.000000	46	8.912520	-2.280830
14	-3.971970	.000000	47	8.977840	-2.636040
15	-2.976640	.000000	48	9.000000	-3.000000
16	-2.147200	.000000	49	9.000000	-3.389400
17	-1.456000	.000000	50	9.000000	-3.856690
18	-.880000	.000000	51	9.000000	-4.417430
19	-.400000	.000000	52	9.000000	-5.090320
20	.000000	.000000	53	9.000000	-5.897790
21	.400000	.000000	54	9.000000	-6.866750
22	.800000	.000000	55	9.000000	-8.029500
23	1.200000	.000000	56	9.000000	-9.424800
24	1.600000	.000000	57	9.000000	-11.099170
25	2.000000	.000000	58	9.000000	-13.108800
26	2.400000	.000000	59	9.000000	-15.519490
27	2.800000	.000000	60	9.000000	-18.412790
28	3.200000	.000000	61	9.000000	-21.884750
29	3.600000	.000000	62	9.000000	-25.507800
30	4.000000	.000000	63	9.000000	-29.130850
31	4.400000	.000000	64	9.000000	-32.753900
32	4.800000	.000000	65	9.000000	-36.376950
33	5.200000	.000000	66	9.000000	-40.000000

ELEMENT COORDINATE DATA FOR BODY ID = 1.   NOZZLE WITH CENTERBO

I	X(I)	Y(I)	DL	DS	SIN(ALF)	COS(ALF)	CURVATURE
1	-45.000000	.000000					
	-42.866555	.000000					
2	-40.733110	.000000	4.266890	4.266890	.000000	1.000000	.000000
	-38.599660	.000000					
3	-36.466210	.000000	4.266900	4.266900	.000000	1.000000	.000000
	-34.332765	.000000					
4	-32.199320	.000000	4.266890	4.266890	.000000	1.000000	.000000
	-30.065875	.000000					
5	-27.932430	.000000	4.266890	4.266890	.000000	1.000000	.000000
	-25.798980	.000000					
6	-23.665530	.000000	4.266900	4.266900	.000000	1.000000	.000000
	-21.532085	.000000					
7	-19.398640	.000000	4.266890	4.266890	.000000	1.000000	.000000
	-17.615420	.000000					
8	-15.832200	.000000	3.566440	3.566440	.000000	1.000000	.000000
	-14.346185	.000000					
9	-12.860170	.000000	2.972030	2.972030	.000000	1.000000	.000000
	-11.621820	.000000					
10	-10.383470	.000000	2.476700	2.476700	.000000	1.000000	.000000
	-9.351515	.000000					
11	-8.319560	.000000	2.063910	2.063910	.000000	1.000000	.000000
	-7.459595	.000000					
12	-6.599630	.000000	1.719930	1.719930	.000000	1.000000	.000000
	-5.882995	.000000					
13	-5.166360	.000000	1.433270	1.433270	.000000	1.000000	.000000
	-4.569165	.000000					
14	-3.971970	.000000	1.194390	1.194390	.000000	1.000000	.000000
	-3.474305	.000000					
15	-2.976640	.000000	.995330	.995330	.000000	1.000000	.000000
	-2.561920	.000000					
16	-2.147200	.000000	.829440	.829440	.000000	1.000000	.000000
	-1.801600	.000000					
17	-1.456000	.000000	.691200	.691200	.000000	1.000000	.000000
	-1.168000	.000000					
18	-.880000	.000000	.576000	.576000	.000000	1.000000	.000000
	-.640000	.000000					
19	-.400000	.000000	.480000	.480000	.000000	1.000000	.000000
	-.200000	.000000					
20	.000000	.000000	.400000	.400000	.000000	1.000000	.000000
	.200000	.000000					
21	.400000	.000000	.400000	.400000	.000000	1.000000	.000000
	.600000	.000000					
22	.800000	.000000	.400000	.400000	.000000	1.000000	.000000
	1.000000	.000000					
23	1.200000	.000000	.400000	.400000	.000000	1.000000	.000000
	1.400000	.000000					
24	1.600000	.000000	.400000	.400000	.000000	1.000000	.000000
	1.800000	.000000					
25	2.000000	.000000	.400000	.400000	.000000	1.000000	.000000

244

ELEMENT COORDINATE DATA FOR BODY ID = 1. NOZZLE WITH CENTERBO

I	X(I)	Y(I)	DL	DS	SIN(ALF)	COS(ALF)	CURVATURE
	2.200000	.000000	.400000	.400000	.000000	1.000000	.000000
26	2.400000	.000000					
	2.600000	.000000	.400000	.400000	.000000	1.000000	.000000
27	2.800000	.000000					
	3.000000	.000000	.400000	.400000	.000000	1.000000	.000000
28	3.200000	.000000					
	3.400000	.000000	.400000	.400000	.000000	1.000000	.000000
29	3.600000	.000000					
	3.800000	.000000	.400000	.400000	.000000	1.000000	.000000
30	4.000000	.000000					
	4.200000	.000000	.400000	.400000	.000000	1.000000	.000000
31	4.400000	.000000					
	4.600000	.000000	.400000	.400000	.000000	1.000000	.000000
32	4.800000	.000000					
	5.000000	.000000	.400000	.400000	.000000	1.000000	.000000
33	5.200000	.000000					
	5.400000	.000000	.400000	.400000	.000000	1.000000	.000000
34	5.600000	.000000					
	5.800000	.000000	.400000	.400000	.000000	1.000000	.000000
35	6.000000	.000000					
	6.183177	-.007327	.366564	.366679	-.061081	.998133	-.230700
36	6.365880	-.022390					
	6.548490	-.050566	.369367	.369601	-.182853	.983140	-.333645
37	6.729020	-.089930					
	6.908226	-.140789	.372383	.372623	-.302726	.953078	-.333618
38	7.083930	-.202660					
	7.252367	-.273904	.365599	.365826	-.417451	.908699	-.333656
39	7.416150	-.355280					
	7.574562	-.446416	.365345	.365571	-.524887	.851197	-.333658
40	7.727130	-.547030					
	7.873128	-.656621	.364938	.365163	-.629381	.781120	-.333669
41	8.012190	-.774890					
	8.143575	-.901160	.364293	.364518	-.714534	.699601	-.333639
42	8.267050	-1.035190					
	8.377462	-1.170339	.348887	.349084	-.792492	.609883	-.333593
43	8.479830	-1.311680					
	8.574709	-1.460242	.352397	.352600	-.858236	.513256	-.333608
44	8.660700	-1.614120					
	8.738146	-1.774216	.355533	.355742	-.912714	.408598	-.333641
45	8.805970	-1.938620					
	8.864360	-2.108132	.358414	.358628	-.954790	.297282	-.333624
46	8.912520	-2.280830					
	8.950530	-2.457451	.361166	.361384	-.983509	.180858	-.333624
47	8.977840	-2.636040					
	8.992769	-2.817786	.364634	.364745	-.998152	.060773	-.231994
48	9.000000	-3.000000					
	9.000000	-3.194700	.389400	.389400	-1.000000	.000000	.000000
49	9.000000	-3.389400					
	9.000000	-3.623045	.467290	.467290	-1.000000	.000000	.000000
50	9.000000	-3.856690					

ELEMENT COORDINATE DATA FOR BODY ID = 1, NOZZLE WITH CENTERBO

I	X(I)	Y(I)	DL	DS	SIN(ALF)	COS(ALF)	CURVATURE
	9.000000	-4.137060	.560740	.560740	-1.000000	.000000	.000000
51	9.000000	-4.417430					
	9.000000	-4.753875	.672890	.672890	-1.000000	.000000	.000000
52	9.000000	-5.090320					
	9.000000	-5.494055	.807470	.807470	-1.000000	.000000	.000000
53	9.000000	-5.897790					
	9.000000	-6.382270	.968960	.968960	-1.000000	.000000	.000000
54	9.000000	-6.866750					
	9.000000	-7.448125	1.162750	1.162750	-1.000000	.000000	.000000
55	9.000000	-8.029500					
	9.000000	-8.727150	1.395300	1.395300	-1.000000	.000000	.000000
56	9.000000	-9.424800					
	9.000000	-10.261985	1.674370	1.674370	-1.000000	.000000	.000000
57	9.000000	-11.099170					
	9.000000	-12.103785	2.009230	2.009230	-1.000000	.000000	.000000
58	9.000000	-13.108400					
	9.000000	-14.313945	2.411090	2.411090	-1.000000	.000000	.000000
59	9.000000	-15.519490					
	9.000000	-16.966140	2.893300	2.893300	-1.000000	.000000	.000000
60	9.000000	-18.412790					
	9.000000	-20.148770	3.471960	3.471960	-1.000000	.000000	.000000
61	9.000000	-21.884750					
	9.000000	-23.696275	3.623050	3.623050	-1.000000	.000000	.000000
62	9.000000	-25.507803					
	9.000000	-27.319325	3.623050	3.623050	-1.000000	.000000	.000000
63	9.000000	-29.130850					
	9.000000	-30.942375	3.623050	3.623050	-1.000000	.000000	.000000
64	9.000000	-32.753900					
	9.000000	-34.565425	3.623050	3.623050	-1.000000	.000000	.000000
65	9.000000	-36.376950					
	9.000000	-38.188475	3.623050	3.623050	-1.000000	.000000	.000000
66	9.000000	-40.000000					

SUMDS = 92.712139

UNTRANSFORMED COORDINATE DATA FOR BODY ID = 2, NOZZLE WITH CENTERBD

I	X(I)	Y(I)	I	X(I)	Y(I)
1	11.000000	6.000000	19	3.090100	6.243240
2	9.894830	5.778970	20	3.315830	6.446620
3	8.918840	5.683770	21	3.608100	6.603580
4	8.105500	5.421100	22	3.932580	6.724570
5	7.427730	5.285540	23	4.267440	6.816380
6	6.862910	5.172580	24	4.608830	6.885980
7	6.392230	5.078450	25	4.954110	6.937260
8	6.000000	5.000000	26	5.301760	6.972540
9	5.650680	5.006800	27	5.650680	6.993200
10	5.301760	5.027460	28	6.000000	7.000000
11	4.954110	5.062740	29	6.392230	6.921550
12	4.608830	5.114020	30	6.862910	6.827420
13	4.267440	5.183620	31	7.427730	6.714460
14	3.932380	5.275430	32	8.105500	6.578900
15	3.608100	5.396420	33	8.918840	6.416230
16	3.315830	5.553380	34	9.894830	6.221030
17	3.090100	5.756760	35	11.000000	6.000000
18	3.000000	6.000000			

ELEMENT COORDINATE DATA FOR BODY ID = 2, NOZZLE WITH CENTERBO

I	X(I)	Y(I)	DL	DS	SIN(ALF)	COS(ALF)	CURVATURE
1	11.000000	6.000000					
	10.447415	5.889486	1.127056	1.127056	-.196113	-.980581	.000005
2	9.894830	5.778970					
	9.406835	5.681370	.995319	.995319	-.196118	-.980580	.000001
3	8.918840	5.583770					
	8.512170	5.502435	.829448	.829449	-.196118	-.980580	.000002
4	8.105500	5.421100					
	7.766615	5.353320	.691194	.691194	-.196125	-.980579	.000000
5	7.427730	5.285540					
	7.145320	5.229059	.576005	.576005	-.196109	-.980582	-.000016
6	6.862910	5.172580					
	6.627570	5.125515	.480000	.480000	-.196104	-.980583	.000000
7	6.392230	5.078850					
	6.196115	5.039225	.399998	.399998	-.196126	-.980579	.000000
8	6.000000	5.000000					
	5.825264	4.999488	.349386	.349503	.019483	-.999811	-.256431
9	5.650680	5.006800					
	5.476115	5.015349	.349531	.349555	.059108	-.998252	-.116807
10	5.301760	5.027460					
	5.127741	5.043183	.349436	.349464	.100963	-.994890	-.126222
11	4.954110	5.062740					
	4.781150	5.086228	.349067	.349103	.146986	-.989151	-.142858
12	4.608830	5.114020					
	4.437616	5.146273	.348412	.348464	.199763	-.979844	-.171316
13	4.267440	5.183620					
	4.099025	5.226294	.347411	.347497	.264269	-.964449	-.222055
14	3.932380	5.275430					
	3.768555	5.331409	.346116	.346295	.349565	-.936912	-.321881
15	3.610100	5.396420					
	3.458378	5.468221	.331750	.332212	.473127	-.880994	-.551048
16	3.315830	5.553380					
	3.194132	5.645266	.303838	.305366	.669370	-.742929	-1.143543
17	3.090100	5.756760					
	3.027862	5.872013	.259391	.262845	.937735	-.347352	-2.179330
18	3.000000	6.000000					
	3.027862	6.127987	.259391	.262845	.937735	.347352	-2.179330
19	3.090100	6.243240					
	3.194132	6.354734	.303838	.305366	.669370	.742929	-1.143543
20	3.315830	6.446620					
	3.458378	6.531779	.331750	.332212	.473127	.880994	-.551048
21	3.608100	6.603580					
	3.768555	6.668591	.346116	.346295	.349565	.936912	-.321881
22	3.932380	6.724570					
	4.099025	6.773706	.347411	.347497	.264269	.964449	-.222055
23	4.267440	6.816380					
	4.437616	6.853727	.348412	.348460	.199763	.979844	-.171316
24	4.608830	6.885980					
	4.781150	6.913772	.349067	.349103	.146986	.989151	-.142858
25	4.958110	6.937260					

ELEMENT COORDINATE DATA FOR BODY ID = 2, NOZZLE WITH BENTERBO

I	X(I)	Y(I)	DL	DS	SIN(ALF)	COS(ALF)	CURVATURE
	5.127741	6.956817	.349436	.349466	.100963	.994890	-.126222
26	5.301760	6.972540					
	5.476115	6.984651	.349531	.349555	.059108	.998252	-.116807
27	5.650680	6.993200					
	5.825264	7.000512	.349386	.349505	.019463	.999811	-.256431
28	6.000000	7.000000					
	6.196115	6.960775	.399998	.399998	-.196126	.980579	.000000
29	6.392230	6.921550					
	6.627570	6.874485	.480000	.480000	-.196104	.980583	.000000
30	6.862910	6.827420					
	7.145320	6.770941	.576005	.576005	-.196109	.980582	-.000016
31	7.427730	6.714460					
	7.766615	6.646680	.691194	.691194	-.196125	.980579	.000000
32	8.105500	6.578900					
	8.512170	6.497565	.829448	.829448	-.196118	.980580	.000002
33	8.918840	6.416230					
	9.406835	6.318633	.995319	.995319	-.196118	.980580	.000001
34	9.894830	6.221030					
	10.447415	6.110514	1.127056	1.127056	-.196113	.980581	.000005
35	11.000000	6.000000					

SUMDS = 16.878647

UNTRANSFORMED COORDINATE DATA FOR BODY ID = 3, NOZZLE WITH CENTERBO

I	X(I)	Y(I)	I	X(I)	Y(I)
1	21.000000	-40.000000	51	15.983070	8.186180
2	21.000000	-36.479730	52	15.700620	8.441070
3	21.000000	-32.959460	53	15.401100	8.688430
4	21.000000	-29.439180	54	15.095100	8.928080
5	21.000000	-25.918910	55	14.782830	9.159850
6	21.000000	-22.398640	56	14.464490	9.383550
7	21.000000	-18.832200	57	14.140300	9.599030
8	21.000000	-15.860170	58	13.810460	9.806120
9	21.000000	-13.383870	59	13.476210	10.004660
10	21.000000	-11.319560	60	13.134758	10.194520
11	21.000000	-9.599630	61	12.789820	10.375590
12	21.000000	-8.166360	62	12.439160	10.547590
13	21.000000	-6.971970	63	12.084400	10.710500
14	21.000000	-5.976640	64	11.725560	10.864270
15	21.000000	-5.187200	65	11.362600	11.008660
16	21.000000	-4.456000	66	10.995860	11.143600
17	21.000000	-3.880000	67	10.625590	11.268990
18	21.000000	-3.400000	68	10.252030	11.384720
19	21.000000	-3.000000	69	9.875450	11.490720
20	20.994860	-2.607310	70	9.496090	11.586890
21	20.979940	-2.214750	71	9.114210	11.673160
22	20.953740	-1.822600	72	8.730080	11.749460
23	20.917770	-1.431120	73	8.343950	11.815730
24	20.871540	-1.340580	74	7.956080	11.871910
25	20.814970	-.651250	75	7.566790	11.917950
26	20.748260	-.263400	76	7.176190	11.953810
27	20.671360	.122700	77	6.784710	11.979460
28	20.584320	.506780	78	6.392560	11.994860
29	20.487200	.888590	79	6.000000	12.000000
30	20.380030	1.267860	80	5.600000	12.000000
31	20.262900	1.644320	81	5.200000	12.000000
32	20.135860	2.017720	82	4.800000	12.000000
33	19.998990	2.387790	83	4.400000	12.000000
34	19.852380	2.754270	84	4.000000	12.000000
35	19.696110	3.116920	85	3.600000	12.000000
36	19.530270	3.475480	86	3.200000	12.000000
37	19.354970	3.829690	87	2.800000	12.000000
38	19.170320	4.179320	88	2.400000	12.000000
39	18.976430	4.524110	89	2.000000	12.000000
40	18.773420	4.863830	90	1.600000	12.000000
41	18.561410	5.198230	91	1.200000	12.000000
42	18.340540	5.527080	92	.800000	12.000000
43	18.110940	5.850150	93	.400000	12.000000
44	17.872760	6.167200	94	.000000	12.000000
45	17.626130	6.478030	95	-.400000	12.000000
46	17.371230	6.782390	96	-.880000	12.000000
47	17.108190	7.080090	97	-1.456000	12.000000
48	16.837190	7.370890	98	-2.147200	12.000000
49	16.558380	7.659600	99	-2.976640	12.000000
50	16.279450	7.923960	100	-3.971970	12.000000

UNTRANSFORMED COORDINATE DATA FOR BODY ID = 3, NOZZLE WITH CENTERBO

I	X(I)	Y(I)	I	X(I)	Y(I)
101	-5.166360	12.000000	108	-23.665630	12.000000
102	-6.599630	12.000000	109	-27.932430	12.000000
103	-8.319560	12.000000	110	-32.199320	12.000000
104	-10.383470	12.000000	111	-36.466210	12.000000
105	-12.860170	12.000000	112	-40.733110	12.000000
106	-15.832200	12.000000	113	-45.000000	12.000000
107	-19.398640	12.000000			

ELEMENT COORDINATE DATA FOR BODY ID = 3, NOZZLE WITH CENTERBO

I	X(I)	Y(I)	DL	DS	SIN(ALF)	COS(ALF)	CURVATURE
1	21.000000	-40.000000					
	21.000000	-38.239865	3.520270	3.520270	1.000000	.000000	.000000
2	21.000000	-36.479731					
	21.000000	-34.719595	3.520270	3.520270	1.000000	.000000	.000000
3	21.000000	-32.959460					
	21.000000	-31.199320	3.520280	3.520280	1.000000	.000000	.000000
4	21.000000	-29.439180					
	21.000000	-27.679045	3.520270	3.520270	1.000000	.000000	.000000
5	21.000000	-25.918910					
	21.000000	-24.158775	3.520270	3.520270	1.000000	.000000	.000000
6	21.000000	-22.398640					
	21.000000	-20.615420	3.566440	3.566440	1.000000	.000000	.000000
7	21.000000	-18.832200					
	21.000000	-17.346185	2.972030	2.972030	1.000000	.000000	.000000
8	21.000000	-15.860170					
	21.000000	-14.621820	2.476700	2.476700	1.000000	.000000	.000000
9	21.000000	-13.383470					
	21.000000	-12.351515	2.063910	2.063910	1.000000	.000000	.000000
10	21.000000	-11.319560					
	21.000000	-10.459595	1.719930	1.719930	1.000000	.000000	.000000
11	21.000000	-9.599630					
	21.000000	-8.882995	1.433270	1.433270	1.000000	.000000	.000000
12	21.000000	-8.166360					
	21.000000	-7.569165	1.194390	1.194390	1.000000	.000000	.000000
13	21.000000	-6.971970					
	21.000000	-6.474305	.995330	.995330	1.000000	.000000	.000000
14	21.000000	-5.976640					
	21.000000	-5.561920	.829440	.829440	1.000000	.000000	.000000
15	21.000000	-5.147200					
	21.000000	-4.801600	.691200	.691200	1.000000	.000000	.000000
16	21.000000	-4.456000					
	21.000000	-4.168000	.576000	.576000	1.000000	.000000	.000000
17	21.000000	-3.880000					
	21.000000	-3.640000	.480000	.480000	1.000000	.000000	.000000
18	21.000000	-3.400000					
	21.000000	-3.200000	.400000	.400000	1.000000	.000000	.000000
19	21.000000	-3.000000					
	20.998334	-2.803643	.392724	.392729	.999914	-.013088	.046907
20	20.994860	-2.607310					
	20.988434	-2.410988	.392863	.392874	.999229	-.039250	.066633
21	20.979440	-2.214750					
	20.967873	-2.018591	.392991	.393007	.997859	-.065396	.066623
22	20.953740	-1.822600					
	20.937036	-1.626742	.393129	.393140	.995805	-.091497	.066624
23	20.917770	-1.431120					
	20.895940	-1.235698	.393267	.393278	.993067	-.117553	.066960
24	20.871540	-1.040580					
	20.844532	-.845729	.393418	.393430	.989688	-.143791	.066732
25	20.818970	-.651250					

ELEMENT COORDINATE DATA FOR BODY ID = 3. NOZZLE WITH CENTERBO

I	X(I)	Y(I)	DL	DS	SIN(ALF)	COS(ALF)	CURVATURE
	20.782883	-.457107	.393545	.393556	.985528	-.169510	.066456
26	20.748260	-.263400					
	20.711077	-.070098	.393684	.393695	.980737	-.195335	.066707
27	20.671360	.122700					
	20.629100	.315026	.393819	.393830	.975270	-.221015	.066638
28	20.584320	.506780					
	20.537014	.698004	.393969	.393980	.969138	-.246517	.066674
29	20.487200	.888590					
	20.434861	1.078577	.394121	.394132	.962319	-.271922	.066676
30	20.380030	1.267860					
	20.322701	1.456475	.394261	.394272	.954880	-.297087	.066645
31	20.262900	1.644320					
	20.200608	1.831438	.394420	.394431	.946704	-.322094	.066689
32	20.135860	2.017720					
	20.068642	2.203215	.394570	.394581	.937908	-.346884	.066666
33	19.998990	2.387790					
	19.926890	2.571512	.394718	.394729	.928461	-.371430	.066656
34	19.852380	2.754270					
	19.775439	2.936109	.394886	.394898	.918365	-.395734	.066685
35	19.696110	3.116920					
	19.614371	3.296746	.395055	.395068	.907621	-.419790	.066696
36	19.530270	3.475480					
	19.443786	3.653162	.395215	.395226	.896286	-.443556	.066652
37	19.354970	3.829690					
	19.263797	4.005113	.395394	.395406	.884286	-.467002	.066652
38	19.170320	4.179320					
	19.074511	4.352354	.395567	.395579	.871684	-.490157	.066665
39	18.976430	4.524110					
	18.876045	4.694680	.395756	.395767	.858408	-.512967	.066679
40	18.773420	4.863830					
	18.668518	5.031730	.395944	.395956	.844564	-.535455	.066680
41	18.561410	5.198230					
	18.452060	5.363384	.396139	.396150	.830138	-.557557	.066660
42	18.340540	5.527080					
	18.226807	5.689373	.396346	.396358	.815121	-.579291	.066676
43	18.110940	5.850150					
	17.992898	6.009462	.396548	.396560	.799525	-.600633	.066673
44	17.872760	6.167200					
	17.750473	6.323430	.396789	.396801	.783363	-.621564	.066659
45	17.626130	6.478030					
	17.499686	6.631053	.397000	.397012	.766650	-.642066	.066658
46	17.371230	6.782390					
	17.240695	6.932111	.397260	.397271	.749384	-.662136	.066683
47	17.108190	7.080090					
	16.973653	7.226388	.397499	.397511	.731574	-.681762	.066694
48	16.837190	7.370890					
	16.698725	7.513669	.397777	.397788	.713239	-.700921	.066652
49	16.558380	7.654600					
	16.419785	7.790181	.387759	.387769	.694659	-.719339	.066645
50	16.279450	7.923960					

ELEMENT COORDINATE DATA FOR BODY ID = 3, NOZZLE WITH CENTERBO

I	X(I)	Y(I)	DL	DS	SIN(ALF)	COS(ALF)	CURVATURE
	16.137308	8.055995	.388000	.388011	.675825	-.737062	.066676
51	15.993470	8.186180					
	15.847870	8.314573	.388240	.388250	.656528	-.754302	.066682
52	15.700620	8.441070					
	15.551661	8.565720	.388457	.388468	.636775	-.771050	.066690
53	15.401100	8.688430					
	15.248876	8.809246	.388675	.388686	.616582	-.787290	.066653
54	15.095100	8.928080					
	14.939716	9.044977	.388883	.388894	.595989	-.802992	.066671
55	14.782830	9.159850					
	14.624385	9.272732	.389078	.389089	.574998	-.818190	.066667
56	14.464490	9.383550					
	14.303098	9.492381	.389270	.389280	.553580	-.832816	.066658
57	14.140300	9.599830					
	13.976052	9.703646	.389462	.389473	.531733	-.846912	.066690
58	13.810460	9.806120					
	13.643479	9.906478	.389629	.389640	.509562	-.860434	.066657
59	13.475210	10.004660					
	13.305597	10.100696	.389820	.389831	.487085	-.873377	.066673
60	13.134750	10.194520					
	12.962623	10.286153	.389987	.389998	.464169	-.885747	.066673
61	12.789320	10.375540					
	12.614799	10.462703	.390145	.390156	.440990	-.897512	.066666
62	12.439160	10.547590					
	12.262355	10.630219	.390312	.390323	.417486	-.908683	.066669
63	12.084490	10.710540					
	11.905525	10.788578	.390466	.390477	.393709	-.919235	.066663
64	11.725560	10.864270					
	11.544550	10.937647	.390626	.390637	.369638	-.929176	.066679
65	11.362600	11.008660					
	11.179669	11.077324	.390777	.390789	.345312	-.938488	.066658
66	10.995860	11.143600					
	10.811134	11.207502	.390925	.390936	.320752	-.947163	.066695
67	10.625590	11.268990					
	10.439187	11.328072	.391076	.391087	.295927	-.955211	.066666
68	10.252030	11.384720					
	10.064085	11.438947	.391214	.391225	.270952	-.962593	.066652
69	9.875450	11.490720					
	9.686084	11.540043	.391360	.391371	.245733	-.969338	.066695
70	9.496090	11.586890					
	9.305431	11.631271	.391503	.391514	.220356	-.975420	.066664
71	9.114210	11.673160					
	8.922394	11.712563	.391634	.391646	.194825	-.980838	.066656
72	8.730080	11.749460					
	8.537231	11.783856	.391776	.391787	.169153	-.985590	.066667
73	8.343950	11.815730					
	8.150198	11.845087	.391918	.391929	.143346	-.989673	.066680
74	7.956080	11.871910					
	7.761560	11.896202	.392053	.392064	.117433	-.993081	.066675
75	7.566740	11.917950					

ELEMENT COORDINATE DATA FOR BODY ID = 3, NOZZLE WITH CENTERBO

I	X(I)	Y(I)	DL	DS	SIN(ALF)	COS(ALF)	CURVATURE
	7.371582	11.937156	.392193	.392208	.091434	-.995811	.066650
76	7.176190	11.953810					
	6.980534	11.967915	.392319	.392331	.065380	-.997860	.066670
77	6.784710	11.979460					
	6.588685	11.988443	.392452	.392463	.039241	-.999230	.066677
78	6.392560	11.994860					
	6.196292	11.998334	.392594	.392599	.013092	-.999914	.046922
79	6.000000	12.000000					
	5.800000	12.000000	.400000	.400000	.000000	-1.000000	.000000
80	5.600000	12.000000					
	5.400000	12.000000	.400000	.400000	.000000	-1.000000	.000000
81	5.200000	12.000000					
	5.000000	12.000000	.400000	.400000	.000000	-1.000000	.000000
82	4.800000	12.000000					
	4.600000	12.000000	.400000	.400000	.000000	-1.000000	.000000
83	4.400000	12.000000					
	4.200000	12.000000	.400000	.400000	.000000	-1.000000	.000000
84	4.000000	12.000000					
	3.800000	12.000000	.400000	.400000	.000000	-1.000000	.000000
85	3.600000	12.000000					
	3.400000	12.000000	.400000	.400000	.000000	-1.000000	.000000
86	3.200000	12.000000					
	3.000000	12.000000	.400000	.400000	.000000	-1.000000	.000000
87	2.800000	12.000000					
	2.600000	12.000000	.400000	.400000	.000000	-1.000000	.000000
88	2.400000	12.000000					
	2.200000	12.000000	.400000	.400000	.000000	-1.000000	.000000
89	2.000000	12.000000					
	1.800000	12.000000	.400000	.400000	.000000	-1.000000	.000000
90	1.600000	12.000000					
	1.400000	12.000000	.400000	.400000	.000000	-1.000000	.000000
91	1.200000	12.000000					
	1.000000	12.000000	.400000	.400000	.000000	-1.000000	.000000
92	.800000	12.000000					
	.600000	12.000000	.400000	.400000	.000000	-1.000000	.000000
93	.400000	12.000000					
	.200000	12.000000	.400000	.400000	.000000	-1.000000	.000000
94	.000000	12.000000					
	-.200000	12.000000	.400000	.400000	.000000	-1.000000	.000000
95	-.400000	12.000000					
	-.640000	12.000000	.480000	.480000	.000000	-1.000000	.000000
96	-.880000	12.000000					
	-1.168000	12.000000	.576000	.576000	.000000	-1.000000	.000000
97	-1.456000	12.000000					
	-1.801600	12.000000	.691200	.691200	.000000	-1.000000	.000000
98	-2.147200	12.000000					
	-2.561920	12.000000	.829440	.829440	.000000	-1.000000	.000000
99	-2.976640	12.000000					
	-3.474305	12.000000	.995330	.995330	.000000	-1.000000	.000000
100	-3.971970	12.000000					

ELEMENT COORDINATE DATA FOR BODY ID = 3, NOZZLE WITH CENTERBO

I	X(I)	Y(I)	DL	DS	SIN(ALF)	COS(ALF)	CURVATURE
	-4.569165	12.000000	1.194390	1.194390	.000080	-1.000000	.000000
101	-5.166360	12.000000					
	-5.882995	12.000000	1.433270	1.433270	.000080	-1.000000	.000000
102	-6.599630	12.000000					
	-7.459595	12.000000	1.719930	1.719930	.000080	-1.000000	.000000
103	-8.319560	12.000000					
	-9.351515	12.000000	2.063910	2.063910	.000080	-1.000000	.000000
104	-10.383470	12.000000					
	-11.621820	12.000000	2.476700	2.476700	.000080	-1.000000	.000000
105	-12.860170	12.000000					
	-14.346185	12.000000	2.972030	2.972030	.000080	-1.000000	.000000
106	-15.832200	12.000000					
	-17.615420	12.000000	3.566440	3.566440	.000080	-1.000000	.000000
107	-19.398640	12.000000					
	-21.532085	12.000000	4.266890	4.266890	.000080	-1.000000	.000000
108	-23.665530	12.000000					
	-25.798980	12.000000	4.266900	4.266900	.000080	-1.000000	.000000
109	-27.932430	12.000000					
	-30.065875	12.000000	4.266890	4.266890	.000080	-1.000000	.000000
110	-32.199320	12.000000					
	-34.332765	12.000000	4.266890	4.266890	.000080	-1.000000	.000000
111	-36.466210	12.000000					
	-38.599660	12.000000	4.266900	4.266900	.000080	-1.000000	.000000
112	-40.733110	12.000000					
	-42.866555	12.000000	4.266890	4.266890	.000080	-1.000000	.000000
113	-45.000000	12.000000					

SUMDS = 111.561986

BODY GEOMETRY SUMMARY

BODY DESCRIPTION	BODY ID	LIFT TYPE	N/O	SID	TFORM	NORM	CHORD	TYPE	SIGMA F	C	S	C	ELEMENT STORAGE N/O	FIRST	NO.
NOZZLE WITH CENTERBO	1	YES	NEW		NO	NO	.300000	P	1	1	1	1	NEW	1	65
NOZZLE WITH CENTERBO	2	YES	NEW		NO	NO	.090000	P	1	1	1	1	NEW	66	34
NOZZLE WITH CENTERBO	3	YES	NEW		NO	NO	.000000	P	1	1	1	1	NEW	100	112

TOTAL NUMBER OF BODIES = 3

TOTAL NUMBER OF ELEMENTS = 211

ELFORM COMPLETE, CALL MAFORM, T = .000SECONDS.

VORTICITY WEIGHTING FUNCTION = S/L*(1 - S/L)

.022882	.064269	.101819	.135134	.164212	.189054
.208127	.221315	.230405	.236661	.240969	.243903
.245916	.247286	.248214	.248843	.249259	.249536
.249718	.249845	.249933	.249985	.250000	.249977
.249917	.249820	.249686	.249514	.249305	.249059
.248776	.248455	.248097	.247702	.247289	.246860
.246395	.245901	.245380	.244829	.244248	.243649
.243032	.242379	.241692	.240969	.240209	.239388
.238415	.237190	.235640	.233668	.231182	.227831
.223532	.217872	.210356	.230294	.186720	.168271
.144929	.118068	.088162	.055182	.019158	

VORTICITY WEIGHTING FUNCTION = S/L*(1 - S/L)

.032272	.086993	.127720	.157195	.178657	.194389
.206005	.214826	.222165	.228645	.234265	.239022
.242918	.245954	.248107	.249394	.249989	.249939
.249394	.248107	.245954	.242918	.239022	.234265
.228645	.222165	.214826	.206005	.194390	.178657
.157195	.127720	.086993	.032272		

VORTICITY WEIGHTING FUNCTION = S/L*(1 - S/L)

.015528	.045091	.072663	.098245	.121832	.143565
.161827	.175733	.186411	.194676	.201124	.206193
.210205	.213401	.215962	.218028	.219695	.221053
.222249	.223410	.224546	.225658	.226745	.227808
.228847	.229861	.230850	.231816	.232755	.233670
.234561	.235428	.236269	.237086	.237878	.238646
.239388	.240106	.240799	.241467	.242110	.242729
.243322	.243891	.244434	.244956	.245446	.245914
.246352	.246760	.247188	.247604	.247890	.248152
.248439	.248703	.248942	.249157	.249548	.249514
.249656	.249773	.249867	.249935	.249979	.249999
.249994	.249964	.249910	.249831	.249728	.249600
.249447	.249269	.249067	.248840	.248588	.248311
.248007	.247674	.247315	.246930	.246520	.246084
.245623	.245136	.244623	.244084	.243519	.242929
.242313	.241672	.241005	.240312	.239520	.238528
.237280	.235696	.233673	.231068	.227689	.223267
.217434	.209675	.199271	.185212	.166108	.142489
.115945	.086475	.054079	.018758		

MAFORM COMPLETE. CALL SOLVE. T = .000SECONDS.

SOLVIT TIME = .000 SECONDS.

THE 211 X 211 MATRIX WITH 5 RIGHT SIDES WAS SOLVED DIRECTLY IN .000 MINUTES.

SOLVIT TIME = .000 SECONDS.

SOLVE COMPLETE. READ FLOW TITLE & CONTROL CARD. CALL COMBO. T = .000SECONDS.

ON RETURN FROM MIS1. NERR = 0

```
COMBINATION CONSTANTS
        ALPHA  =          0             90            .000000
          1         -.151631       .134504       -.151631
          2         -.007417       .008360       -.087417
          3          .926002      -.788141        .926002

LIFT CURVE CONSTANTS
        RK1 =    14.858142  RK2 =   -12.556113  RK3 =      .000000
        ALPHA0 =   130.200014
        ALPHA  =       .000000
        CLT    =   373.425842
COMBO COMPLETE, CALL FLOWS, T =       .000SECONDS.
```

INDIVIDUAL FLOW NUMBER 1

PT.NO.	VN	VT	SIGMA
1	.000000	.538021	.012913
2	.000000	.540255	.016034
3	.000000	.540333	.019160
4	.000000	.540226	.022292
5	.000000	.540219	.025521
6	.000000	.540345	.028969
7	.000000	.540778	.032468
8	.000000	.541715	.035762
9	.000000	.543885	.038886
10	.000000	.546240	.041853
11	.000000	.550332	.044665
12	.000000	.555838	.047302
13	.000000	.562712	.049787
14	.000000	.570770	.051977
15	.000000	.579765	.053978
16	.000000	.589258	.055739
17	.000000	.598854	.057265
18	.000000	.608207	.058569
19	.000000	.617090	.059673
20	.000000	.625960	.060684
21	.000000	.635761	.061696
22	.000000	.646456	.062705
23	.000000	.658100	.063708
24	.000000	.670656	.064695
25	-.000000	.684123	.065669
26	.000000	.698524	.066632
27	.000000	.713907	.067587
28	-.000000	.730400	.068546
29	-.000000	.748326	.069527
30	-.000000	.768270	.070545
31	-.000000	.791309	.071620
32	-.000000	.819573	.072757
33	-.000000	.857507	.073897
34	-.000000	.927918	.074656
35	-.000000	1.033690	.069888
36	-.000000	1.111885	.049184
37	.000000	1.147150	.036263
38	.000000	1.167324	.024533
39	.000000	1.176368	.013694
40	.000000	1.177476	.003347
41	.000000	1.171941	-.006742
42	.000000	1.160240	-.016577
43	.000000	1.142017	-.026391
44	.000000	1.115539	-.036611
45	.000000	1.078498	-.047432
46	.000000	1.029065	-.059144
47	.000000	.942387	-.073725
48	.000000	.835184	-.083479
49	.000000	.759448	-.083074
50	.000000	.711291	-.081730
51	.000000	.673120	-.080001
52	-.000000	.641805	-.078079
53	.000000	.615959	-.076093
54	.000000	.594805	-.074155

55	-.000000	.577846	-.072380
56	.000000	.564718	-.070884
57	.000000	.555823	-.069803
58	.000000	.548215	-.069295
59	.000000	.543512	-.069547
60	.000000	.539655	-.070841
61	.000000	.534203	-.073433
62	.000000	.522242	-.077458
63	.000000	.492124	-.083271
64	.000000	.413423	-.091368
65	.000000	.175095	-.101797
66	-.000000	.128797	-.301799
67	-.000000	-.284804	-.181558
68	-.000000	-.407885	-.138063
69	.000000	-.512437	-.115352
70	.000000	-.601777	-.101374
71	.000000	-.687209	-.091875
72	.000000	-.815228	-.083518
73	-.000000	-.921897	-.050925
74	-.000000	-.875905	-.049581
75	-.000000	-.890119	-.044736
76	-.000000	-.902535	-.039302
77	-.000000	-.914280	-.033330
78	-.000000	-.924987	-.026549
79	-.000000	-.934267	-.018236
80	-.000000	-.939054	-.006800
81	-.000000	-.914630	.011705
82	-.000000	-.704562	.043834
83	-.000000	-.198528	.069112
84	-.000000	.205329	.067164
85	-.000000	.380496	.062368
86	-.000000	.462906	.059035
87	.000000	.508965	.057078
88	-.000000	.538365	.056064
89	-.000000	.559628	.055680
90	-.000000	.577254	.055702
91	-.000000	.597152	.055863
92	-.000000	.627039	.053962
93	-.000000	.644518	.042098
94	-.000000	.563293	.055945
95	-.000000	.546409	.070896
96	-.000000	.540562	.089995
97	-.000000	.546928	.117221
98	-.000000	.561381	.164766
99	-.000000	.844420	.289031
100	.000001	-.941931	.135812
101	.000001	-.677051	.131870
102	.000001	-.593390	.130208
103	.000001	-.561299	.130746
104	.000001	-.548412	.133109
105	.000001	-.542555	.137073
106	.000001	-.538662	.142153
107	.000001	-.534322	.147617
108	.000001	-.528342	.153266
109	.000001	-.520296	.158971
110	.000001	-.510249	.164640
111	.000000	-.498564	.170209

112	.000001	-.485777	.175648
113	.000001	-.472385	.180960
114	.000000	-.458751	.186180
115	.000000	-.445113	.191389
116	.000000	-.431615	.196740
117	.000000	-.415808	.202689
118	.000000	-.400723	.209635
119	.000000	-.389819	.214196
120	.000000	-.379369	.217003
121	.000000	-.371508	.219190
122	.000000	-.364935	.220927
123	.000000	-.359592	.222231
124	.000000	-.354545	.223238
125	.000000	-.350313	.223992
126	-.000000	-.346519	.224480
127	-.000000	-.343116	.224748
128	-.000000	-.340178	.224798
129	-.000000	-.337529	.224652
130	-.000000	-.335255	.224322
131	-.000000	-.333295	.223811
132	-.000000	-.331634	.223132
133	-.000000	-.330294	.222293
134	-.000000	-.329131	.221295
135	-.000000	-.328279	.220144
136	-.000001	-.327674	.218851
137	-.000000	-.327371	.217417
138	-.000000	-.327294	.215848
139	-.000000	-.327558	.214145
140	-.000001	-.328022	.212314
141	-.000001	-.328805	.210362
142	-.000001	-.329884	.208289
143	-.000001	-.331282	.206103
144	-.000001	-.332990	.203807
145	-.000001	-.335009	.201910
146	-.000001	-.337365	.198911
147	-.000001	-.340082	.196316
148	-.000001	-.343041	.193672
149	-.000001	-.346280	.190987
150	-.000001	-.349796	.188230
151	-.000001	-.353560	.185410
152	-.000001	-.357548	.182526
153	-.000000	-.361667	.179595
154	-.000000	-.365906	.176612
155	-.000001	-.370100	.173587
156	-.000001	-.374238	.170519
157	-.000001	-.378199	.167405
158	-.000000	-.381903	.164249
159	-.000000	-.385312	.161042
160	-.000001	-.388366	.157781
161	-.000000	-.391051	.154459
162	-.000000	-.393334	.151073
163	-.000000	-.395284	.147615
164	-.000000	-.396938	.144079
165	-.000000	-.398353	.140463
166	-.000000	-.399655	.136758
167	-.000000	-.400898	.132969
168	-.000000	-.402080	.129092

169	-.000000	-.403401	.125121
170	-.000000	-.404915	.121059
171	-.000000	-.406674	.116900
172	-.000000	-.408798	.112636
173	-.000000	-.411398	.108256
174	-.000000	-.414682	.103742
175	-.000000	-.418760	.099063
176	-.000000	-.423711	.094159
177	-.000000	-.433473	.088563
178	-.000000	-.446155	.083853
179	-.000000	-.455308	.080201
180	-.000000	-.460967	.077678
181	-.000000	-.464982	.075486
182	-.000000	-.467960	.073508
183	-.000000	-.470288	.071679
184	-.000000	-.472233	.069963
185	-.000000	-.474014	.068336
186	-.000000	-.475815	.066788
187	-.000000	-.477725	.065310
188	-.000000	-.479816	.063899
189	-.000000	-.482129	.062555
190	-.000000	-.484696	.061276
191	-.000000	-.487354	.060064
192	-.000000	-.490205	.058915
193	-.000000	-.493162	.057832
194	-.000000	-.496437	.056708
195	-.000000	-.500350	.055450
196	-.000000	-.504921	.054058
197	-.000000	-.510013	.052537
198	-.000000	-.515474	.050890
199	-.000000	-.520966	.049117
200	-.000000	-.526141	.047205
201	-.000000	-.530659	.045138
202	-.000000	-.534245	.042881
203	-.000000	-.536830	.040400
204	-.000000	-.538490	.037658
205	-.000000	-.539398	.034617
206	-.000000	-.539832	.031297
207	-.000000	-.539979	.027822
208	-.000000	-.539974	.024569
209	-.000000	-.539730	.021809
210	-.000000	-.538443	.018286
211	-.000000	-.527832	.015214

INDIVIDUAL FLOW NUMBER 2

PT.NO.	VN	VT	SIGMA
1	.000000	-.151918	-.099427
2	.000000	-.375604	-.087920
3	.000000	-.440228	-.079460
4	.000000	-.461093	-.073538
5	.000000	-.467888	-.069271
6	.000000	-.470207	-.066059
7	.000000	-.471269	-.063784
8	.000000	-.472329	-.062846
9	.000000	-.473959	-.061511
10	.000000	-.476422	-.061138
11	.000000	-.480009	-.061113
12	.000000	-.484822	-.061341
13	.000000	-.490825	-.061739
14	.000000	-.497873	-.062232
15	.000000	-.505706	-.062761
16	.000000	-.513988	-.063281
17	.000000	-.522366	-.063768
18	.000080	-.530512	-.064192
19	.000000	-.538216	-.064563
20	.000000	-.545996	-.064906
21	.000000	-.554587	-.065249
22	.000000	-.563884	-.065588
23	.000000	-.574025	-.065917
24	.000000	-.584966	-.066235
25	.000000	-.596710	-.066540
26	.000000	-.609270	-.066833
27	.000000	-.622687	-.067118
28	.000000	-.637073	-.067405
29	.000000	-.652710	-.067707
30	.000000	-.670107	-.068037
31	.000000	-.690208	-.068410
32	.000000	-.714853	-.068825
33	.000000	-.747924	-.069224
34	.000000	-.809232	-.069237
35	.000000	-.901684	-.060016
36	.000000	-.969975	-.045902
37	.000000	-1.000616	-.034404
38	.000000	-1.018204	-.024023
39	.000000	-1.026066	-.014472
40	.000000	-1.027015	-.005897
41	.000000	-1.022184	.003417
42	.000000	-1.011967	.011975
43	.000000	-.996060	.020480
44	.000000	-.972950	.029298
45	.000000	-.940605	.038586
46	.000000	-.897366	.048576
47	-.000000	-.821837	.060879
48	-.000000	-.728541	.068690
49	-.000000	-.662367	.067639
50	-.000000	-.620371	.065702
51	-.000000	-.587077	.063308
52	-.000000	-.559766	.060587
53	-.000000	-.537232	.057609
54	-.000000	-.518893	.054428

55	-.000000	-.504061	.051085
56	-.000000	-.492678	.047614
57	-.000000	-.484340	.044043
58	-.000000	-.478664	.040381
59	-.000000	-.475139	.036622
60	-.000000	-.473194	.032710
61	-.000000	-.472401	.028835
62	-.000000	-.472261	.025176
63	-.000000	-.472519	.021657
64	-.000000	-.472663	.018180
65	-.000000	-.465196	.014764
66	-.000000	-.112396	.263310
67	-.000000	.248480	.158381
68	-.000000	.355761	.120440
69	-.000000	.446998	.100631
70	-.000000	.524931	.088438
71	-.000000	.599446	.080152
72	-.000000	.711113	.072859
73	-.000000	.804160	.044431
74	-.000000	.764041	.043223
75	-.000000	.776427	.039032
76	-.000000	.787249	.054294
77	-.000000	.797507	.029385
78	-.000000	.806824	.023167
79	-.000000	.814952	.015916
80	-.000000	.819135	.005939
81	-.000000	.797824	-.010202
82	-.000000	.614642	-.038232
83	.000000	.173251	-.060285
84	.000000	-.179123	-.058588
85	.000000	-.331736	-.054406
86	.000000	-.403690	-.051500
87	.000000	-.443829	-.049793
88	.000000	-.469444	-.048923
89	.000000	-.488046	-.048594
90	.000000	-.503478	-.048612
91	.000000	-.520871	-.048755
92	.000000	-.546986	-.047101
93	.000000	-.562363	-.036748
94	-.000000	-.491496	-.048811
95	.000000	-.476729	-.061849
96	.000000	-.471633	-.078508
97	.000000	-.477130	-.102257
98	.000000	-.489648	-.143734
99	-.000000	-.736559	-.252188
100	.000000	.449277	.019959
101	.000000	.467179	.023335
102	.000000	.470295	.026875
103	.000000	.471039	.030559
104	.000000	.471012	.034490
105	.000000	.470339	.038641
106	.000000	.468566	.042902
107	.000000	.465427	.046870
108	.000000	.460491	.050611
109	.000000	.453642	.054192
110	.000000	.444990	.057671
111	.000000	.434892	.061095

112	.000000	.423794	.064492
113	-.000000	.412160	.067878
114	-.000000	.400335	.071266
115	-.000000	.388493	.074670
116	-.000000	.376606	.078151
117	-.000000	.363709	.081886
118	-.000000	.349296	.087252
119	-.000000	.337615	.092876
120	-.000000	.330334	.097848
121	-.000000	.323862	.102628
122	-.000000	.318355	.107278
123	-.000000	.313878	.111812
124	-.000000	.309377	.116230
125	-.000000	.305581	.120600
126	-.000000	.302302	.124887
127	-.000000	.299391	.129108
128	-.000000	.296780	.133268
129	-.000000	.294526	.137360
130	-.000000	.292521	.141395
131	-.000000	.290790	.145366
132	-.000000	.289323	.149272
133	-.000000	.288102	.153116
134	-.000001	.287111	.156894
135	-.000000	.286375	.160602
136	-.000000	.285877	.164241
137	-.000000	.285576	.167810
138	-.000000	.285540	.171305
139	-.000000	.285714	.174724
140	-.000001	.286162	.178061
141	-.000000	.286895	.181317
142	-.000001	.287791	.184485
143	-.000000	.288999	.187562
144	-.000000	.290471	.190545
145	-.000000	.292236	.193431
146	-.000000	.294283	.196213
147	-.000000	.296620	.198879
148	-.000000	.299221	.201400
149	-.000000	.302036	.203773
150	-.000000	.305107	.206024
151	-.000000	.308398	.208146
152	-.000000	.311861	.210129
153	-.000000	.315483	.211974
154	-.000000	.319122	.213673
155	-.000000	.322832	.215219
156	-.000001	.326417	.216614
157	-.000001	.329880	.217849
158	-.000000	.333131	.218930
159	-.000000	.336185	.219855
160	-.000000	.338739	.220625
161	-.000000	.341053	.221245
162	-.000000	.343072	.221711
163	-.000000	.344769	.222029
164	-.000001	.346217	.222193
165	-.000000	.347469	.222208
166	-.000000	.348560	.222061
167	-.000001	.349661	.221745
168	-.000001	.350709	.221259

169	-.000001	.351871	.220577
170	-.000000	.353192	.219689
171	-.000000	.354749	.218572
172	-.000001	.356616	.217199
173	-.000001	.358934	.215527
174	-.000001	.361838	.213497
175	-.000001	.365566	.211024
176	-.000001	.371682	.207957
177	-.000000	.378320	.203194
178	-.000001	.387666	.196169
179	-.000001	.396761	.190647
180	-.000001	.401570	.186400
181	-.000001	.405105	.182844
182	-.000001	.407744	.179767
183	-.000001	.409806	.177055
184	-.000001	.411528	.174634
185	-.000001	.413102	.172451
186	-.000001	.414679	.170466
187	-.000001	.416363	.168645
188	-.000001	.418207	.166960
189	-.000001	.420247	.165388
190	-.000001	.422467	.163908
191	-.000001	.424854	.162506
192	-.000001	.427340	.161168
193	-.000001	.429907	.159882
194	-.000001	.432774	.158523
195	-.000001	.436194	.156956
196	-.000001	.440172	.155155
197	-.000001	.444651	.153097
198	-.000001	.449413	.150761
199	-.000001	.454221	.148140
200	-.000001	.458760	.145237
201	-.000001	.462740	.142074
202	-.000001	.465925	.138689
203	-.000001	.468243	.135133
204	-.000001	.469797	.131473
205	-.000001	.470841	.127791
206	-.000001	.471906	.124237
207	-.000001	.474225	.121366
208	-.000001	.481035	.119695
209	-.000001	.502040	.119699
210	-.000001	.568037	.122161
211	-.000001	.808716	.127634

INDIVIDUAL FLOW NUMBER 3

PT.NO.	VN	VT	SIGMA
1	-.000001	1.181783	-.187672
2	-.000001	.974528	-.201442
3	-.000001	.918760	-.203768
4	-.000000	.901838	-.199696
5	-.000001	.896534	-.191191
6	-.000000	.894965	-.179444
7	-.000000	.895086	-.166596
8	-.000000	.896382	-.154769
9	-.000000	.899128	-.144292
10	-.000000	.903702	-.135147
11	-.000000	.910883	-.127187
12	-.000000	.919530	-.120245
13	-.000000	.930923	-.114189
14	-.000000	.944285	-.108901
15	-.000000	.959170	-.104304
16	-.000000	.974871	-.100320
17	-.000000	.990778	-.096897
18	-.000000	1.006276	-.093973
19	-.000000	1.020902	-.091483
20	-.000000	1.035664	-.089178
21	-.000000	1.051889	-.086836
22	-.000000	1.069599	-.084451
23	-.000000	1.088891	-.082111
24	-.000000	1.109610	-.079517
25	-.000000	1.131889	-.076938
26	-.000000	1.155710	-.074242
27	.000000	1.181158	-.071379
28	-.000000	1.208423	-.068278
29	-.000000	1.238071	-.064839
30	-.000000	1.271043	-.060908
31	-.000000	1.309157	-.056302
32	-.000000	1.355936	-.050727
33	-.000000	1.418658	-.043713
34	-.000000	1.522506	-.034289
35	-.000000	1.707970	-.022810
36	-.000000	1.836622	-.015827
37	-.000000	1.897693	-.011064
38	-.000000	1.931045	-.007013
39	-.000000	1.945970	-.003203
40	-.000000	1.947758	.000591
41	-.000000	1.938600	.004856
42	-.000000	1.919236	.008390
43	-.000000	1.889065	.012539
44	-.000000	1.845190	.017239
45	-.000000	1.783809	.022877
46	-.000000	1.698285	.030128
47	-.000000	1.556255	.041173
48	-.000000	1.370364	.057560
49	-.000000	1.256161	.071917
50	-.000000	1.176700	.085296
51	-.000000	1.113612	.098279
52	-.000000	1.061852	.111046
53	-.000000	1.019144	.123659
54	-.000000	.984177	.136137

55	-.000000	.956183	.148478
56	-.000000	.934630	.160676
57	-.000000	.918951	.172677
58	-.000000	.908412	.184413
59	-.000000	.902257	.195702
60	-.000000	.899980	.206153
61	-.000000	.902068	.214382
62	-.000000	.911281	.219114
63	-.000000	.937200	.219660
64	-.000000	1.010611	.214663
65	-.000000	1.258577	.199943
66	-.000000	.213285	-.499299
67	.000000	-.470938	-.300368
68	.000000	-.674819	-.228420
69	.000000	-.847442	-.190859
70	.000000	-.995262	-.167738
71	.000000	-1.136642	-.152027
72	.000000	-1.348425	-.138189
73	.000000	-1.524895	-.084279
74	.000000	-1.448842	-.081992
75	.000000	-1.472382	-.074047
76	.000000	-1.492950	-.065061
77	-.000000	-1.512476	-.055183
78	.000000	-1.530336	-.043968
79	.000000	-1.545854	-.030211
80	.000000	-1.553854	-.011283
81	-.000000	-1.513480	.019340
82	-.000000	-1.166047	.072513
83	.000000	-.328796	.114355
84	-.000000	.339427	.111140
85	-.000000	.629128	.103209
86	-.000000	.765635	.097696
87	-.000000	.841819	.094459
88	-.000000	.890421	.092790
89	-.000000	.925637	.092159
90	-.000000	.954835	.092194
91	-.000000	.987779	.092461
92	-.000000	1.037211	.089316
93	-.000000	1.066236	.069691
94	-.000000	.931860	.092589
95	-.000000	.903910	.117319
96	-.000000	.894191	.148911
97	-.000000	.904692	.193948
98	-.000001	.928529	.272598
99	-.000001	1.396762	.478195
100	-.000000	-.622165	-.049791
101	-.000000	-.789117	-.038423
102	-.000000	-.852048	-.028415
103	-.000000	-.877386	-.020000
104	-.000000	-.887057	-.012749
105	-.000000	-.889664	-.006066
106	-.000000	-.887853	-.000005
107	-.000000	-.882381	.005250
108	-.000000	-.873249	.009934
109	-.000000	-.860336	.014131
110	.000000	-.843915	.017857
111	.000000	-.824723	.021113

112	.000000	-.803640	.023915
113	.000000	-.781521	.026297
114	-.000000	-.759041	.028311
115	-.000000	-.736521	.030024
116	-.000000	-.714057	.031494
117	-.000000	-.688726	.032848
118	-.000000	-.662518	.032934
119	-.000000	-.641702	.031388
120	-.000000	-.626592	.029620
121	.000000	-.614217	.027806
122	-.000000	-.603590	.025967
123	-.000000	-.594498	.024084
124	-.000000	-.586476	.022236
125	-.000000	-.579847	.020366
126	-.000000	-.573886	.018496
127	-.000000	-.567526	.016627
128	-.000000	-.562616	.014757
129	-.000000	-.558291	.012895
130	-.000000	-.554503	.011034
131	-.000000	-.551290	.009176
132	.000000	-.548460	.007323
133	.000000	-.546138	.005473
134	.000000	-.544279	.003626
135	.000000	-.542850	.001785
136	.000000	-.541867	-.000050
137	.000000	-.541317	-.001882
138	.000000	-.541214	-.003708
139	.000000	-.541570	-.005530
140	.000000	-.542391	-.007343
141	.000000	-.543682	-.009148
142	.000000	-.545465	-.010943
143	.000000	-.547759	-.012726
144	.000000	-.550565	-.014491
145	.000000	-.553905	-.016240
146	.000000	-.557795	-.017966
147	.000000	-.562236	-.019662
148	.000000	-.567129	-.021303
149	.000000	-.572463	-.022894
150	.000000	-.578294	-.024435
151	.000000	-.584590	-.025929
152	.000000	-.591140	-.027364
153	.000000	-.597983	-.028736
154	.000000	-.604954	-.030846
155	.000000	-.611942	-.031284
156	.000000	-.618771	-.032459
157	.000000	-.625355	-.033570
158	.000000	-.631482	-.034623
159	.000000	-.637110	-.035631
160	.000000	-.642145	-.036599
161	.000000	-.646566	-.037546
162	.000000	-.650387	-.038478
163	.000000	-.653608	-.039410
164	.000000	-.656341	-.040352
165	.000000	-.658710	-.041308
166	.000000	-.660833	-.042288
167	.000000	-.662848	-.043284
168	.000000	-.664858	-.044311

169	.000000	-.667070	-.045354
170	.000000	-.669574	-.046412
171	.000000	-.672534	-.047480
172	.000000	-.676070	-.048553
173	.000000	-.680415	-.049619
174	.000000	-.685842	-.050665
175	.000000	-.692760	-.051667
176	.000000	-.702235	-.052598
177	.000000	-.717047	-.053170
178	.000000	-.736855	-.051909
179	.000000	-.752776	-.049732
180	.000000	-.762077	-.047911
181	.000000	-.768762	-.046322
182	.000000	-.773719	-.044917
183	.000000	-.777605	-.043673
184	.000000	-.780853	-.042567
185	.000000	-.783828	-.041581
186	.000000	-.786819	-.040694
187	.000000	-.789999	-.039884
188	.000000	-.793501	-.039135
189	.000000	-.797385	-.038422
190	.000000	-.801591	-.037729
191	.000000	-.806081	-.037049
192	.000000	-.810823	-.036371
193	.000000	-.815718	-.035682
194	.000000	-.821147	-.034915
195	.000000	-.827656	-.033978
196	.000000	-.835220	-.032819
197	.000000	-.843698	-.031392
198	.000000	-.852744	-.029649
199	.000000	-.861858	-.027552
200	.000000	-.870412	-.025071
201	.000000	-.877886	-.022192
202	.000000	-.883862	-.018909
203	-.000000	-.888136	-.015213
204	.000000	-.890803	-.011055
205	.000000	-.892196	-.006326
206	-.000000	-.892416	-.000792
207	-.000000	-.891031	.005419
208	-.000000	-.885864	.012272
209	-.000000	-.869864	.020405
210	-.000000	-.821758	.030555
211	-.000000	-.674881	.042807

INDIVIDUAL FLOW NUMBER 4

PT.NO.	VN	VT	SIGMA
1	-.000000	.061620	-.013095
2	-.000000	.039010	-.013617
3	-.000000	.032766	-.014641
4	-.000000	.030648	-.016221
5	-.000000	.029560	-.018850
6	-.000000	.027924	-.021551
7	-.000000	.023908	-.025547
8	-.000000	.015510	-.030180
9	-.000000	.000354	-.035870
10	-.000000	-.023722	-.040952
11	-.000000	-.058053	-.046679
12	-.000000	-.102705	-.052256
13	-.000000	-.156342	-.057404
14	-.000000	-.216523	-.061900
15	-.000000	-.280033	-.065613
16	-.000000	-.343874	-.068517
17	-.000000	-.405367	-.070663
18	-.000000	-.462542	-.072163
19	-.000000	-.514259	-.073132
20	-.000000	-.564177	-.073737
21	-.000000	-.616692	-.074054
22	-.000000	-.671207	-.074025
23	-.000000	-.727279	-.073620
24	-.000000	-.784144	-.072814
25	-.000000	-.841032	-.071596
26	-.000000	-.897100	-.069971
27	-.000000	-.951532	-.067962
28	-.000000	-1.003650	-.065619
29	-.000000	-1.052941	-.063010
30	-.000000	-1.099325	-.060227
31	-.000000	-1.143422	-.057383
32	-.000000	-1.186959	-.054617
33	.000000	-1.234281	-.052111
34	.000000	-1.315299	-.050030
35	-.000000	-1.422903	-.035757
36	-.000000	-1.465075	-.012003
37	-.000000	-1.423365	.008671
38	-.000000	-1.336928	.026407
39	-.000000	-1.220306	.040945
40	-.000000	-1.085419	.052424
41	-.000000	-.943196	.061013
42	-.000000	-.805416	.066973
43	-.000000	-.677185	.070926
44	-.000000	-.558916	.073524
45	-.000001	-.453013	.075309
46	-.000001	-.362277	.076918
47	-.000000	-.276053	.080784
48	-.000000	-.202970	.085222
49	-.000000	-.153689	.084360
50	-.000000	-.114984	.081643
51	-.000000	-.082772	.077829
52	-.000000	-.055797	.073214
53	-.000000	-.033300	.067992
54	-.000000	-.014968	.062330

55	-.000000	-.000495	.056400
56	-.000000	.010447	.050394
57	-.000000	.018371	.044463
58	-.000000	.023729	.038760
59	-.000000	.027088	.033415
60	-.000000	.029072	.028530
61	-.000000	.030321	.024452
62	-.000000	.031816	.021544
63	-.000000	.035068	.019046
64	-.000000	.043571	.017413
65	-.000000	.071391	.016335
66	.000000	2.731107	-1.542334
67	.000000	1.369045	-.865141
68	-.000000	1.347339	-.587330
69	.000000	1.348747	-.426367
70	.000000	1.395435	-.318086
71	.000000	1.480980	-.239066
72	.000000	1.708175	-.173681
73	.000000	1.750619	-.144001
74	.000000	1.745652	-.125555
75	.000000	1.771940	-.098494
76	.000000	1.829568	-.069084
77	.000000	1.917520	-.037576
78	.000000	2.051739	-.003425
79	.000000	2.267590	.032189
80	.000000	2.642706	.068209
81	.000000	3.339180	.092264
82	.000000	4.246910	.059102
83	.000000	4.280816	-.051104
84	.000000	3.415442	-.086244
85	.000000	2.736011	-.063333
86	.000000	2.371525	-.027844
87	.000000	2.165286	.007970
88	.000000	2.041842	.041709
89	-.000000	1.965494	.073304
90	-.000000	1.922275	.102842
91	-.000000	1.913843	.130018
92	-.000000	1.949518	.148465
93	-.000000	1.924242	.172051
94	-.000000	1.696154	.237772
95	-.000000	1.628807	.316637
96	-.000000	1.606645	.424360
97	-.000000	1.633858	.584330
98	-.000000	1.684398	.860867
99	.000001	3.058841	1.537064
100	-.000000	.007923	-.010933
101	-.000000	-.016447	-.010474
102	-.000000	-.024744	-.010292
103	-.000000	-.028094	-.010424
104	-.000000	-.029658	-.010869
105	-.000000	-.030888	-.011649
106	-.000000	-.032697	-.012725
107	-.000000	-.035626	-.013955
108	-.000000	-.040060	-.015272
109	-.000000	-.046066	-.016607
110	-.000000	-.053405	-.017899
111	-.000000	-.061603	-.019101

112	.000000	-.070068	-.020190
113	.000000	-.078248	-.021167
114	.000000	-.085781	-.022046
115	-.000000	-.092078	-.022839
116	-.000000	-.097321	-.023624
117	-.000000	-.100827	-.024887
118	-.000000	-.103220	-.025713
119	-.000000	-.106103	-.027085
120	.000000	-.110308	-.028200
121	-.000000	-.114892	-.029315
122	-.000000	-.119957	-.030412
123	.000000	-.125471	-.031496
124	.000000	-.131438	-.032564
125	.000000	-.137878	-.033646
126	.000000	-.144924	-.034726
127	.000000	-.152831	-.035815
128	.000000	-.160506	-.036913
129	.000000	-.169202	-.038018
130	.000000	-.178541	-.039134
131	.000000	-.188586	-.040261
132	.000000	-.199386	-.041397
133	.000000	-.211001	-.042543
134	.000000	-.223495	-.043698
135	.000000	-.236947	-.044858
136	.000000	-.251427	-.046023
137	.000000	-.267800	-.047188
138	.000000	-.283778	-.048351
139	.000000	-.301838	-.049506
140	.000000	-.321291	-.050643
141	.000000	-.342219	-.051759
142	.000000	-.364729	-.052840
143	.000000	-.388915	-.053876
144	.000000	-.414862	-.054851
145	.000000	-.442648	-.055751
146	.000000	-.472339	-.056654
147	.000000	-.503945	-.057232
148	.000000	-.536955	-.057763
149	.000000	-.571282	-.058132
150	.000000	-.607227	-.058305
151	.000000	-.644535	-.058261
152	.000000	-.682928	-.057974
153	.000000	-.722014	-.057419
154	.000000	-.761325	-.056586
155	.000000	-.800353	-.055454
156	.000000	-.838464	-.054030
157	.000000	-.875111	-.052324
158	.000000	-.909735	-.050356
159	.000000	-.941861	-.048163
160	.000000	-.971134	-.045776
161	.000000	-.997345	-.043298
162	-.000000	-1.020376	-.040617
163	-.000000	-1.040299	-.037934
164	.000000	-1.057283	-.035230
165	.000000	-1.071603	-.032542
166	.000000	-1.083601	-.029899
167	.000000	-1.093606	-.027300
168	-.000000	-1.101929	-.024774

169	.000000	-1.108937	-.022310
170	.000000	-1.114867	-.019910
171	.000000	-1.119982	-.017571
172	.000000	-1.124512	-.015292
173	-.000000	-1.128676	-.013068
174	-.000000	-1.132799	-.010899
175	-.000000	-1.137289	-.008788
176	-.000000	-1.143048	-.006763
177	-.000000	-1.155495	-.004840
178	-.000000	-1.172057	-.000685
179	-.000000	-1.176839	.005667
180	-.000000	-1.165659	.011774
181	-.000000	-1.145497	.017606
182	-.000000	-1.117286	.023103
183	-.000000	-1.082096	.028208
184	-.000000	-1.040847	.032868
185	-.000000	-.994573	.037033
186	-.000000	-.944423	.040670
187	-.000000	-.891596	.043760
188	-.000000	-.836943	.046299
189	-.000000	-.781797	.048303
190	-.000000	-.727008	.049799
191	-.000000	-.673391	.050825
192	-.000000	-.621638	.051423
193	-.000000	-.572251	.051661
194	-.000000	-.520977	.051598
195	-.000000	-.464188	.050986
196	-.000000	-.403309	.049859
197	-.000000	-.339415	.048046
198	-.000000	-.276083	.045491
199	-.000000	-.216023	.042203
200	-.000000	-.162464	.038303
201	-.000000	-.117858	.033996
202	-.000000	-.083554	.029549
203	-.000000	-.059488	.025237
204	-.000000	-.044338	.021295
205	-.000000	-.035945	.017877
206	-.000000	-.031925	.015054
207	-.000000	-.030279	.012994
208	-.000000	-.029184	.011642
209	-.000000	-.027075	.010829
210	-.000000	-.020947	.010503
211	-.000000	.000238	.010624

INDIVIDUAL FLOW NUMBER 5

PT.NO.	VN	VT	SIGMA
1	-.000000	-.195852	-.057685
2	-.000000	-.343500	-.050286
3	-.000000	-.388880	-.044972
4	-.000000	-.403790	-.041649
5	-.000000	-.408676	-.039752
6	-.000000	-.410334	-.038866
7	-.000000	-.411163	-.038780
8	-.000000	-.412060	-.039228
9	-.000000	-.413989	-.040009
10	-.000000	-.415681	-.041001
11	-.000000	-.418839	-.042121
12	-.000000	-.423064	-.043501
13	-.000000	-.428326	-.044484
14	-.000000	-.434482	-.045624
15	-.000000	-.441358	-.046688
16	-.000000	-.448600	-.047648
17	-.000000	-.455915	-.048495
18	-.000000	-.463043	-.049226
19	-.000000	-.469773	-.049849
20	-.000000	-.476569	-.050423
21	-.000000	-.484037	-.050997
22	-.000000	-.492191	-.051569
23	-.000000	-.501074	-.052134
24	-.000000	-.510623	-.052690
25	-.000000	-.520872	-.053237
26	-.000000	-.531834	-.053775
27	-.000000	-.543588	-.054307
28	-.000000	-.556099	-.054843
29	-.000000	-.569746	-.055394
30	-.000000	-.584930	-.055973
31	-.000000	-.602472	-.056593
32	-.000000	-.624009	-.057257
33	-.000000	-.652873	-.057915
34	-.000000	-.706434	-.058267
35	-.000000	-.787027	-.050587
36	-.000000	-.846589	-.038486
37	-.000000	-.873396	-.028570
38	-.000000	-.888756	-.019587
39	-.000000	-.895647	-.011300
40	-.000000	-.896464	-.003405
41	-.000000	-.892249	.004282
42	-.000000	-.883337	.011763
43	-.000000	-.869461	.019216
44	-.000000	-.849303	.026964
45	-.000000	-.821084	.035153
46	-.000000	-.783390	.043993
47	-.000000	-.717909	.054951
48	-.000000	-.635848	.062135
49	-.000000	-.578148	.061588
50	-.000000	-.541481	.060503
51	-.000000	-.512909	.058684
52	-.000000	-.488555	.056866
53	-.000000	-.468860	.054933
54	-.000000	-.452731	.052958

55	-.000000	-.439801	.051011
56	-.000000	-.429789	.049165
57	-.000000	-.422395	.047509
58	-.000000	-.417184	.046133
59	-.000080	-.413637	.045170
60	-.000000	-.410739	.044809
61	-.000000	-.406854	.045335
62	-.000000	-.398447	.046962
63	-.000000	-.377482	.049953
64	-.000000	-.323760	.054581
65	-.000000	-.169261	.060684
66	.000000	-.098132	.229816
67	.000000	.216857	.138257
68	.000000	.380526	.105136
69	-.000000	.390151	.087843
70	-.000000	.458187	.077199
71	-.000000	.523232	.069966
72	-.000000	.620699	.063698
73	.000000	.701924	.038785
74	.000000	.666923	.037733
75	.000000	.677753	.034074
76	.000000	.687196	.029936
77	.000000	.696138	.025890
78	.000000	.704285	.020228
79	.000000	.711401	.013899
80	.000000	.715073	.005191
81	.000000	.696501	-.008900
82	.000000	.536659	-.033369
83	.000000	.151365	-.052629
84	.000000	-.156202	-.051153
85	.000000	-.289566	-.047503
86	.000000	-.352420	-.044966
87	.000000	-.387519	-.043475
88	.000000	-.409831	-.042699
89	.000000	-.425984	-.042407
90	.000000	-.439420	-.042425
91	-.000000	-.454574	-.042549
92	.000000	-.477381	-.041100
93	-.000000	-.490669	-.032072
94	.000000	-.428933	-.042610
95	.000000	-.416085	-.053994
96	.000000	-.411642	-.068536
97	.000000	-.416485	-.089268
98	.000000	-.427478	-.125470
99	.000000	-.642840	-.220095
100	-.000000	.709329	-.148331
101	-.000000	.511375	-.156733
102	-.000000	.449204	-.157199
103	-.000000	.426131	-.153127
104	-.000000	.416958	-.145652
105	-.000000	.412742	-.135229
106	-.000000	.409844	-.123333
107	-.000000	.406596	-.111892
108	-.000000	.402072	-.101355
109	-.000000	.395969	-.091868
110	-.000000	.388350	-.083446
111	-.000000	.379516	-.076029

112	-.000000	.369835	-.069535
113	-.000000	.359675	-.063877
114	-.000000	.349354	-.058973
115	-.000000	.339049	-.054768
116	.000000	.328730	-.051212
117	.000000	.320103	-.048333
118	.000000	.304927	-.048254
119	.000000	.295079	-.049938
120	.000000	.288184	-.051130
121	.000000	.282557	-.052106
122	.000000	.277676	-.052925
123	.000000	.273566	-.053648
124	.000000	.269869	-.054179
125	.000000	.266619	-.054660
126	.000000	.263763	-.055048
127	.000000	.261207	-.055353
128	.000000	.258968	-.055600
129	.000000	.256988	-.055772
130	.000000	.255271	-.055894
131	.000000	.253782	-.055965
132	.000000	.252513	-.055987
133	.000000	.251448	-.055964
134	.000000	.250593	-.055906
135	.000000	.249928	-.055806
136	.000000	.249484	-.055667
137	.000000	.249240	-.055506
138	.000000	.249190	-.055311
139	.000000	.249363	-.055099
140	.000000	.249732	-.054857
141	.000000	.250350	-.054598
142	.000000	.251164	-.054324
143	.000000	.252233	-.054139
144	.000000	.253550	-.053743
145	.000000	.255066	-.053440
146	.000000	.256853	-.053140
147	.000000	.258910	-.052840
148	.000000	.261151	-.052542
149	.000000	.263614	-.052265
150	.000000	.266292	-.052001
151	.000000	.269161	-.051758
152	.000000	.272204	-.051537
153	.000000	.275354	-.051381
154	.000000	.278565	-.051184
155	.000000	.281753	-.051047
156	.000000	.284896	-.050953
157	.000000	.287917	-.050889
158	.000000	.290745	-.050854
159	.000000	.293351	-.050853
160	.000000	.295676	-.050876
161	.000000	.297692	-.050917
162	.000000	.299435	-.050972
163	.000000	.300922	-.051040
164	.000000	.302195	-.051115
165	.000000	.303271	-.051195
166	.000000	.304252	-.051282
167	.000000	.305170	-.051358
168	.000000	.306111	-.051446

169	.000000	.307117	-.051532
170	.000000	.308256	-.051616
171	.000000	.309584	-.051699
172	.000000	.311177	-.051782
173	.000000	.313125	-.051854
174	.000000	.315562	-.051908
175	.000000	.318607	-.051931
176	.000000	.322166	-.051893
177	.000000	.330007	-.051525
178	.000000	.343187	-.048361
179	.000000	.346802	-.043436
180	.000000	.351112	-.038975
181	.000000	.354112	-.034775
182	.000000	.356336	-.030755
183	.000000	.358079	-.026869
184	-.000000	.359539	-.023092
185	-.000000	.360884	-.019404
186	-.000000	.362237	-.015799
187	-.000000	.363673	-.012266
188	-.000000	.365254	-.008805
189	-.000000	.367007	-.005413
190	-.000000	.368951	-.002088
191	-.000000	.370997	.001163
192	-.000000	.373173	.004351
193	-.000000	.375425	.007466
194	-.000000	.377909	.010825
195	-.000000	.380882	.014759
196	-.000000	.384343	.019352
197	-.000000	.388218	.024699
198	-.000000	.392367	.030898
199	-.000000	.396519	.038077
200	-.000000	.400431	.046371
201	-.000000	.403847	.055928
202	-.000000	.406578	.066894
203	-.000000	.408565	.079379
204	-.000000	.409887	.093416
205	-.000000	.410755	.108861
206	-.000000	.411635	.125200
207	-.000000	.413396	.139995
208	-.000001	.418414	.151180
209	-.000000	.433943	.158074
210	-.000000	.484363	.159377
211	-.000000	.665693	.151122

DOUGLAS AIRCRAFT COMPANY TWO-DIMENSIONAL POTENTIAL FLOW PROGRAM

COMBINED FLOW NOZZLE WITH CENTERBODY

ALPHA = .000000 ALPHA 0 = 130.200014 NO. OF BODIES 3

CL = 373.425842 CHORD = 1.000000 TOTAL ELEMENTS 211

BODY ID = 1 NOZZLE WITH CENTERBO NO. OF ELEMENTS 65

I	X	Y	S	VT	CP	J	SIGMA	VN
1	-42.866555	.000000	.023011	.173010	.970068	1	-.011949	.000000
2	-38.599660	.000000	.069035	.074115	.994507	2	.000114	.000000
3	-34.332765	.000000	.115058	.040674	.998346	3	.008522	-.000000
4	-30.065875	.000000	.161081	.029341	.999139	4	.014125	.000000
5	-25.798980	.000000	.207104	.025627	.999343	5	.017838	.000000
6	-21.532085	.000000	.253127	.024463	.999402	6	.020349	.000000
7	-17.615420	.000000	.295372	.024146	.999417	7	.022008	.000000
8	-14.346185	.000000	.330634	.024112	.999419	8	.023128	.000000
9	-11.621820	.000000	.360020	.024215	.999414	9	.023979	.000000
10	-9.351515	.000000	.384507	.024466	.999401	10	.024682	-.000000
11	-7.459595	.000000	.404914	.024865	.999382	11	.025292	-.000000
12	-5.882995	.000000	.421919	.025413	.999354	12	.025826	-.000000
13	-4.569165	.000000	.436090	.026084	.999320	13	.026295	-.000000
14	-3.474305	.000000	.447899	.026862	.999278	14	.026700	-.000000
15	-2.561920	.000000	.457740	.027698	.999233	15	.027048	-.000000
16	-1.801600	.000000	.465941	.028583	.999183	16	.027337	.000000
17	-1.168000	.000000	.472775	.029449	.999133	17	.027576	-.000000
18	-.640000	.000000	.478470	.030277	.999083	18	.027770	-.000000
19	-.200000	.000000	.483216	.031043	.999036	19	.027926	-.000000
20	.200000	.000000	.487530	.031802	.998989	20	.028062	-.000000
21	.600000	.000000	.491845	.032617	.998936	21	.028189	-.000000
22	1.000000	.000000	.496159	.033480	.998879	22	.028307	-.000000
23	1.400000	.000000	.500474	.034396	.998817	23	.028414	-.000000
24	1.800000	.000000	.504788	.035383	.998748	24	.028501	-.000000
25	2.200000	.000000	.509103	.036402	.998675	25	.028569	-.000000
26	2.600000	.000000	.513417	.037457	.998597	26	.028612	-.000000
27	3.000000	.000000	.517731	.038544	.998514	27	.028625	-.000000
28	3.400000	.000000	.522046	.039660	.998427	28	.028601	-.000000
29	3.800000	.000000	.526360	.040819	.998334	29	.028530	-.000000
30	4.200000	.000000	.530675	.042048	.998252	30	.028396	-.000000
31	4.600000	.000000	.534989	.043391	.998117	31	.028177	-.000000
32	5.000000	.000000	.539304	.044946	.997980	32	.027833	-.000000
33	5.400000	.000000	.543618	.046987	.997792	33	.027282	-.000000
34	5.800000	.000000	.547932	.052655	.997227	34	.026271	-.000000
35	6.183177	-.007327	.552067	.056473	.996811	35	.021768	.000000
36	6.548490	-.050566	.556038	.060319	.996362	36	.016035	.000000
37	6.908226	-.140784	.560041	.061191	.996256	37	.011421	.000000
38	7.252367	-.273904	.564023	.061444	.996225	38	.007263	.000000
39	7.574562	-.446416	.567968	.060979	.996282	39	.003413	.000000
40	7.873128	-.656621	.571908	.060059	.996393	40	-.000284	.000000
41	8.143575	-.901168	.575844	.058761	.996547	41	-.003905	.000000

DOUGLAS AIRCRAFT COMPANY TWO-DIMENSIONAL POTENTIAL FLOW PROGRAM

COMBINED FLOW NOZZLE WITH CENTERBODY

ALPHA = .000000 ALPHA 0 = 130.200014 NO. OF BODIES 3

CL = 373.425842 CHORD = 1.000000 TOTAL ELEMENTS 211

BODY ID = 1 NOZZLE WITH CENTERBO NO. OF ELEMENTS 65

I	X	Y	S	VT	CP	J	SIGMA	VN
42	8.377462	-1.170339	.579692	.057226	.996725	42	-.007453	.000000
43	8.574709	-1.460242	.583476	.055476	.996922	43	-.011024	.000000
44	8.738146	-1.774216	.587296	.053441	.997144	44	-.014802	.000000
45	8.864360	-2.108132	.591149	.051052	.997394	45	-.018907	.000000
46	8.950530	-2.457451	.595032	.048818	.997617	46	-.023545	-.000000
47	8.992769	-2.817786	.598948	.044136	.998052	47	-.029682	-.000000
48	9.000000	-3.194700	.603015	.040104	.998392	48	-.035301	-.000000
49	9.000000	-3.623045	.607635	.034748	.998793	49	-.037574	.000000
50	9.000000	-4.137060	.613180	.032307	.998956	50	-.039429	-.000000
51	9.000000	-4.753875	.619833	.030383	.999077	51	-.041139	-.000000
52	9.000000	-5.494055	.627816	.028806	.999170	52	-.042802	-.000000
53	9.000000	-6.382270	.637397	.027507	.999243	53	-.044479	-.000000
54	9.000000	-7.448125	.648893	.026455	.999300	54	-.046221	.000000
55	9.000000	-8.727150	.662689	.025606	.999344	55	-.048076	-.000000
56	9.000000	-10.261985	.679243	.024926	.999379	56	-.050094	-.000000
57	9.000000	-12.103785	.699109	.024407	.999404	57	-.052323	-.000000
58	9.000000	-14.313945	.722948	.023982	.999425	58	-.054825	-.000000
59	9.000000	-16.966140	.751555	.023472	.999449	59	-.057643	-.000000
60	9.000000	-20.148770	.785883	.022609	.999489	60	-.060819	.000000
61	9.000000	-23.696275	.824147	.020449	.999582	61	-.064141	-.000000
62	9.000000	-27.319325	.863225	.014865	.999729	62	-.067354	.000000
63	9.000000	-30.942375	.902304	.000207	1.000000	63	-.070463	.000000
64	9.000000	-34.565425	.941382	-.039942	.998405	64	-.073505	.000000
65	9.000000	-38.188475	.980461	-.173010	.970068	65	-.076042	.000000

INTEGRATED VALUES

CY = -53.80214 CX = -39.86365

CL = -53.80214 CD = -39.86365 CM = -169.67242

PARABOLIC INTEGRATION

INTEGRATED VALUES

CY = -53.79242 CX = -39.85365

CL -53.79242 CD = -39.85365 CM = -169.65198

DOUGLAS AIRCRAFT COMPANY TWO-DIMENSIONAL POTENTIAL FLOW PROGRAM

COMBINED FLOW NOZZLE WITH CENTERBODY

ALPHA = .000000 ALPHA 0 = 130.200014 NO. OF BODIES 3

CL = 373.425842 CHORD = 1.000000 TOTAL ELEMENTS 211

BODY ID = 2 NOZZLE WITH CENTERBO NO. OF ELEMENTS 34

I	X	Y	S	VT	CP	J	SIGMA	VN
1	10.447415	5.889486	.033387	-.014670	.999785	66	-.001891	-.000000
2	9.406835	5.681370	.096259	-.022746	.999483	67	-.001570	-.000000
3	8.512170	5.502435	.150314	-.028029	.999214	68	-.001715	-.000000
4	7.766615	5.353320	.195360	-.032662	.998933	69	-.001907	-.000000
5	7.145320	5.229059	.232899	-.036932	.998636	70	-.002094	.000000
6	6.627570	5.125515	.264181	-.041329	.998292	71	-.002262	.000000
7	6.196115	5.039225	.290250	-.048666	.997682	72	-.002381	.000000
8	5.825264	4.999488	.312452	-.053676	.997119	73	-.001163	-.000000
9	5.476115	5.015349	.333161	-.051591	.997338	74	-.001237	-.000000
10	5.127741	5.043183	.353068	-.052402	.997254	75	-.001225	-.000000
11	4.781150	5.086228	.374562	-.053382	.997150	76	-.001204	-.000000
12	4.437616	5.146273	.395226	-.054538	.997026	77	-.001173	-.000000
13	4.099025	5.226290	.415882	-.056948	.996870	78	-.001123	-.000000
14	3.768555	5.331409	.436395	-.057927	.996644	79	-.001024	-.000000
15	3.458378	5.468221	.456899	-.060883	.996293	80	-.000788	-.000000
16	3.194132	5.645266	.475381	-.064915	.995786	81	-.000153	-.000000
17	3.027862	5.872013	.492219	-.062309	.996118	82	.001501	-.000000
18	3.027862	6.127987	.507786	-.040257	.998379	83	.003417	-.000000
19	3.194132	6.354734	.524619	-.016119	.999740	84	.003584	-.000000
20	3.458378	6.531779	.543506	-.003419	.999988	85	.003200	-.000000
21	3.768555	6.668591	.563805	.002882	.999992	86	.002790	-.000000
22	4.099025	6.773706	.584158	.006416	.999959	87	.002438	.000000
23	4.437616	6.853727	.604774	.008705	.999924	88	.002145	-.000000
24	4.781150	6.913772	.625438	.010233	.999895	89	.001893	-.000000
25	5.127741	6.956817	.646132	.011311	.999872	90	.001674	.000000
26	5.476115	6.984651	.666839	.012243	.999850	91	.001478	-.000000
27	5.825264	7.000512	.687548	.013325	.999822	92	.001259	.000000
28	6.196115	6.960775	.709750	.014212	.999798	93	.000556	-.000000
29	6.627570	6.874485	.735819	.012221	.999851	94	.000685	.000000
30	7.145320	6.770941	.767101	.011972	.999857	95	.000759	.000000
31	7.766615	6.646680	.804640	.011878	.999859	96	.000803	.000000
32	8.512170	6.497565	.849686	.011965	.999857	97	.000816	.000000
33	9.406835	6.318630	.903741	.012250	.999850	98	.000861	.000000
34	10.447415	6.110514	.966613	.014670	.999785	99	.001313	-.000000

INTEGRATED VALUES

CY = -.01276 CX = .00617

CL = -.01276 CD = .00617 CM = .10526

PARABOLIC INTEGRATION

INTEGRATED VALUES

CY = -.01276 CX = -.00299

CL = -.01276 CD = -.00299 CM = .05020

DOUGLAS AIRCRAFT COMPANY TWO-DIMENSIONAL POTENTIAL FLOW PROGRAM

COMBINED FLOW NOZZLE WITH CENTERBODY

ALPHA = .000000 ALPHA 0 = 130.200014 NO. OF BODIES 3

CL = 373.425842 CHORD = 1.000000 TOTAL ELEMENTS 211

BODY ID = 3 NOZZLE WITH CENTERBO NO. OF ELEMENTS 112

I	X	Y	S	VT	CP	J	SIGMA	VN
1	21.000000	-38.239865	.015777	-.190811	.963581	100	.006088	.000000
2	21.000000	-34.719595	.047332	-.083741	.992987	101	-.007361	.000000
3	21.000000	-31.199320	.078886	-.048046	.997692	102	-.010973	.000000
4	21.000000	-27.679045	.110441	-.033454	.998881	103	-.007940	.000000
5	21.000000	-24.158775	.141995	-.027583	.999259	104	.000248	.000000
6	21.000000	-20.615420	.173756	-.025225	.999364	105	.012857	.000000
7	21.000000	-17.346185	.203060	-.024280	.999410	106	.028042	.000000
8	21.000000	-14.621820	.227481	-.023753	.999436	107	.043312	.000000
9	21.000000	-12.351515	.247831	-.023314	.999456	108	.058018	.000000
10	21.000000	-10.459595	.264789	-.022833	.999479	109	.071881	.000000
11	21.000000	-8.882995	.278921	-.022276	.999504	110	.084794	.000000
12	21.000000	-7.569165	.290698	-.021620	.999533	111	.096747	.000000
13	21.000000	-6.474305	.300512	-.020932	.999562	112	.107782	.000000
14	21.000000	-5.561920	.308690	-.020241	.999590	113	.117978	.000000
15	21.000000	-4.801600	.315506	-.019518	.999619	114	.127441	.000000
16	21.000000	-4.168000	.321185	-.018791	.999647	115	.136290	.000000
17	21.000000	-3.640000	.325918	-.018215	.999668	116	.144717	.000000
18	21.000000	-3.200000	.329862	-.013807	.999809	117	.153134	.000000
19	20.998334	-2.803643	.333415	-.017136	.999706	118	.160149	.000000
20	20.988434	-2.410980	.336935	-.018087	.999673	119	.163394	.000000
21	20.967873	-2.018591	.340458	-.016381	.999732	120	.165375	.000000
22	20.937036	-1.626742	.343981	-.015873	.999748	121	.166941	.000000
23	20.895940	-1.235698	.347506	-.015393	.999763	122	.168206	.000000
24	20.844532	-.845729	.351031	-.015194	.999769	123	.169135	.000000
25	20.782883	-.457107	.354559	-.014743	.999783	124	.169938	.000000
26	20.711077	-.070098	.358087	-.014555	.999788	125	.170538	.000000
27	20.629100	.315026	.361616	-.014302	.999795	126	.170958	-.000000
28	20.537014	.698004	.365147	-.014053	.999803	127	.171235	-.000000
29	20.434861	1.078577	.368679	-.013873	.999808	128	.171348	-.000000
30	20.322701	1.456475	.372213	-.013648	.999814	129	.171334	-.000000
31	20.200608	1.831438	.375748	-.013469	.999819	130	.171181	-.000000
32	20.068642	2.203205	.379284	-.013309	.999823	131	.170895	-.000000
33	19.926890	2.571512	.382821	-.013164	.999827	132	.170484	-.000000
34	19.775439	2.936109	.386360	-.013016	.999831	133	.169955	-.000000
35	19.614371	3.296746	.389901	-.012894	.999834	134	.169300	-.000000
36	19.443786	3.653162	.393443	-.012775	.999837	135	.168530	-.000000
37	19.263797	4.005113	.396986	-.012623	.999841	136	.167652	-.000000
38	19.074511	4.352354	.400531	-.012514	.999843	137	.166654	-.000000
39	18.876045	4.694640	.404078	-.012373	.999847	138	.165551	-.000000
40	18.668518	5.031730	.407626	-.012270	.999849	139	.164329	-.000000
41	18.452060	5.363384	.411176	-.012143	.999853	140	.163006	-.000000

DOUGLAS AIRCRAFT COMPANY TWO-DIMENSIONAL POTENTIAL FLOW PROGRAM

COMBINED FLOW NOZZLE WITH CENTERBODY

ALPHA = .000000 ALPHA 0 = 130.200014 NO. OF BODIES 3

CL = 373.425842 CHORD = 1.000000 TOTAL ELEMENTS 211

BODY ID = 3 NOZZLE WITH CENTERBO NO. OF ELEMENTS 112

I	X	Y	S	VT	CP	J	SIGMA	VN
42	18.226807	5.689373	.414728	-.012022	.999856	141	.161575	-.000000
43	17.992898	6.009462	.418282	-.011891	.999859	142	.160036	-.000000
44	17.750473	6.323430	.421838	-.011772	.999861	143	.158392	-.000000
45	17.499686	6.631053	.425395	-.011661	.999864	144	.156645	-.000000
46	17.240695	6.932111	.428955	-.011546	.999867	145	.154800	-.000000
47	16.973653	7.226388	.482517	-.011437	.999869	146	.152847	-.000000
48	16.698725	7.513669	.436082	-.011342	.999871	147	.150792	-.000000
49	16.419785	7.790181	.439602	-.011238	.999874	148	.148677	-.000000
50	16.137308	8.055995	.443079	-.011133	.999876	149	.146493	-.000000
51	15.847870	8.314573	.446558	-.011019	.999879	150	.144215	-.000000
52	15.551661	8.565720	.450039	-.010902	.999881	151	.141845	-.000000
53	15.248876	8.809246	.453522	-.010786	.999884	152	.139382	-.000000
54	14.939716	9.044977	.457007	-.010661	.999886	153	.136837	-.000000
55	14.624385	9.272732	.460494	-.010587	.999888	154	.134192	-.000000
56	14.303094	9.492341	.463983	-.010471	.999890	155	.131473	-.000000
57	13.976052	9.703646	.467473	-.010381	.999892	156	.128659	-.000000
58	13.643479	9.906478	.470965	-.010277	.999894	157	.125760	-.000000
59	13.305597	10.100696	.474458	-.010174	.999897	158	.122781	-.000000
60	12.962623	10.286193	.477953	-.010077	.999898	159	.119712	-.000000
61	12.614799	10.462703	.481450	-.009998	.999900	160	.116558	-.000000
62	12.267355	10.630219	.484948	-.009951	.999901	161	.113323	-.000000
63	11.905525	10.788573	.488447	-.009870	.999903	162	.110009	-.000000
64	11.544550	10.937647	.491948	-.009807	.999904	163	.106609	-.000000
65	11.179669	11.077324	.495450	-.009742	.999905	164	.103126	-.000000
66	10.811134	11.207502	.498954	-.009695	.999906	165	.099561	-.000000
67	10.439187	11.328072	.502459	-.009677	.999906	166	.095904	-.000000
68	10.064085	11.438947	.505965	-.009640	.999907	167	.092178	-.000000
69	9.686084	11.540043	.509472	-.009635	.999907	168	.088356	-.000000
70	9.305431	11.631271	.512981	-.009637	.999907	169	.084445	-.000000
71	8.922394	11.712563	.516491	-.009672	.999906	170	.080447	-.000000
72	8.537231	11.783856	.520002	-.009714	.999906	171	.076357	-.000000
73	8.150198	11.845087	.523515	-.009794	.999904	172	.072162	-.000000
74	7.761560	11.896202	.527028	-.009901	.999902	173	.067860	-.000000
75	7.371582	11.937156	.530543	-.010035	.999899	174	.063438	-.000000
76	6.980534	11.967915	.534059	-.010250	.999895	175	.058875	-.000000
77	6.588685	11.988403	.537577	-.010426	.999891	176	.054131	-.000000
78	6.196292	11.998334	.541095	-.010586	.999888	177	.048950	-.000000
79	5.800000	12.000000	.544647	-.007941	.999937	178	.046447	-.000000
80	5.400000	12.000000	.548233	-.011300	.999872	179	.047478	-.000000
81	5.000000	12.000000	.551818	-.011637	.999865	180	.048765	-.000000
82	4.600000	12.000000	.555404	-.012012	.999856	181	.050177	-.000000

DOUGLAS AIRCRAFT COMPANY TWO-DIMENSIONAL POTENTIAL FLOW PROGRAM

COMBINED FLOW NOZZLE WITH CENTERBODY

ALPHA = .000000 ALPHA 0 = 130.200014 NO. OF BODIES 3

CL = 373.425842 CHORD = 1.000000 TOTAL ELEMENTS 211

BODY ID = 3 NOZZLE WITH CENTERBO NO. OF ELEMENTS 182

I	X	Y	S	VT	CP	J	SIGMA	VN
83	4.200000	12.000000	.558989	-.012386	.999847	182	.151668	-.000000
84	3.800000	12.000000	.562675	-.012771	.999837	183	.053211	-.000000
85	3.400000	12.000000	.566160	-.013178	.999826	184	.054790	-.000000
86	3.000000	12.000000	.569746	-.013606	.999815	185	.056398	-.000000
87	2.600000	12.000000	.573331	-.014072	.999802	186	.058027	-.000000
88	2.200000	12.000000	.576916	-.014563	.999788	187	.059674	-.000000
89	1.800000	12.000000	.580502	-.015064	.999773	188	.061337	-.000000
90	1.400000	12.000000	.584087	-.015574	.999757	189	.063010	-.000000
91	1.000000	12.000000	.587673	-.016077	.999742	190	.064694	-.000000
92	.600000	12.000000	.591258	-.016589	.999726	191	.066382	-.000000
93	.200000	12.000000	.594844	-.017089	.999718	192	.068078	-.000000
94	-.200000	12.000000	.598429	-.017585	.999691	193	.069773	-.000000
95	-.640000	12.000000	.602373	-.018117	.999672	194	.071644	-.000000
96	-1.168000	12.000000	.607106	-.018711	.999680	195	.073891	-.000000
97	-1.801600	12.000000	.612785	-.019375	.999625	196	.076584	-.000000
98	-2.561920	12.000000	.619600	-.020074	.999597	197	.079812	-.000000
99	-3.474305	12.000000	.627779	-.020792	.999568	198	.083660	-.000000
100	-4.569165	12.000000	.637593	-.021505	.999588	199	.088241	-.000000
101	-5.882995	12.000000	.649369	-.022155	.999509	200	.093662	-.000000
102	-7.459595	12.000000	.663501	-.022708	.999484	201	.100040	-.000001
103	-9.351515	12.000000	.680460	-.023113	.999466	202	.107473	-.000001
104	-11.621820	12.000000	.700810	-.023388	.999463	203	.116025	-.000000
105	-14.346185	12.000000	.725230	-.023531	.999446	204	.125680	-.000001
106	-17.615420	12.000000	.754534	-.023488	.999448	205	.136250	-.000001
107	-21.532085	12.000000	.789642	-.023103	.999466	206	.147191	-.000001
108	-25.798980	12.000000	.827889	-.021841	.999523	207	.156540	-.000000
109	-30.065875	12.000000	.866136	-.017981	.999677	208	.162615	-.000001
110	-34.332765	12.000000	.904383	-.005798	.999968	209	.164612	-.000000
111	-38.599660	12.000000	.942629	.034838	.998786	210	.161158	-.000000
112	-42.866555	12.000000	.980876	.190811	.963591	211	.148583	-.000000

INTEGRATED VALUES

CY = 65.82139 CX = 51.82014

CL = 65.82139 CD = 51.82014 CM = 63.23408

PARABOLIC INTEGRATION

INTEGRATED VALUES

CY = 65.80866 CX = 51.81179

CL = 65.80866 CD = 51.81179 CM = 62.98027

TOTAL CM = -106.33309

TOTAL CM = -106.62152 (PARABOLIC)

FLOWS COMPLETE, T = .000SECONDS.

INDIVIDUAL FLOW NO. 1 OFFBODY POINTS NOZZLE WITH CENTERBOD

I	X(I)	Y(I)	VX	VY	VT	THETA(DEG)
1	-1.000000	.400000	.601310	-.006906	.601349	-.657988
2	-1.000000	.989470	.599412	-.016936	.599651	-1.618414
3	-1.000000	1.578950	.595891	-.026651	.596487	-2.560867
4	-1.000000	2.168420	.590728	-.035788	.591811	-3.466922
5	-1.000000	2.757890	.583910	-.044037	.585568	-4.312897
6	-1.000000	3.347370	.575500	-.051032	.577758	-5.067389
7	-1.000000	3.936840	.565691	-.056380	.568490	-5.687648
8	-1.000000	4.526320	.554853	-.059491	.558033	-6.119815
9	-1.000000	5.115790	.543608	-.060146	.546925	-6.313605
10	-1.000000	5.705260	.532791	-.058238	.535964	-6.238114
11	-1.000000	6.294740	.523166	-.054069	.526953	-5.900591
12	-1.000000	6.884210	.515308	-.048296	.517567	-5.354214
13	-1.000000	7.473680	.509428	-.041673	.511130	-4.676583
14	-1.000000	8.063160	.505343	-.034997	.506547	-3.951520
15	-1.000000	8.652630	.502695	-.028450	.503499	-3.239226
16	-1.000000	9.242110	.501075	-.022486	.501579	-2.569507
17	-1.000000	9.831580	.500109	-.017083	.500399	-1.951860
18	-1.000000	10.421050	.499564	-.012063	.499710	-1.383218
19	-1.000000	11.010530	.499272	-.007398	.499527	-.848974
20	-1.000000	11.600000	.499142	-.002960	.499151	-.339725
21	6.000000	.200000	.946952	-.065057	.949184	-3.930716
22	6.000000	.711110	.874860	-.119835	.883029	-7.799611
23	6.000000	1.222220	.821629	-.137451	.833047	-9.497109
24	6.000000	1.733330	.783286	-.138793	.795488	-10.048145
25	6.000000	2.244440	.757098	-.130787	.768305	-9.798062
26	6.000000	2.755560	.741719	-.115819	.750707	-8.875051
27	6.000000	3.266670	.737122	-.094428	.743144	-7.298514
28	6.000000	3.777780	.744753	-.065180	.747599	-5.001722
29	6.000000	4.288890	.769383	-.024227	.769764	-1.803590
30	6.000000	4.800000	.835865	.044135	.837028	3.020452
31	6.000000	7.200000	.609438	-.037780	.610605	-3.543588
32	6.000000	7.711110	.553464	-.034815	.554558	-3.599414
33	6.000000	8.222220	.525109	-.034053	.526212	-3.710391
34	6.000000	8.733330	.505513	-.034229	.506671	-3.873681
35	6.000000	9.244440	.490427	-.033889	.491596	-3.952922
36	6.000000	9.755560	.478095	-.032435	.479192	-3.877546
37	6.000000	10.266670	.467605	-.029428	.468530	-3.601031
38	6.000000	10.777780	.458387	-.024734	.459054	-3.088590
39	6.000000	11.288890	.450116	-.017831	.450469	-2.268536
40	6.000000	11.800000	.442567	-.007416	.442629	-.959964
41	12.000000	-5.000000	.086198	-.596523	.602699	-81.777357
42	12.000000	-3.000000	.165414	-.603641	.625895	-74.675541
43	14.000000	-5.000000	.089798	-.545235	.552580	-80.647583
44	14.000000	-3.000000	.141743	-.527937	.546634	-74.971422
45	16.000000	-5.000000	.075043	-.507323	.512823	-81.585509
46	16.000000	-3.000000	.110802	-.480951	.493559	-77.026744
47	18.000000	-5.000000	.050663	-.485561	.488224	-83.981828
48	18.000000	-3.000000	.076025	-.447354	.453778	-80.355344
49	20.000000	-5.000000	.018509	-.464815	.465183	-87.719661
50	20.000000	-3.000000	.033462	-.420734	.422063	-85.452641

INDIVIDUAL FLOW NO. 2 OFFBODY POINTS NOZZLE WITH CENTERBOD

I	X(I)	Y(I)	VX	VY	VT	THETA(DEG)
1	-1.000000	.400000	-.524496	.006024	.524530	179.341942
2	-1.000000	.989470	-.522840	.014773	.523048	178.381575
3	-1.000000	1.578950	-.519777	.023259	.520297	177.437778
4	-1.000000	2.168420	-.515262	.031229	.516208	176.531672
5	-1.000000	2.757890	-.509316	.038421	.510763	175.686031
6	-1.000000	3.347370	-.501982	.044550	.503955	174.928358
7	-1.000000	3.936840	-.493427	.049179	.495872	174.308189
8	-1.000000	4.526320	-.483977	.051081	.486756	173.874363
9	-1.000000	5.115790	-.474170	.052512	.477069	173.680574
10	-1.000000	5.705260	-.464733	.050848	.467506	173.755941
11	-1.000000	6.294740	-.456346	.047232	.458783	174.090937
12	-1.000000	6.884210	-.449500	.042218	.451478	174.634407
13	-1.000000	7.473680	-.444320	.036894	.445816	175.304640
14	-1.000000	8.063160	-.440714	.030609	.441776	176.026943
15	-1.000000	8.652630	-.438365	.024981	.439077	176.738375
16	-1.000000	9.242110	-.436891	.019757	.437337	177.410727
17	-1.000000	9.831580	-.436027	.015021	.436285	178.026991
18	-1.000000	10.421050	-.435524	.010634	.435654	178.601347
19	-1.000000	11.010530	-.435277	.006532	.435326	179.140301
20	-1.000000	11.600000	-.435147	.002424	.485155	179.654552
21	6.000000	.200000	-.825943	.066669	.827885	176.074995
22	6.000000	.711110	-.763087	.104490	.770208	172.202961
23	6.000000	1.222220	-.716662	.119868	.726617	170.504659
24	6.000000	1.733330	-.688220	.121087	.698860	169.953060
25	6.000000	2.244440	-.660375	.114050	.670151	170.201416
26	6.000000	2.755560	-.646964	.101833	.654805	171.124084
27	6.000000	3.266670	-.642967	.082349	.648219	172.701494
28	6.000000	3.777780	-.649628	.056856	.652111	174.998138
29	6.000000	4.288890	-.671117	.021135	.671450	178.196190
30	6.000000	4.800000	-.729116	-.038470	.730130	-176.979725
31	6.000000	7.200000	-.531673	.033032	.532698	176.444820
32	6.000000	7.711110	-.482786	.030613	.488756	176.371830
33	6.000000	8.222220	-.458012	.030008	.458994	176.251450
34	6.000000	8.733330	-.440875	.030192	.441907	176.082367
35	6.000000	9.244440	-.427684	.029911	.428729	175.999477
36	6.000000	9.755560	-.416898	.028637	.417879	176.074642
37	6.000000	10.266670	-.407697	.026031	.408528	176.346651
38	6.000000	10.777780	-.399611	.021946	.400213	176.856590
39	6.000000	11.288890	-.392329	.016112	.392659	177.648294
40	6.000000	11.800000	-.385484	.007637	.385560	178.865036
41	12.000000	-5.000000	-.075146	.521234	.526633	98.219316
42	12.000000	-3.000000	-.144280	.526443	.545856	105.326459
43	14.000000	-5.000000	-.078218	.475471	.481862	99.341839
44	14.000000	-3.000000	-.123593	.460561	.476663	105.027782
45	16.000000	-5.000000	-.065271	.442350	.447140	98.393661
46	16.000000	-3.000000	-.096498	.419332	.430263	102.960416
47	18.000000	-5.000000	-.045903	.419230	.421493	95.978830
48	18.000000	-3.000000	-.066002	.390237	.395779	99.599730
49	20.000000	-5.000000	-.015989	.405633	.405948	92.257294
50	20.000000	-3.000000	-.028761	.367150	.368275	94.479228

INDIVIDUAL FLOW NO. 3 OFFBODY POINTS NOZZLE WITH CENTERBOD

I	X(I)	Y(I)	VX	VY	VT	THETA(DEG)
1	-1.000000	.000000	.998823	-.011392	.998888	-.656057
2	-1.000000	.989470	.991658	-.027962	.992052	-1.615174
3	-1.000000	1.578950	.985819	-.049048	.986798	-2.558395
4	-1.000000	2.168420	.977221	-.059129	.979009	-3.462618
5	-1.000000	2.757890	.965963	-.072795	.968702	-4.309638
6	-1.000000	3.347370	.952037	-.084349	.955766	-5.063083
7	-1.000000	3.936840	.935855	-.093114	.940876	-5.682022
8	-1.000000	4.526320	.917960	-.098811	.923210	-6.112912
9	-1.000000	5.115790	.899882	-.098392	.904917	-6.305835
10	-1.000000	5.705260	.881532	-.096271	.884773	-6.232521
11	-1.000000	6.294740	.865613	-.089418	.870219	-5.897764
12	-1.000000	6.884210	.852615	-.079903	.856351	-5.353881
13	-1.000000	7.473680	.842833	-.068886	.845651	-4.679207
14	-1.000000	8.063160	.836056	-.057838	.838054	-3.957405
15	-1.000000	8.652630	.831632	-.047154	.832968	-3.245196
16	-1.000000	9.242110	.828909	-.037897	.829748	-2.577004
17	-1.000000	9.831580	.827281	-.028296	.827765	-1.958970
18	-1.000000	10.421050	.826351	-.020015	.826593	-1.387518
19	-1.000000	11.010530	.825872	-.012284	.825963	-.852149
20	-1.000000	11.600000	.825648	-.004922	.825663	-.341542
21	6.000000	.200000	1.566534	-.105988	1.570116	-3.870610
22	6.000000	.711110	1.447106	-.197998	1.460589	-7.791037
23	6.000000	1.222220	1.359190	-.227331	1.378070	-9.495114
24	6.000000	1.733330	1.295760	-.229559	1.315937	-10.046381
25	6.000000	2.244440	1.252396	-.216262	1.270931	-9.797132
26	6.000000	2.755560	1.226896	-.191591	1.241765	-8.875556
27	6.000000	3.266670	1.219263	-.156150	1.229222	-7.298084
28	6.000000	3.777780	1.231839	-.107835	1.236547	-5.001517
29	6.000000	4.288890	1.272555	-.040057	1.278185	-1.802949
30	6.000000	4.800000	1.382574	.072975	1.384498	3.021375
31	6.000000	7.200000	1.008134	-.062501	1.010070	-3.547618
32	6.000000	7.711110	.915494	-.057785	.917316	-3.611641
33	6.000000	8.222220	.868556	-.056562	.870396	-3.725952
34	6.000000	8.733330	.836095	-.056862	.838027	-3.890619
35	6.000000	9.244440	.811089	-.056350	.816044	-3.974203
36	6.000000	9.755560	.790668	-.053876	.792502	-3.898094
37	6.000000	10.266670	.773272	-.048856	.774820	-3.622583
38	6.000080	10.777780	.757993	-.041185	.759111	-3.110032
39	6.000000	11.288890	.744264	-.029884	.744864	-2.299313
40	6.000000	11.800000	.731566	-.013095	.731683	-1.025461
41	12.000000	-5.000000	.142570	-.986758	.997034	-81.778878
42	12.000000	-3.000000	.273724	-.998489	1.035329	-74.669663
43	14.000000	-5.000000	.148360	-.901837	.914028	-80.658711
44	14.000000	-3.000000	.234408	-.873216	.904131	-74.973667
45	16.000000	-5.000000	.123858	-.859060	.868153	-81.602928
46	16.000000	-3.000000	.183102	-.795379	.816183	-77.035971
47	18.000000	-5.000000	.083459	-.795090	.799368	-84.007076
48	18.000000	-3.000000	.125428	-.740029	.750583	-80.380304
49	20.000000	-5.000000	.030909	-.769104	.769709	-87.735832
50	20.000000	-3.000000	.054847	-.696110	.698267	-85.494925

INDIVIDUAL FLOW NO. 4 OFFBODY POINTS NOZZLE WITH CENTERBOD

I	X(I)	Y(I)	VX	VY	VT	THETA(DEG)
1	-1.000000	.400000	-.421815	.042591	.423562	174.228960
2	-1.000000	.989470	-.418358	.105472	.426601	165.685917
3	-1.000000	1.578950	-.398012	.168818	.432139	157.076693
4	-1.000000	2.168420	-.374530	.230971	.440024	148.338053
5	-1.000000	2.757890	-.341785	.292638	.449948	139.429703
6	-1.000000	3.347370	-.298493	.351003	.461447	130.305466
7	-1.000000	3.936840	-.243552	.406888	.473781	120.934614
8	-1.000000	4.526320	-.176712	.452717	.485983	111.322522
9	-1.000000	5.115790	-.099110	.486871	.496856	101.506245
10	-1.000000	5.705260	-.013803	.505117	.506306	91.565322
11	-1.000000	6.294740	.074324	.505048	.510488	81.628258
12	-1.000000	6.884210	.159639	.486647	.512162	71.838522
13	-1.000000	7.473680	.237209	.452297	.510725	62.325070
14	-1.000000	8.063160	.303948	.405746	.506966	53.162766
15	-1.000000	8.652630	.358689	.351059	.501896	44.384117
16	-1.000000	9.242110	.401692	.291658	.496408	35.982430
17	-1.000000	9.831580	.434091	.229969	.491244	27.913391
18	-1.000000	10.421050	.457223	.167438	.486917	20.113021
19	-1.000000	11.010530	.472278	.104849	.483777	12.517078
20	-1.000000	11.600000	.480159	.042821	.482021	5.037048
21	6.000000	.200000	-1.328713	.073301	1.330733	176.842360
22	6.000000	.711110	-1.257271	.114959	1.262516	174.775648
23	6.000000	1.222220	-1.226931	.121755	1.232955	174.333691
24	6.000000	1.733330	-1.225434	.117677	1.231071	174.514776
25	6.000000	2.244440	-1.246323	.108777	1.251061	175.011969
26	6.000000	2.755560	-1.286266	.095720	1.289822	175.744045
27	6.000000	3.266670	-1.344035	.076388	1.346214	176.747105
28	6.000000	3.777780	-1.419854	.046564	1.420617	178.121666
29	6.000000	4.288890	-1.518894	-.000733	1.518894	-179.972340
30	6.000000	4.800000	-1.685270	-.073555	1.686866	-177.507637
31	6.000000	7.200000	1.881176	-.098983	1.883776	-3.010789
32	6.000000	7.711110	1.693624	-.045364	1.691233	-1.537019
33	6.000000	8.222220	1.573943	-.016868	1.574134	-.614005
34	6.000000	8.733330	1.479885	-.005456	1.479895	-.211231
35	6.000000	9.244440	1.400901	-.008962	1.400907	-.162054
36	6.000000	9.755560	1.334630	-.007674	1.334652	-.329459
37	6.000080	10.266670	1.279539	-.013171	1.279606	-.589745
38	6.000000	10.777780	1.234662	-.017920	1.234792	-.831550
39	6.000000	11.288890	1.199185	-.019279	1.199340	-.921059
40	6.000000	11.800000	1.172439	-.012616	1.172507	-.616500
41	12.000000	-5.000000	-.076351	.014838	.077773	169.023811
42	12.000000	-3.000000	-.155820	.021531	.157296	172.143566
43	14.000000	-5.000000	-.075724	-.033138	.082658	-156.365133
44	14.000000	-3.000000	-.124193	-.059296	.137622	-154.477829
45	16.000000	-5.000000	-.056191	-.063824	.085034	-131.360819
46	16.000000	-3.000000	-.079011	-.096921	.125046	-129.187103
47	18.000000	-5.000000	-.032236	-.078733	.085077	-112.265925
48	18.000000	-3.000000	-.038564	-.108100	.114773	-109.633461
49	20.000000	-5.000000	-.009986	-.083314	.083911	-96.834803
50	20.000000	-3.000000	-.008856	-.105479	.106850	-94.799018

INDIVIDUAL FLOW NO. 5 OFFBODY POINTS NOZZLE WITH CENTERBOD

I	X(I)	Y(I)	VX	VY	VT	THETA(DEG)
1	-1.000000	.400000	-.457787	.008264	.457817	179.381282
2	-1.000000	.989470	-.456343	.012938	.456526	178.379734
3	-1.000000	1.578950	-.453651	.020326	.454106	177.434580
4	-1.000000	2.168420	-.449724	.027291	.450552	176.527300
5	-1.000000	2.757890	-.444536	.033581	.445802	175.680042
6	-1.000000	3.347370	-.438105	.038924	.439831	174.922789
7	-1.000000	3.936840	-.430639	.042978	.432778	174.300701
8	-1.000000	4.526320	-.422373	.045596	.424806	173.865511
9	-1.000000	5.115790	-.413811	.045910	.416350	173.669214
10	-1.000000	5.705260	-.405581	.044474	.408013	173.742231
11	-1.000000	6.294740	-.398232	.041528	.400370	174.075697
12	-1.000000	6.884210	-.392227	.036962	.393964	174.617933
13	-1.000000	7.473680	-.387682	.031876	.388990	175.299618
14	-1.000000	8.063160	-.384550	.026691	.385475	176.029579
15	-1.000000	8.652630	-.382523	.021788	.383181	176.745979
16	-1.000000	9.242110	-.381309	.017158	.381695	177.423586
17	-1.000000	9.831580	-.380555	.013013	.380777	178.041491
18	-1.000000	10.421050	-.380178	.009199	.380289	178.613853
19	-1.000000	11.010530	-.379991	.005665	.380033	179.145821
20	-1.000000	11.600000	-.379928	.002272	.379935	179.657417
21	6.000000	.200000	-.720961	.049598	.722659	176.071714
22	6.000000	.711110	-.666093	.091232	.672311	172.200932
23	6.000000	1.222220	-.625565	.104696	.634250	170.503332
24	6.000000	1.733330	-.596373	.105669	.605662	169.952223
25	6.000000	2.244440	-.576419	.099542	.584950	170.202192
26	6.000000	2.755560	-.564710	.088176	.571553	171.125246
27	6.000000	3.266670	-.561229	.071881	.565813	172.701366
28	6.000000	3.777780	-.567041	.049627	.569208	174.998247
29	6.000000	4.288890	-.585795	.018445	.586086	178.196472
30	6.000000	4.800000	-.636415	-.033583	.637300	-176.979362
31	6.000000	7.200000	-.463865	.028786	.464755	176.453913
32	6.000000	7.711110	-.421190	.026528	.422024	176.396105
33	6.000000	8.222220	-.399577	.025951	.400419	176.284027
34	6.000000	8.733330	-.384656	.026079	.385539	176.121418
35	6.000000	9.244440	-.373191	.025805	.374082	176.044495
36	6.000000	9.755560	-.363845	.024649	.364679	176.124414
37	6.000000	10.266670	-.355891	.022824	.356590	176.410637
38	6.000000	10.777780	-.348959	.018680	.349459	176.935886
39	6.000000	11.288890	-.342705	.013302	.342963	177.777174
40	6.000000	11.800000	-.337080	.004657	.337112	179.208387
41	12.000000	-5.000000	-.065668	.454065	.458789	98.229232
42	12.000080	-3.000000	-.125993	.459538	.476497	105.332175
43	14.000000	-5.000000	-.068432	.414936	.420590	99.363956
44	14.000000	-3.000000	-.108002	.401871	.416130	105.042644
45	16.000000	-5.000000	-.057212	.386081	.390277	98.429567
46	16.000000	-3.000000	-.084483	.366014	.375638	102.997381
47	18.000000	-5.000000	-.038505	.365715	.367737	96.010348
48	18.000000	-3.000000	-.057760	.340442	.345308	99.629173
49	20.000000	-5.000000	-.014050	.353910	.354189	92.273434
50	20.000000	-3.000000	-.025137	.320825	.321310	94.486993

COMBINED FLOW FOR OFFBODY POINTS NOZZLE WITH CENTERBOD

I	X(I)	Y(I)	VX	VY	VT	THETA(DEG)
1	-1.000000	.400000	.029678	-.000620	.029684	-1.197022
2	-1.000000	.989470	.029537	-.001525	.029576	-2.955497
3	-1.000000	1.578950	.029282	-.002399	.029380	-4.683645
4	-1.000000	2.168420	.028883	-.003264	.029067	-6.446681
5	-1.000000	2.757890	.028334	-.004073	.028625	-8.181103
6	-1.000000	3.347370	.027670	-.004838	.028084	-9.857625
7	-1.000000	3.936840	.026820	-.005438	.027366	-11.460879
8	-1.000000	4.526320	.025855	-.005905	.026520	-12.865062
9	-1.000000	5.115790	.024770	-.006173	.025527	-13.993022
10	-1.000000	5.705260	.023657	-.006234	.024457	-14.694319
11	-1.000000	6.294740	.022598	-.005991	.023378	-14.847136
12	-1.000000	6.884210	.021639	-.005571	.022345	-14.437365
13	-1.000000	7.473680	.020875	-.005050	.021477	-13.599834
14	-1.000000	8.063160	.020223	-.004431	.020702	-12.358459
15	-1.000000	8.652630	.019717	-.003765	.020073	-10.811596
16	-1.000000	9.242110	.019315	-.003135	.019563	-9.131309
17	-1.000000	9.831580	.019054	-.002428	.019205	-7.203524
18	-1.000000	10.421050	.018828	-.001751	.018909	-5.313341
19	-1.000000	11.010530	.018670	-.001067	.018700	-3.272056
20	-1.000000	11.600000	.018574	-.000424	.018579	-1.306921
21	6.000000	.200000	.051660	-.003695	.051792	-4.091141
22	6.000000	.711110	.047956	-.006184	.048353	-7.347325
23	6.000000	1.222220	.045359	-.006991	.045893	-8.749097
24	6.000000	1.733330	.043656	-.007008	.044214	-9.119477
25	6.000000	2.244440	.042675	-.006586	.043181	-8.772961
26	6.000000	2.755560	.042301	-.005827	.042700	-7.842710
27	6.000000	3.266670	.042519	-.004735	.042777	-6.355403
28	6.000000	3.777780	.043418	-.003224	.043537	-4.246782
29	6.000000	4.288890	.045242	-.001067	.045254	-1.351381
30	6.000000	4.800000	.049402	.002496	.049465	2.880611
31	6.000000	7.200000	.013082	-.000911	.013114	-3.982323
32	6.000000	7.711110	.012086	-.001152	.012141	-5.445735
33	6.000000	8.222220	.011727	-.001320	.011801	-6.423805
34	6.000000	8.733330	.011567	-.001418	.011654	-6.987826
35	6.000000	9.244440	.011475	-.001420	.011563	-7.054155
36	6.000000	9.755560	.011386	-.001354	.011466	-6.782645
37	6.000000	10.266670	.011307	-.001234	.011375	-6.229708
38	6.000000	10.777780	.011158	-.001059	.011208	-5.419569
39	6.000000	11.288890	.011023	-.000839	.011055	-4.351880
40	6.000000	11.800000	.010807	-.001024	.010855	-5.411529
41	12.000000	-5.000000	.004337	-.026520	.026872	-80.711228
42	12.000000	-3.000000	.008396	-.026865	.028146	-72.645399
43	14.000080	-5.000000	.004495	-.023954	.024372	-79.371725
44	14.000000	-3.000000	.007111	-.022958	.024034	-72.790654
45	16.000000	-5.000000	.003701	-.022109	.022416	-80.497566
46	16.000000	-3.000000	.005393	-.020738	.021399	-75.404182
47	18.000000	-5.000000	.002592	-.020777	.020938	-82.890094
48	18.000000	-3.000000	.003806	-.019191	.019476	-78.730321
49	20.000000	-5.000000	.000962	-.019855	.019879	-87.226678
50	20.000000	-3.000000	.001935	-.017778	.017883	-83.789759

OFFBODY POINTS COMPLETE, T = .000 SECONDS.

```
@XQT .PROGC
 NOZZLE WITH HUB
  211  65  99  50   1   1   0   0   1
       .000     .000     .000     .000     .000     .000     .400     .400     .000     .000
       .000     .000     .000     .000     .000     .000   20.000   40.000     .000  350.000
      6.000   -2.000    4.000             6.000  -10.000             4.000
     -1.000     .000   12.000
      6.000     .001    5.000
      6.000    7.000   12.000
       .000     .000    3.000
       .000   12.000    6.000
```

NOZZLE WITH HUB

2-D COMBINATION SOLUTION

COMPRESSIBLE VERSION

COMBINATION OF THE FOLLOWING BASIC SOLUTIONS
1. UNIFORM AXIAL
2. UNIFORM CROSSFLOW
3. VORTICITY ABOUT BODY 1
4. VORTICITY ABOUT BODY 2
5. VORTICITY ABOUT BODY 3

	VELOCITY	MACH NO.	DYNAMIC PRESSURE INC	DYNAMIC PRESSURE COMP	PRESSURE RATIO INC	PRESSURE RATIO COMP	DENSITY RATIO
CONTROL	3.564+02	3.226-01	1.510+02	1.434+02	9.287-01	9.304-01	9.498-01
LOWER PASSAGE	4.395+02	4.000-01	2.296+02	2.123+02	8.915-01	8.956-01	9.243-01
UPPER PASSAGE	4.395+02	4.000-01	2.296+02	2.123+02	8.915-01	8.956-01	9.243-01
FREE STREAM	0.000	0.000	0.000	0.000	1.000+00	1.000+00	1.000+00

ALPHA	VINF/VC	VSONIC	VSONICC	WDOTCR	WDOTLCR	WDOTUCR
0.000	0.000	6.459+02	1.019+03	2.589+01	1.294+01	1.295+01

TSTAT	PSTAT	PSTATC	ASTAT	RHOSTAT	WDOTC	WDOTL	WDOTU
5.187+02	2.116+03	2.116+03	1.116+03	2.378-03	2.589+01	1.294+01	1.295+01
VIC	VICL	VICU					
3.385+02	4.062+02	4.062+02					
TTOT	PTOT	PTOTC	ATOT	RHOTOT	THET	DEL	
5.187+02	2.116+03	2.116+03	1.116+03	2.378-03	1.000+00	1.000+00	

XRI1	YRI1	XRI2	YRI2	XTEST	YCL	YCU	LND
0.000	0.000	0.000	1.200+01	-1.000+00	0.000	1.200+01	1.000+00
[illegible]	YCL1	YCU1	XTEST2	YCL2	YCU2		
[illegible]	1.000-03	5.000+00	6.000+00	7.000+00	1.200+01		

P [illegible] CUTOFF 1	P-S CUTOFF 2	P-S CUTOFF 3
[illegible]000+01	0.000	4.000+01

NT	NP	NS1	NH	KND	IW	NX	ICOMP1	IHUB
211	50	65	99	0	1	1	0	1

	V1	V2	V3	V4	V5
LOWER PASSAGE	8.031-01	-7.005-01	1.328+00	-1.357+00	-6.114-01
UPPER PASSAGE	4.998-01	-4.358-01	8.266-01	1.429+00	-3.805-01

A	B	C	D	VINFR
0.000	0.000	2.354+02	-3.028+02	0.000

BODY 1
ON-BODY POINTS

I	X	Y	S	V	MACH	CP	PS/PT
1	-4.287+01	0.000	-4.287+01	3.589+02	3.249-01	9.999+03	.9295
2	-3.860+01	0.000	-3.860+01	3.528+02	3.192-01	9.999+03	.9318
3	-3.433+01	0.000	-3.433+01	3.530+02	3.194-01	9.999+03	.9317
4	-3.007+01	0.000	-3.007+01	3.534+02	3.198-01	9.999+03	.9316
5	-2.580+01	0.000	-2.580+01	3.536+02	3.200-01	9.999+03	.9315
6	-2.153+01	0.000	-2.153+01	3.536+02	3.200-01	9.999+03	.9315
7	-1.762+01	0.000	-1.762+01	3.536+02	3.200-01	9.999+03	.9515
8	-1.435+01	0.000	-1.435+01	3.536+02	3.200-01	9.999+03	.9315
9	-1.162+01	0.000	-1.162+01	3.537+02	3.201-01	9.999+03	.9315
10	-9.352+00	0.000	-9.352+00	3.538+02	3.202-01	9.999+03	.9314
11	-7.460+00	0.000	-7.460+00	3.541+02	3.205-01	9.999+03	.9313
12	-5.883+00	0.000	-5.883+00	3.545+02	3.209-01	9.999+03	.9311
13	-4.569+00	0.000	-4.569+00	3.552+02	3.215-01	9.999+03	.9309
14	-3.474+00	0.000	-3.474+00	3.563+02	3.225-01	9.999+03	.9305
15	-2.562+00	0.000	-2.562+00	3.577+02	3.238-01	9.999+03	.9299
16	-1.802+00	0.000	-1.802+00	3.594+02	3.254-01	9.999+03	.9293
17	-1.168+00	0.000	-1.168+00	3.614+02	3.273-01	9.999+03	.9285
18	-6.400-01	0.000	-6.400-01	3.636+02	3.293-01	9.999+03	.9277
19	-2.000-01	0.000	-2.000-01	3.658+02	3.313-01	9.999+03	.9268
20	2.000-01	0.000	2.000-01	3.682+02	3.335-01	9.999+03	.9259
21	6.000-01	0.000	6.000-01	3.710+02	3.361-01	9.999+03	.9248
22	1.000+00	0.000	1.000+00	3.743+02	3.391-01	9.999+03	.9235
23	1.400+00	0.000	1.400+00	3.781+02	3.427-01	9.999+03	.9220
24	1.800+00	0.000	1.800+00	3.824+02	3.467-01	9.999+03	.9202
25	2.200+00	0.000	2.200+00	3.874+02	3.514-01	9.999+03	.9182
26	2.600+00	0.000	2.600+00	3.932+02	3.567-01	9.999+03	.9158
27	3.000+00	0.000	3.000+00	3.997+02	3.628-01	9.999+03	.9131
28	3.400+00	0.000	3.400+00	4.072+02	3.697-01	9.999+03	.9099
29	3.800+00	0.000	3.800+00	4.159+02	3.779-01	9.999+03	.9062
30	4.200+00	0.000	4.200+00	4.262+02	3.875-01	9.999+03	.9016
31	4.600+00	0.000	4.600+00	4.389+02	3.994-01	9.999+03	.8959
32	5.000+00	0.000	5.000+00	4.553+02	4.149-01	9.999+03	.8883
33	5.400+00	0.000	5.400+00	4.784+02	4.367-01	9.999+03	.8772
34	5.800+00	0.000	5.800+00	5.185+02	4.749-01	9.999+03	.8569
35	6.183+00	-7.327-03	6.183+00	5.889+02	5.429-01	9.999+03	.8183
36	6.548+00	-5.057-02	6.551+00	6.423+02	5.955-01	9.999+03	.7868
37	6.908+00	-1.408-01	6.922+00	6.723+02	6.255-01	9.999+03	.7682
38	7.252+00	-2.739-01	7.291+00	6.942+02	6.475-01	9.999+03	.7544
39	7.575+00	-4.464-01	7.656+00	7.103+02	6.639-01	9.999+03	.7441
40	7.873+00	-6.566-01	8.022+00	7.221+02	6.758-01	9.999+03	.7364
41	8.144+00	-9.012-01	8.386+00	7.295+02	6.834-01	9.999+03	.7316
42	8.377+00	-1.170+00	8.743+00	7.320+02	6.859-01	9.999+03	.7300
43	8.575+00	-1.460+00	9.093+00	7.288+02	6.827-01	9.999+03	.7321
44	8.738+00	-1.774+00	9.447+00	7.186+02	6.723-01	9.999+03	.7387
45	8.864+00	-2.108+00	9.804+00	6.996+02	6.529-01	9.999+03	.7510
46	8.951+00	-2.457+00	1.016+01	6.692+02	6.224-01	9.999+03	.7702
47	8.993+00	-2.818+00	1.053+01	6.130+02	5.665-01	9.999+03	.8043
48	9.000+00	-3.195+00	1.090+01	5.391+02	4.947-01	9.999+03	.8460
49	9.000+00	-3.623+00	1.133+01	4.921+02	4.497-01	9.999+03	.8704
50	9.000+00	-4.137+00	1.185+01	4.612+02	4.204-01	9.999+03	.8855

I	X	Y	S	V	MACH	CP	PS/PT
51	9.000+00	-4.754+00	1.246+01	4.369+02	3.976-01	9.999+03	.8968
52	9.000+00	-5.494+00	1.320+01	4.172+02	3.791-01	9.999+03	.9056
53	9.000+00	-6.382+00	1.409+01	4.010+02	3.640-01	9.999+03	.9125
54	9.000+00	-7.448+00	1.516+01	3.878+02	3.517-01	9.999+03	.9180
55	9.000+00	-8.727+00	1.644+01	3.772+02	3.419-01	9.999+03	.9223
56	9.000+00	-1.026+01	1.797+01	3.691+02	3.344-01	9.999+03	.9255
57	9.000+00	-1.210+01	1.981+01	3.631+02	3.288-01	9.999+03	.9278
58	9.000+00	-1.431+01	2.202+01	3.591+02	3.251-01	9.999+03	.9294
59	9.000+00	-1.697+01	2.468+01	3.565+02	3.227-01	9.999+03	.9304
60	9.000+00	-2.015+01	2.786+01	3.551+02	3.214-01	9.999+03	.9309
61	9.000+00	-2.370+01	3.141+01	3.545+02	3.208-01	9.999+03	.9312
62	9.000+00	-2.732+01	3.503+01	3.542+02	3.205-01	9.999+03	.9313
63	9.000+00	-3.094+01	3.865+01	3.541+02	3.205-01	9.999+03	.9313
64	9.000+00	-3.457+01	4.227+01	3.559+02	3.221-01	9.999+03	.9306
65	9.000+00	-3.819+01	4.590+01	3.707+02	3.359-01	9.999+03	.9249

BODY 2
ON-BODY POINTS

I	X	Y	S	V	MACH	CP	PS/PT
66	1.045+01	5.889+00	-7.869+00	2.745+02	2.474-01	9.999+03	.9583
67	9.407+00	5.681+00	-6.808+00	-8.511+01	7.630-02	9.999+03	.9959
68	8.512+00	5.502+00	-5.896+00	-1.657+02	1.488-01	9.999+03	.9847
69	7.767+00	5.353+00	-5.136+00	-2.342+02	2.107-01	9.999+03	.9695
70	7.145+00	5.229+00	-4.502+00	-2.903+02	2.619-01	9.999+03	.9534
71	6.628+00	5.126+00	-3.974+00	-3.418+02	3.091-01	9.999+03	.9359
72	6.196+00	5.039+00	-3.534+00	-4.127+02	3.749-01	9.999+03	.9075
73	5.825+00	4.999+00	-3.161+00	-4.841+02	4.421-01	9.999+03	.8744
74	5.476+00	5.015+00	-2.811+00	-4.522+02	4.119-01	9.999+03	.8898
75	5.128+00	5.043+00	-2.462+00	-4.601+02	4.194-01	9.999+03	.8860
76	4.781+00	5.086+00	-2.113+00	-4.644+02	4.235-01	9.999+03	.8840
77	4.438+00	5.146+00	-1.764+00	-4.660+02	4.250-01	9.999+03	.8832
78	4.099+00	5.226+00	-1.416+00	-4.633+02	4.224-01	9.999+03	.8845
79	3.769+00	5.331+00	-1.069+00	-4.534+02	4.131-01	9.999+03	.8892
80	3.458+00	5.468+00	-7.302-01	-4.283+02	3.895-01	9.999+03	.9007
81	3.194+00	5.645+00	-4.122-01	-3.592+02	3.252-01	9.999+03	.9294
82	3.028+00	5.872+00	-1.310-01	-1.537+02	1.380-01	9.999+03	.9868
83	3.028+00	6.128+00	1.310-01	1.696+02	1.523-01	9.999+03	.9839
84	3.194+00	6.355+00	4.122-01	3.766+02	3.413-01	9.999+03	.9225
85	3.458+00	6.532+00	7.302-01	4.467+02	4.068-01	9.999+03	.8923
86	3.769+00	6.669+00	1.069+00	4.768+02	4.352-01	9.999+03	.8780
87	4.099+00	6.774+00	1.416+00	4.934+02	4.510-01	9.999+03	.8697
88	4.438+00	6.854+00	1.764+00	5.047+02	4.617-01	9.999+03	.8641
89	4.781+00	6.914+00	2.113+00	5.139+02	4.705-01	9.999+03	.8593
90	5.128+00	6.957+00	2.462+00	5.231+02	4.793-01	9.999+03	.8545
91	5.476+00	6.985+00	2.811+00	5.367+02	4.923-01	9.999+03	.8473
92	5.825+00	7.001+00	3.161+00	5.608+02	5.157-01	9.999+03	.8341
93	6.196+00	6.961+00	3.534+00	5.715+02	5.260-01	9.999+03	.8282
94	6.628+00	6.874+00	3.974+00	4.958+02	4.532-01	9.999+03	.8686
95	7.145+00	6.771+00	4.502+00	4.787+02	4.370-01	9.999+03	.8770
96	7.767+00	6.647+00	5.136+00	4.729+02	4.315-01	9.999+03	.8799
97	8.512+00	6.498+00	5.896+00	4.795+02	4.377-01	9.999+03	.8767
98	9.407+00	6.319+00	6.808+00	4.935+02	4.511-01	9.999+03	.8697
99	1.045+01	6.111+00	7.869+00	8.110+02	7.683-01	9.999+03	.6767

BODY 3
ON-BODY POINTS

I	X	Y	S	V	MACH	CP	PS/PT
100	2.100+01	-3.824+01	6.480+01	-3.800+02	3.445-01	9.999+03	.9212
101	2.100+01	-3.472+01	6.128+01	-3.590+02	3.250-01	9.999+03	.9294
102	2.100+01	-3.120+01	5.776+01	-3.552+02	3.215-01	9.999+03	.9309
103	2.100+01	-2.768+01	5.424+01	-3.543+02	3.207-01	9.999+03	.9312
104	2.100+01	-2.416+01	5.072+01	-3.539+02	3.203-01	9.999+03	.9314
105	2.100+01	-2.062+01	4.718+01	-3.532+02	3.197-01	9.999+03	.9316
106	2.100+01	-1.735+01	4.391+01	-3.519+02	3.185-01	9.999+03	.9321
107	2.100+01	-1.462+01	4.118+01	-3.497+02	3.164-01	9.999+03	.9330
108	2.100+01	-1.235+01	3.891+01	-3.461+02	3.131-01	9.999+03	.9345
109	2.100+01	-1.046+01	3.702+01	-3.412+02	3.086-01	9.999+03	.9361
110	2.100+01	-8.883+00	3.544+01	-3.350+02	3.029-01	9.999+03	.9384
111	2.100+01	-7.569+00	3.413+01	-3.277+02	2.962-01	9.999+03	.9409
112	2.100+01	-6.474+00	3.304+01	-3.197+02	2.888-01	9.999+03	.9437
113	2.100+01	-5.562+00	3.212+01	-3.113+02	2.811-01	9.999+03	.9466
114	2.100+01	-4.802+00	3.136+01	-3.027+02	2.732-01	9.999+03	.9495
115	2.100+01	-4.168+00	3.073+01	-2.940+02	2.652-01	9.999+03	.9523
116	2.100+01	-3.640+00	3.020+01	-2.853+02	2.573-01	9.999+03	.9550
117	2.100+01	-3.200+00	2.976+01	-2.763+02	2.490-01	9.999+03	.9578
118	2.100+01	-2.804+00	2.936+01	-2.648+02	2.386-01	9.999+03	.9611
119	2.099+01	-2.411+00	2.897+01	-2.566+02	2.311-01	9.999+03	.9635
120	2.097+01	-2.019+00	2.858+01	-2.508+02	2.259-01	9.999+03	.9651
121	2.094+01	-1.627+00	2.819+01	-2.462+02	2.217-01	9.999+03	.9664
122	2.090+01	-1.236+00	2.779+01	-2.423+02	2.181-01	9.999+03	.9674
123	2.084+01	-8.457-01	2.740+01	-2.391+02	2.152-01	9.999+03	.9682
124	2.078+01	-4.571-01	2.701+01	-2.363+02	2.127-01	9.999+03	.9690
125	2.071+01	-7.010-02	2.661+01	-2.339+02	2.105-01	9.999+03	.9696
126	2.063+01	3.150-01	2.622+01	-2.320+02	2.087-01	9.999+03	.9701
127	2.054+01	6.980-01	2.582+01	-2.303+02	2.072-01	9.999+03	.9705
128	2.043+01	1.079+00	2.543+01	-2.289+02	2.059-01	9.999+03	.9709
129	2.032+01	1.456+00	2.504+01	-2.278+02	2.049-01	9.999+03	.9711
130	2.020+01	1.831+00	2.464+01	-2.270+02	2.042-01	9.999+03	.9714
131	2.007+01	2.203+00	2.425+01	-2.264+02	2.037-01	9.999+03	.9715
132	1.993+01	2.572+00	2.385+01	-2.261+02	2.034-01	9.999+03	.9716
133	1.978+01	2.936+00	2.346+01	-2.260+02	2.033-01	9.999+03	.9716
134	1.961+01	3.297+00	2.306+01	-2.261+02	2.034-01	9.999+03	.9716
135	1.944+01	3.653+00	2.267+01	-2.265+02	2.038-01	9.999+03	.9715
136	1.926+01	4.005+00	2.227+01	-2.272+02	2.044-01	9.999+03	.9713
137	1.907+01	4.352+00	2.188+01	-2.281+02	2.052-01	9.999+03	.9711
138	1.888+01	4.695+00	2.148+01	-2.292+02	2.063-01	9.999+03	.9708
139	1.867+01	5.032+00	2.109+01	-2.307+02	2.076-01	9.999+03	.9704
140	1.845+01	5.363+00	2.069+01	-2.324+02	2.091-01	9.999+03	.9700
141	1.823+01	5.689+00	2.029+01	-2.344+02	2.110-01	9.999+03	.9695
142	1.799+01	6.009+00	1.990+01	-2.368+02	2.131-01	9.999+03	.9689
143	1.775+01	6.323+00	1.950+01	-2.394+02	2.155-01	9.999+03	.9682
144	1.750+01	6.631+00	1.910+01	-2.424+02	2.182-01	9.999+03	.9674
145	1.724+01	6.932+00	1.871+01	-2.458+02	2.213-01	9.999+03	.9665
146	1.697+01	7.226+00	1.831+01	-2.495+02	2.246-01	9.999+03	.9655
147	1.670+01	7.514+00	1.791+01	-2.536+02	2.284-01	9.999+03	.9643
148	1.642+01	7.790+00	1.752+01	-2.579+02	2.323-01	9.999+03	.9631
149	1.614+01	8.356+00	1.713+01	-2.626+02	2.365-01	9.999+03	.9618

I	X	Y	S	V	MACH	CP	PS/PT
150	1.585+01	8.315+00	1.674+01	-2.675+02	2.411-01	9.999+03	.9604
151	1.555+01	8.566+00	1.635+01	-2.727+02	2.458-01	9.999+03	.9588
152	1.525+01	8.809+00	1.597+01	-2.782+02	2.508-01	9.999+03	.9572
153	1.494+01	9.045+00	1.558+01	-2.838+02	2.559-01	9.999+03	.9555
154	1.462+01	9.273+00	1.519+01	-2.895+02	2.612-01	9.999+03	.9537
155	1.430+01	9.492+00	1.480+01	-2.952+02	2.664-01	9.999+03	.9519
156	1.398+01	9.704+00	1.441+01	-3.008+02	2.715-01	9.999+03	.9501
157	1.364+01	9.906+00	1.402+01	-3.061+02	2.764-01	9.999+03	.9483
158	1.331+01	1.010+01	1.363+01	-3.112+02	2.810-01	9.999+03	.9466
159	1.296+01	1.029+01	1.324+01	-3.159+02	2.853-01	9.999+03	.9451
160	1.261+01	1.046+01	1.285+01	-3.201+02	2.891-01	9.999+03	.9436
161	1.226+01	1.063+01	1.246+01	-3.238+02	2.926-01	9.999+03	.9423
162	1.191+01	1.079+01	1.207+01	-3.271+02	2.956-01	9.999+03	.9412
163	1.154+01	1.094+01	1.168+01	-3.299+02	2.981-01	9.999+03	.9402
164	1.118+01	1.108+01	1.129+01	-3.322+02	3.003-01	9.999+03	.9393
165	1.081+01	1.121+01	1.090+01	-3.343+02	3.022-01	9.999+03	.9386
166	1.044+01	1.133+01	1.051+01	-3.360+02	3.038-01	9.999+03	.9380
167	1.006+01	1.144+01	1.012+01	-3.376+02	3.053-01	9.999+03	.9374
168	9.686+00	1.154+01	9.724+00	-3.390+02	3.066-01	9.999+03	.9369
169	9.305+00	1.163+01	9.333+00	-3.405+02	3.079-01	9.999+03	.9364
170	8.922+00	1.171+01	8.941+00	-3.419+02	3.093-01	9.999+03	.9358
171	8.537+00	1.178+01	8.549+00	-3.435+02	3.107-01	9.999+03	.9353
172	8.150+00	1.185+01	8.158+00	-3.453+02	3.124-01	9.999+03	.9346
173	7.762+00	1.190+01	7.766+00	-3.474+02	3.143-01	9.999+03	.9338
174	7.372+00	1.194+01	7.373+00	-3.499+02	3.166-01	9.999+03	.9329
175	6.981+00	1.197+01	6.981+00	-3.530+02	3.195-01	9.999+03	.9317
176	6.589+00	1.199+01	6.589+00	-3.571+02	3.233-01	9.999+03	.9302
177	6.196+00	1.200+01	6.196+00	-3.645+02	3.301-01	9.999+03	.9273
178	5.800+00	1.200+01	5.800+00	-3.753+02	3.401-01	9.999+03	.9231
179	5.400+00	1.200+01	5.400+00	-3.810+02	3.454-01	9.999+03	.9208
180	5.000+00	1.200+01	5.000+00	-3.841+02	3.482-01	9.999+03	.9195
181	4.600+00	1.200+01	4.600+00	-3.853+02	3.494-01	9.999+03	.9190
182	4.200+00	1.200+01	4.200+00	-3.852+02	3.493-01	9.999+03	.9191
183	3.800+00	1.200+01	3.800+00	-3.842+02	3.484-01	9.999+03	.9195
184	3.400+00	1.200+01	3.400+00	-3.824+02	3.467-01	9.999+03	.9202
185	3.000+00	1.200+01	3.000+00	-3.802+02	3.447-01	9.999+03	.9211
186	2.600+00	1.200+01	2.600+00	-3.777+02	3.423-01	9.999+03	.9221
187	2.200+00	1.200+01	2.200+00	-3.751+02	3.399-01	9.999+03	.9232
188	1.800+00	1.200+01	1.800+00	-3.724+02	3.374-01	9.999+03	.9242
189	1.400+00	1.200+01	1.400+00	-3.699+02	3.351-01	9.999+03	.9252
190	1.000+00	1.200+01	1.000+00	-3.676+02	3.329-01	9.999+03	.9261
191	6.000-01	1.200+01	6.000-01	-3.654+02	3.310-01	9.999+03	.9270
192	2.000-01	1.200+01	2.000-01	-3.635+02	3.292-01	9.999+03	.9277
193	-2.000-01	1.200+01	-2.000-01	-3.619+02	3.277-01	9.999+03	.9283
194	-6.400-01	1.200+01	-6.400-01	-3.603+02	3.262-01	9.999+03	.9289
195	-1.168+00	1.200+01	-1.168+00	-3.587+02	3.248-01	9.999+03	.9295
196	-1.802+00	1.200+01	-1.802+00	-3.573+02	3.235-01	9.999+03	.9301
197	-2.562+00	1.200+01	-2.562+00	-3.561+02	3.223-01	9.999+03	.9306
198	-3.474+00	1.200+01	-3.474+00	-3.551+02	3.214-01	9.999+03	.9309
199	-4.569+00	1.200+01	-4.569+00	-3.544+02	3.207-01	9.999+03	.9312
200	-5.883+00	1.200+01	-5.883+00	-3.539+02	3.203-01	9.999+03	.9314
201	-7.460+00	1.200+01	-7.460+00	-3.537+02	3.201-01	9.999+03	.9315
202	-9.352+00	1.200+01	-9.352+00	-3.536+02	3.200-01	9.999+03	.9315
203	-1.162+01	1.200+01	-1.162+01	-3.536+02	3.200-01	9.999+03	.9315
204	-1.435+01	1.200+01	-1.435+01	-3.536+02	3.200-01	9.999+03	.9315
205	-1.762+01	1.200+01	-1.762+01	-3.536+02	3.200-01	9.999+03	.9315
206	-2.153+01	1.200+01	-2.153+01	-3.537+02	3.201-01	9.999+03	.9315
207	-2.580+01	1.200+01	-2.580+01	-3.538+02	3.202-01	9.999+03	.9314
208	-3.007+01	1.200+01	-3.007+01	-3.540+02	3.204-01	9.999+03	.9313
209	-3.433+01	1.200+01	-3.433+01	-3.549+02	3.212-01	9.999+03	.9310
210	-3.860+01	1.200+01	-3.860+01	-3.588+02	3.248-01	9.999+03	.9295
211	-4.287+01	1.200+01	-4.287+01	-3.795+02	3.440-01	9.999+03	.9214

RAKE NUMBER 1

I	X	Y	VX	VY	VRE	THETA	MACH	PS/PT	WFRACT
	-1.000+00	0.000	3.621+02	0.000	3.621+02	0.000	3.279-01	.9283	0.000
1	-1.000+00	4.000-01	3.619+02	-1.473+00	3.619+02	-2.332-01	3.277-01	.9283	3.404-02
2	-1.000+00	9.895-01	3.612+02	-3.546+00	3.612+02	-5.624-01	3.271-01	.9286	8.415-02
3	-1.000+00	1.579+00	3.600+02	-6.428+00	3.600+02	-8.640-01	3.259-01	.9291	1.341-01
4	-1.000+00	2.168+00	3.582+02	-6.951+00	3.582+02	-1.109+00	3.243-01	.9297	1.839-01
5	-1.000+00	2.758+00	3.559+02	-7.937+00	3.560+02	-1.277+00	3.223-01	.9306	2.335-01
6	-1.000+00	3.347+00	3.534+02	-8.291+00	3.535+02	-1.344+00	3.199-01	.9315	2.827-01
7	-1.000+00	3.937+00	3.508+02	-7.883+00	3.509+02	-1.287+00	3.175-01	.9325	3.317-01
8	-1.000+00	4.526+00	3.484+02	-6.653+00	3.484+02	-1.094+00	3.153-01	.9334	3.803-01
9	-1.000+00	5.116+00	3.465+02	-4.664+00	3.465+02	-7.712-01	3.135-01	.9341	4.286-01
10	-1.000+00	5.705+00	3.454+02	-2.146+00	3.454+02	-3.559-01	3.125-01	.9345	4.768-01
11	-1.000+00	6.295+00	3.454+02	5.416-01	3.454+02	8.983-02	3.125-01	.9546	5.249-01
12	-1.000+00	6.884+00	3.464+02	2.977+00	3.464+02	4.924-01	3.134-01	.9342	5.730-01
13	-1.000+00	7.474+00	3.481+02	4.855+00	3.481+02	7.991-01	3.150-01	.9335	6.214-01
14	-1.000+00	8.063+00	3.502+02	5.961+00	3.503+02	9.751-01	3.170-01	.9327	6.699-01
15	-1.000+00	8.653+00	3.525+02	6.301+00	3.525+02	1.024+00	3.191-01	.9319	7.188-01
16	-1.000+00	9.242+00	3.546+02	5.987+00	3.546+02	9.674-01	3.210-01	.9311	7.679-01
17	-1.000+00	9.832+00	3.563+02	5.163+00	3.563+02	8.302-01	3.226-01	.9305	8.172-01
18	-1.000+00	1.042+01	3.577+02	3.992+00	3.577+02	6.395-01	3.238-01	.9299	8.667-01
19	-1.000+00	1.101+01	3.586+02	2.592+00	3.586+02	4.142-01	3.246-01	.9296	9.164-01
20	-1.000+00	1.163+01	3.591+02	1.060+00	3.591+02	1.692-01	3.251-01	.9294	9.662-01
	-1.000+00	1.200+01	3.592+02	0.000	3.592+02	0.000	3.252-01	.9294	1.000+00

RAKE NUMBER 2

I	X	Y	WX	VY	VRE	THETA	MACH	PS/PT	WFRACT
	6.000+00	1.033-03	5.570+02	-1.065+01	5.571+02	-1.096+00	5.121-01	.8361	0.000
1	6.000+00	2.000-01	5.346+02	-3.761+01	5.359+02	-4.024+00	4.916-01	.8477	4.824-02
2	6.000+00	7.111-01	4.891+02	-7.107+01	4.942+02	-8.267+00	4.517-01	.8693	1.660-01
3	6.000+00	1.222+00	4.542+02	-8.196+01	4.615+02	-1.023+01	4.207-01	.8854	2.763-01
4	6.000+00	1.733+00	4.275+02	-8.286+01	4.355+02	-1.097+01	3.962-01	.8974	3.805-01
5	6.000+00	2.244+00	4.077+02	-7.801+01	4.151+02	-1.083+01	3.772-01	.9065	4.801-01
6	6.000+00	2.756+00	3.940+02	-6.902+01	4.000+02	-9.936+00	3.631-01	.9130	5.763-01
7	6.000+00	3.267+00	3.864+02	-5.630+01	3.905+02	-8.290+00	3.542-01	.9169	6.703-01
8	6.000+00	3.778+00	3.858+02	-3.930+01	3.878+02	-5.815+00	3.517-01	.9180	7.636-01
9	6.000+00	4.289+00	3.953+02	-1.590+01	3.956+02	-2.304+00	3.590-01	.9148	8.578-01
10	6.000+00	4.800+00	4.288+02	2.375+01	4.295+02	3.170+00	3.906-01	.9001	9.565-01
	6.000+00	5.014+00	4.467+02	4.786+01	4.493+02	6.116+00	4.092-01	.8910	1.000+00

RAKE NUMBER 3

I	X	Y	VX	VY	VRE	THETA	MACH	PS/PT	WFRACT
	6.000+00	6.986+00	5.713+02	-6.122+01	5.746+02	-6.116+00	5.289-01	.8267	0.000
1	6.000+00	7.200+00	5.430+02	-3.237+01	5.439+02	-3.412+00	4.993-01	.8439	5.267-02
2	6.000+00	7.711+00	4.884+02	-2.638+01	4.891+02	-3.092+00	4.468-01	.8719	1.712-01
3	6.000+00	8.222+00	4.595+02	-2.375+01	4.601+02	-2.959+00	4.194-01	.8860	2.821-01
4	6.000+00	8.733+00	4.386+02	-2.298+01	4.392+02	-3.000+00	3.997-01	.8958	3.881-01
5	6.000+00	9.244+00	4.221+02	-2.261+01	4.227+02	-3.066+00	3.843-01	.9032	4.905-01
6	6.000+00	9.756+00	4.086+02	-2.185+01	4.091+02	-3.062+00	3.716-01	.9091	5.898-01
7	6.000+00	1.027+01	3.972+02	-2.024+01	3.977+02	-2.918+00	3.609-01	.9139	6.866-01
8	6.000+00	1.078+01	3.875+02	-1.746+01	3.878+02	-2.581+00	3.518-01	.9180	7.811-01
9	6.000+00	1.129+01	3.791+02	-1.302+01	3.793+02	-1.968+00	3.438-01	.9215	8.738-01
10	6.000+00	1.180+01	3.719+02	-5.624+00	3.719+02	-8.664-01	3.370-01	.9244	9.648-01
	6.000+00	1.200+01	3.694+02	-1.553+00	3.694+02	-2.409-01	3.347-01	.9254	1.000+00

RAKE NUMBER 4

I	X	Y	VX	VY	VRE	THETA	MACH	PS/PT	WFRACT
	1.200+01	-3.819+01	3.707+02	0.000	3.707+02	0.000	3.359-01	.9249	0.000
1	1.200+01	-5.000+00	5.096+01	-3.891+02	3.925+02	-8.254+01	3.560-01	.9161	1.107+00
2	1.200+01	-3.000+00	9.732+01	-3.940+02	4.058+02	-7.613+01	3.685-01	.9105	1.130+00
	1.200+01	5.889+00	-2.745+02	0.000	-2.745+02	0.000	2.474-01	.9583	1.000+00

RAKE NUMBER 5

I	X	Y	VX	VY	VRE	THETA	MACH	PS/PT	WFRACT
	1.400+01	-3.819+01	3.707+02	0.000	3.707+02	0.000	3.359-01	.9249	0.000
1	1.400+01	-5.000+00	5.311+01	-3.575+02	3.615+02	-8.155+01	3.273-01	.9285	1.118+00
2	1.400+01	-3.000+00	8.351+01	-3.480+02	3.578+02	-7.650+01	3.239-01	.9299	1.140+00
	1.400+01	5.889+00	-2.745+02	0.000	-2.745+02	0.000	2.474-01	.9583	1.000+00

RAKE NUMBER 6

I	X	Y	VX	VY	VRE	THETA	MACH	PS/PT	WFRACT
	1.600+01	-3.819+01	3.707+02	0.000	3.707+02	0.000	3.359-01	.9249	0.000
1	1.600+01	-5.000+00	4.472+01	-3.339+02	3.369+02	-8.237+01	3.046-01	.9377	1.139+00
2	1.600+01	-3.000+00	6.626+01	-3.186+02	3.254+02	-7.875+01	2.941-01	.9418	1.157+00
	1.600+01	5.889+00	-2.745+02	0.000	-2.745+02	0.000	2.474-01	.9583	1.000+00

RAKE NUMBER 7

I	X	Y	VX	VY	VRE	THETA	MACH	PS/PT	WFRACT

	1.800+01	-3.819+01	3.707+02	0.000	3.707+02	0.000	3.359-01	.9249	0.000
1	1.800+01	-5.000+00	3.044+01	-3.168+02	3.183+02	-8.451+01	2.875-01	.9442	1.169+00
2	1.800+01	-3.000+00	4.630+01	-2.967+02	3.003+02	-8.113+01	2.710-01	.9503	1.183+00
	1.800+01	5.889+00	-2.745+02	0.000	-2.745+02	0.000	2.478-01	.9583	1.000+00

RAKE NUMBER 8

I	X	Y	VX	VY	VRE	THETA	MACH	PS/PT	WFRACT
	2.000+01	-3.819+01	3.707+02	0.000	3.707+02	0.000	3.359-01	.9249	0.000
1	2.000+01	-5.000+00	1.120+01	-3.066+02	3.068+02	-8.791+01	2.770-01	.9481	1.216+00
2	2.000+01	-3.000+00	2.071+01	-2.786+02	2.794+02	-8.575+01	2.519-01	.9568	1.222+00
	2.000+01	5.889+00	-2.745+02	0.000	-2.745+02	0.000	2.478-01	.9583	1.000+00

RAKE WEIGHT FLOW DATA

I	X	IRAK	WDOT	WDOTCA	MACH
1	-1.0000	3	2.57138+01	2.57143+01	3.20048-01
2	6.0000	5	1.26896+01	3.03754+01	3.89041-01
3	6.0000	4	1.26970+01	3.03856+01	3.89200-01
4	12.0000	5	3.80703+01	1.03647+01	1.22483-01
5	14.0000	5	3.79438+01	1.03302+01	1.22069-01
6	16.0000	5	3.65363+01	9.94701+00	1.17463-01
7	18.0000	5	3.43602+01	9.35703+00	1.10390-01
8	20.0000	5	3.14492+01	8.56205+00	1.00890-01

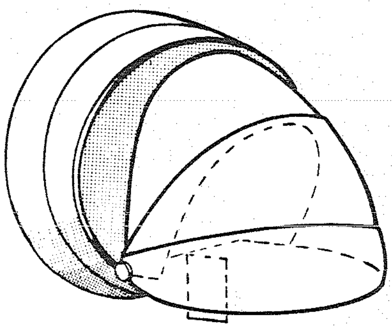
aBRKPT PRINTS

REFERENCES

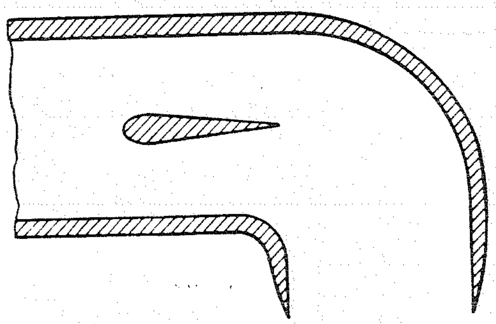
1. Stockman, Norbert O.; and Farrell, Charles A., Jr.: Improved Computer Programs for Calculating Potential Flow in Propulsion System Inlets. NASA TM-73728, 1977.

2. Hawk, J. Dennis; Stockman, Norbert O.; and Farrell, Charles A., Jr.: Computer Programs for Calculating Two-Dimensional Potential Flow in and About Propulsion System Inlets. NASA TM-78930, 1978.

3. Hess, J. L.; and Smith, Apollo M. O.: Calculations of Potential Flow About Arbitrary Bodies. Progress in Aeronautical Sciences, Vol. 8, D. Kuchemann, ed., Pergaman Press, 1967, pp. 1-138.

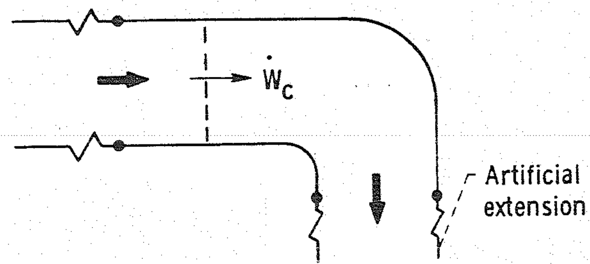


(a) Three-dimensional nozzle.

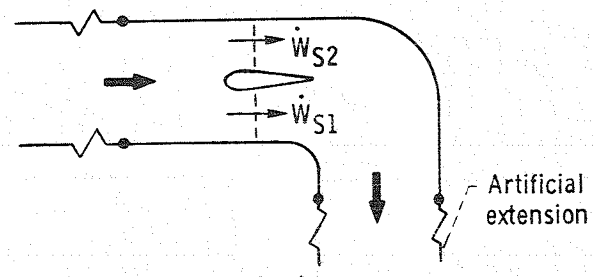


(b) Section through plane of symmetry for two-dimensional approximation.

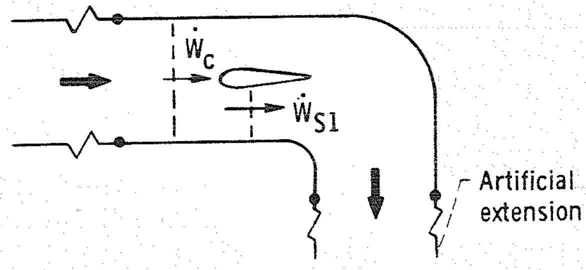
Figure 3. - Two-dimensional approximation of three-dimensional nozzle.



(a) Single passage.



(b) Dual passage.



(c) Dual passage: alternate weight flow specification.

Figure 4. - Nozzle geometries and flow conditions.

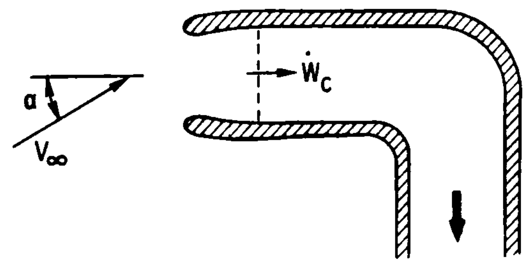


Figure 5. - Inlet and nozzle geometry

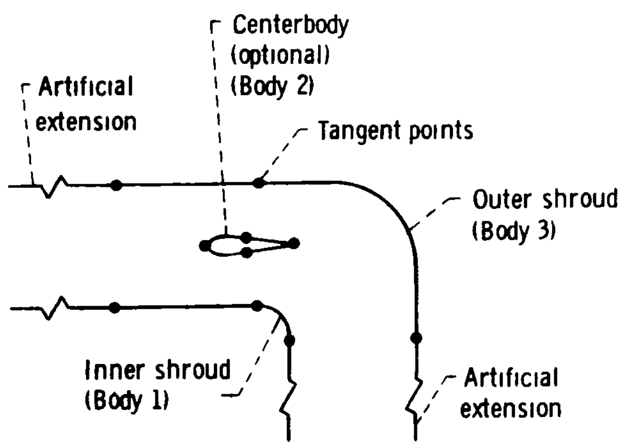


Figure 6. - Typical nozzle segmentation.

At any point

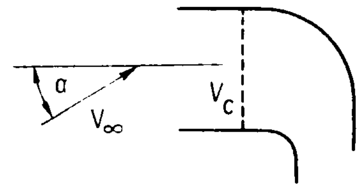
$$\overline{V} = A\overline{V}_1 + B\overline{V}_2 + C(\overline{V}_3 - \overline{V}_4)$$

where A, B, and C are determined by specifying values of

V_c average axial velocities at the control station

V_∞ free stream velocity

α direction of free stream velocity relative to nozzle axis



(a) Single passage nozzle

At any point

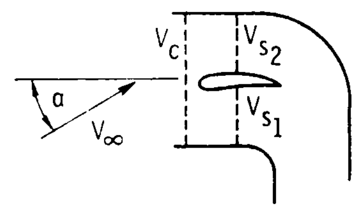
$$\overline{V} = A\overline{V}_1 + B\overline{V}_2 + C(\overline{V}_3 - \overline{V}_4) + D(\overline{V}_5 - \overline{V}_4)$$

where A, B, C, and D are determined by specifying values of

V_c, V_{s_1}, V_{s_2} average axial velocities at any two of the three control stations

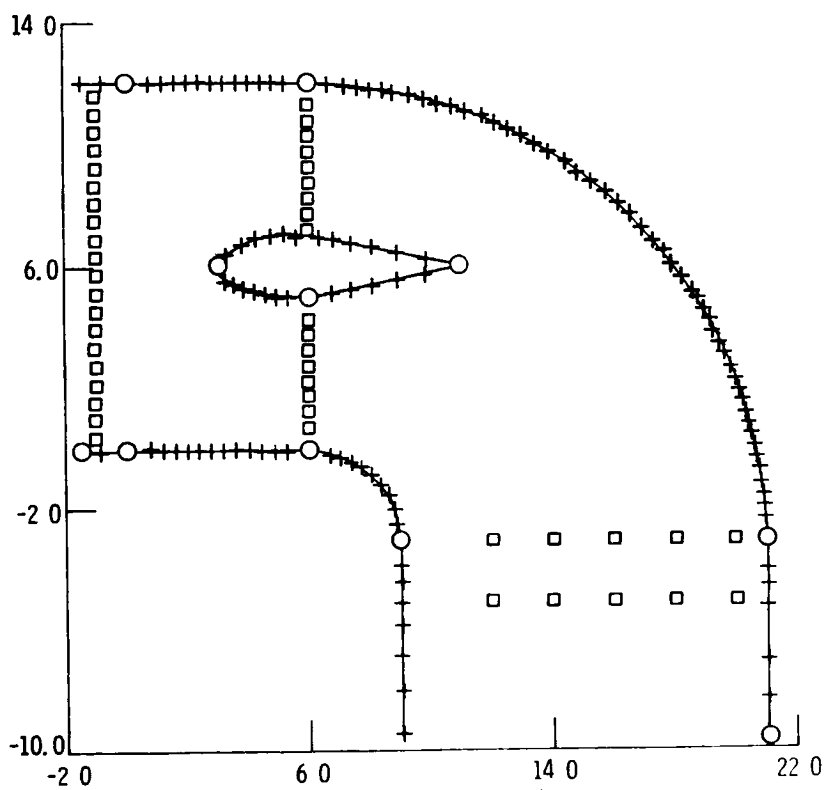
V_∞ free stream velocity

α direction of free stream velocity relative to nozzle axis



(b) Two passage nozzle

Figure 7 - Combined solution, $\overline{V}$



Nozzle with centerbody

Figure 8 - Graphic output of SCIRCL

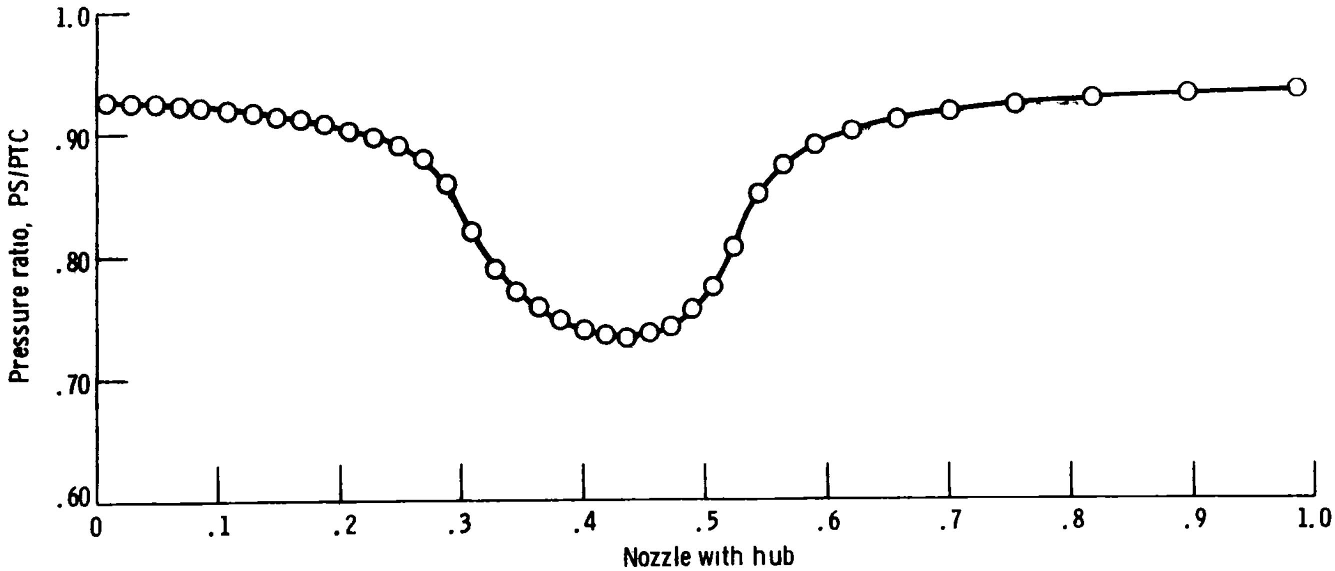


Figure 9. - Surface static pressure ratio from NOZZLE

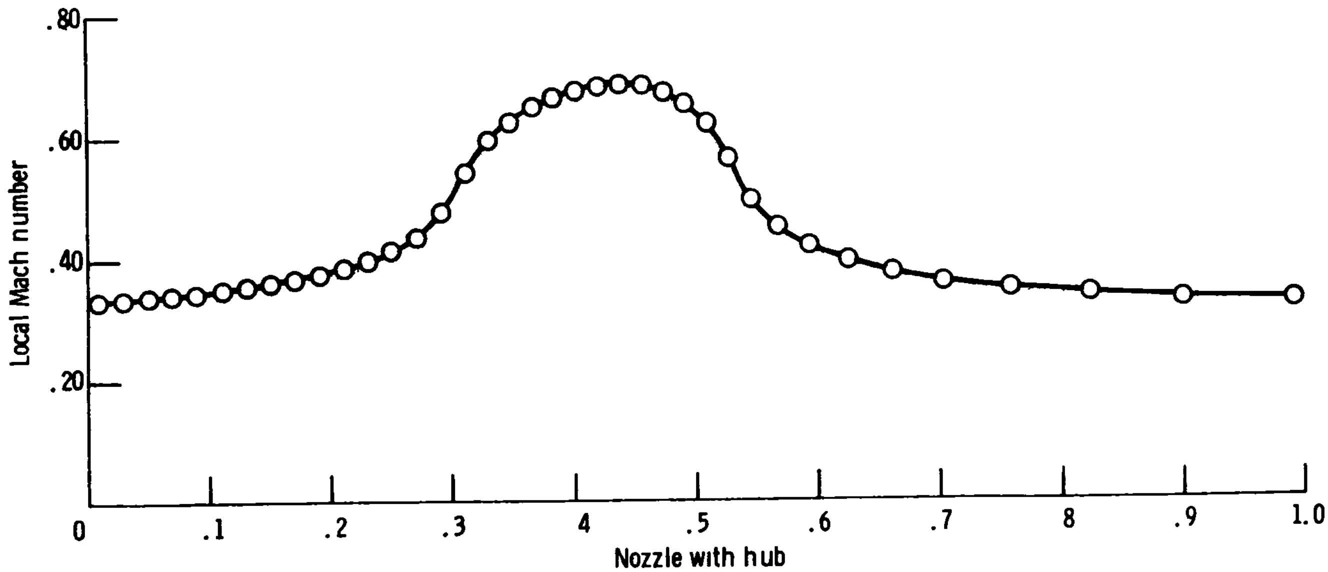


Figure 10. - Surface Mach number distribution from NOZZLE.

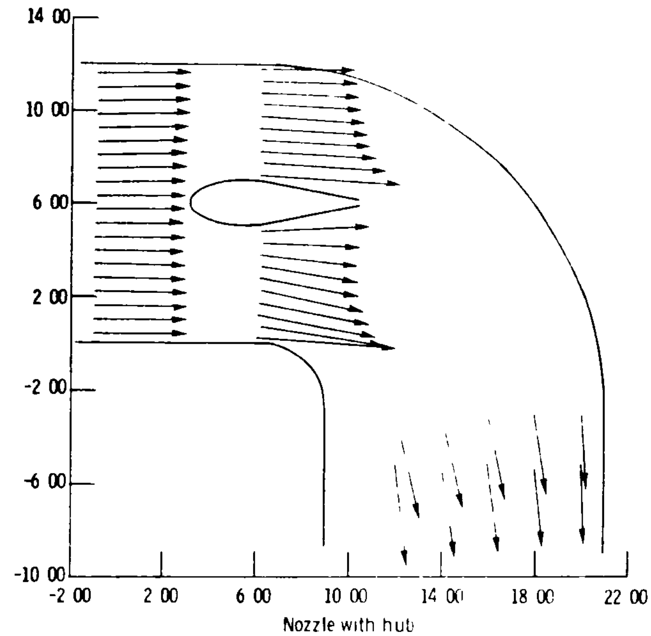
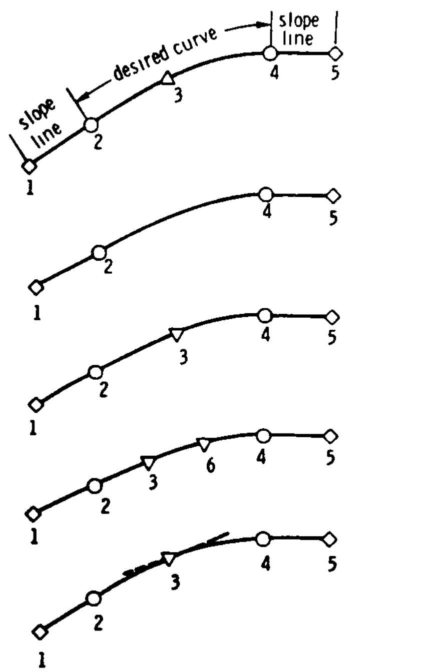


Figure 11 - Vector flow field from nozzle

○ Segment endpoints
◇ Slope line endpoints (length of slope line is arbitrary)
△ Optional superellipse point
▽ Optional bisuperellipse point
◺ Optional bisuperellipse inflection point



(a) Superellipse optional point (X_3, Y_3) specified Exponent N calculated

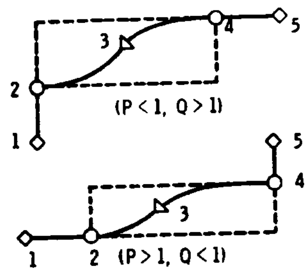
(b) Superellipse exponent N or bisuperellipse exponents P and Q specified

(c) Optional point (X_3, Y_3) and exponent P or Q specified Exponent Q or P respectively calculated.

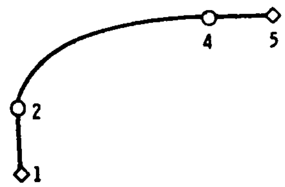
(d) Optional points (X_3, Y_3) and (X_6, Y_6) specified. Exponents P and Q calculated

(e) Optional point (X_3, Y_3) and slope $(dy/dx)_3$. Exponents P and Q calculated.

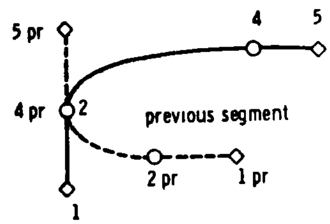
Figure 12. - Sketches for SCIRCL input Bisuperellipse options



(f) Bisuperellipse with inflection point. Axial location X_3 and slope $(dy/dx)_3$ of inflection point specified. Note that the slope line requirements for this option are different from all other options. One slope line must be perpendicular to curve and one must be tangent, thus there are two possibilities as shown. Both lines must lie away from and outside the 'box' surrounding the desired curve. Also, shown are the exponents that will result in each case



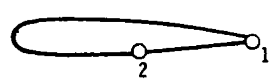
(g) Curvature at either point 2 or point 4 specified



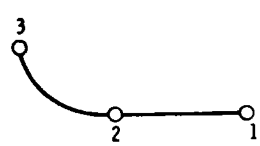
(h) Curvature at endpoint 2 matched to internally calculated curvature at endpoint 4 pr of previous segment

Figure 12. - Concluded

(a) Straight line



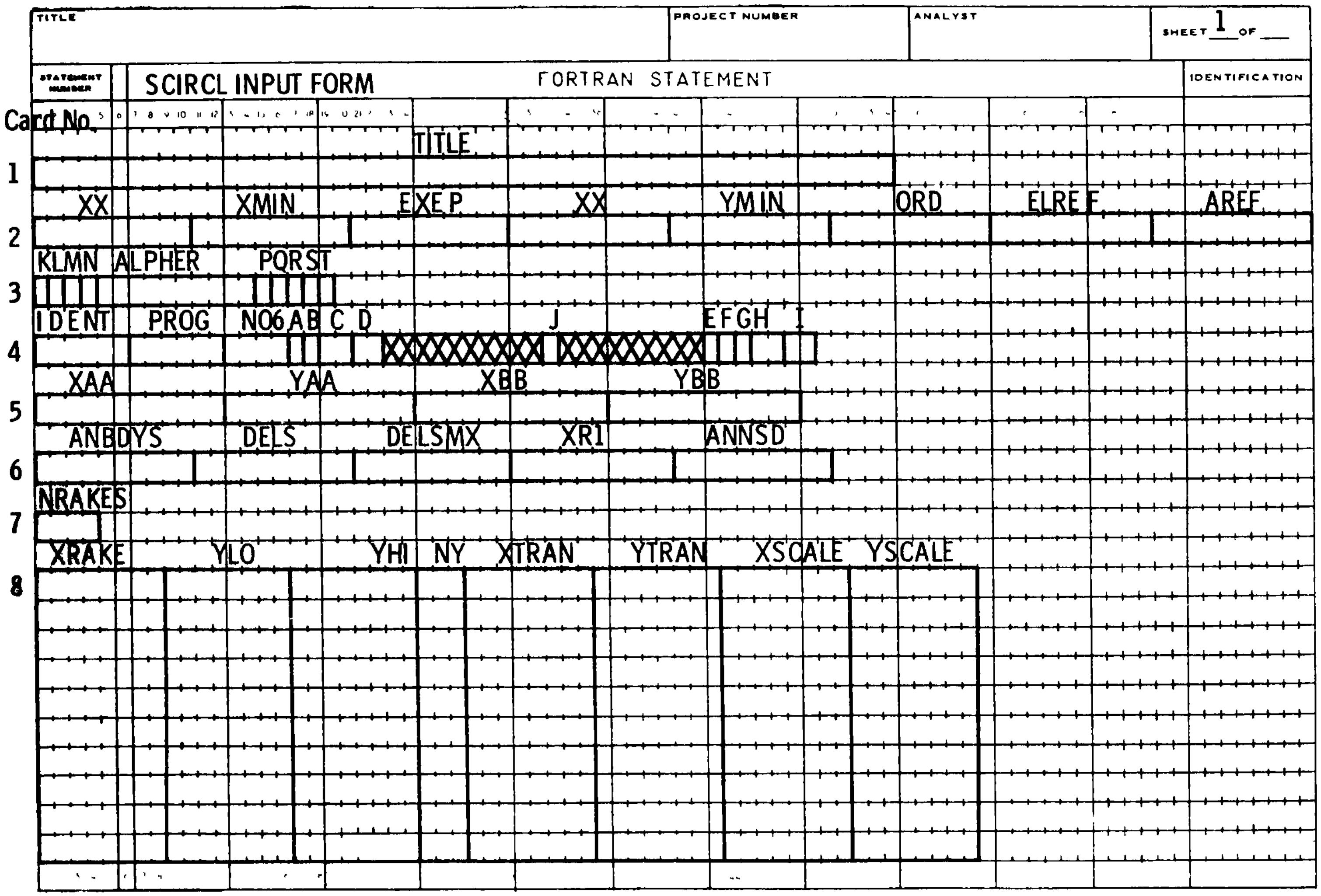
(b) Straight line for closed body



(c) Lemiscate

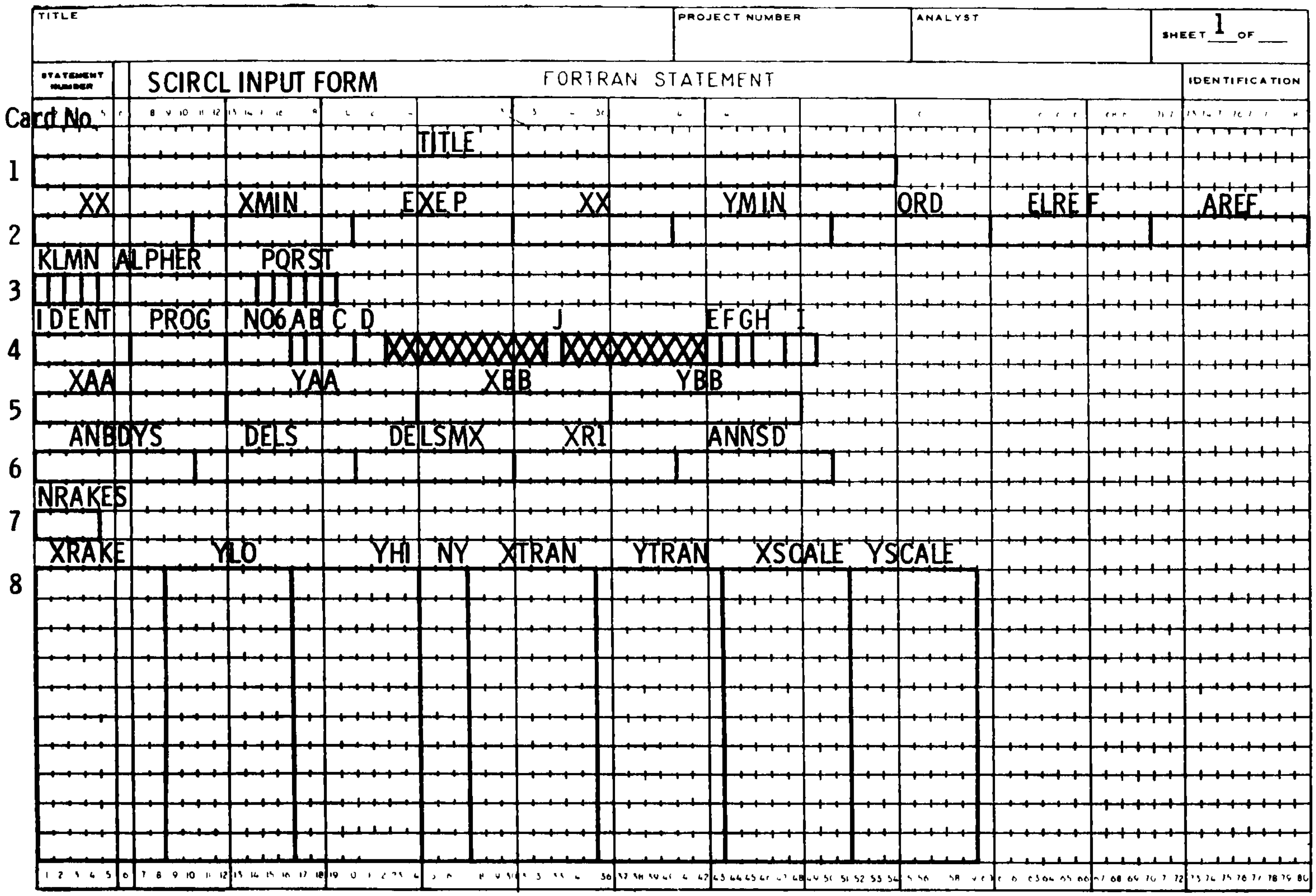
(d) Cubic

Figure 13 - Sketches for SCIRCL input options except bisuperellipse



(a) Title, control, and rake data cards.

Figure 14. - SCIRCL input layout.



(a) Title, control, and rake data cards.

Figure 14. - SCIRCL input layout.

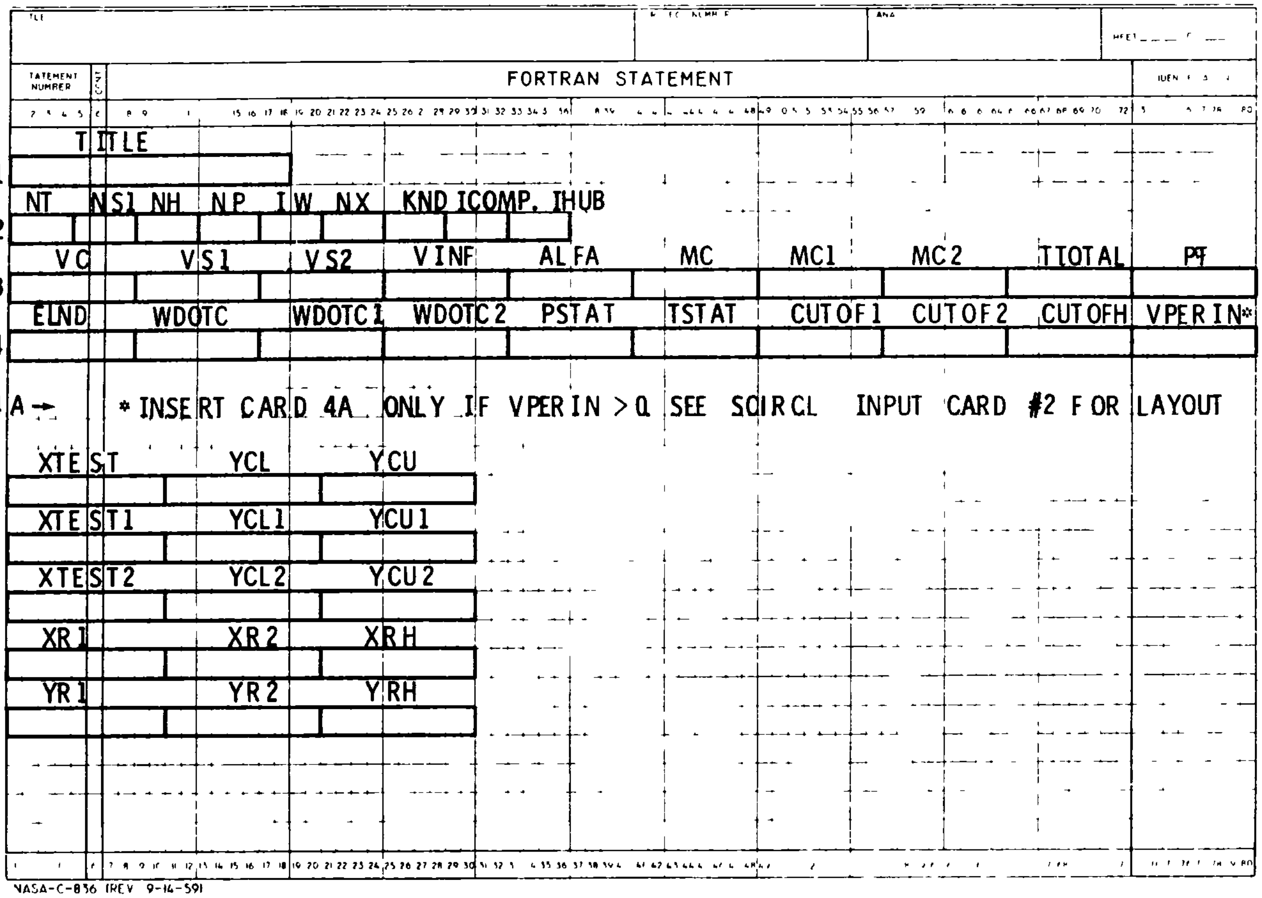
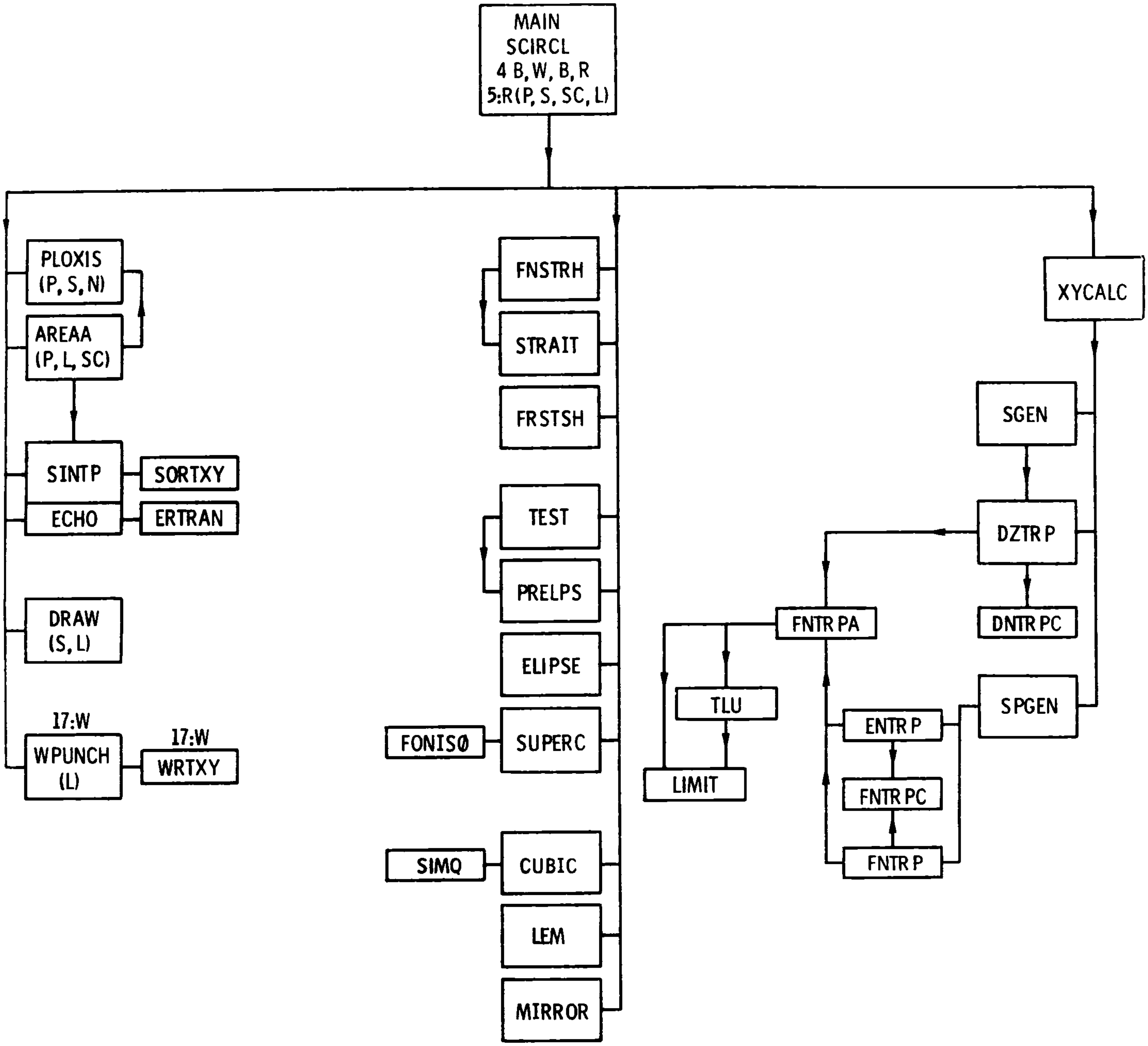


Figure 15. - NOZZLEC input form



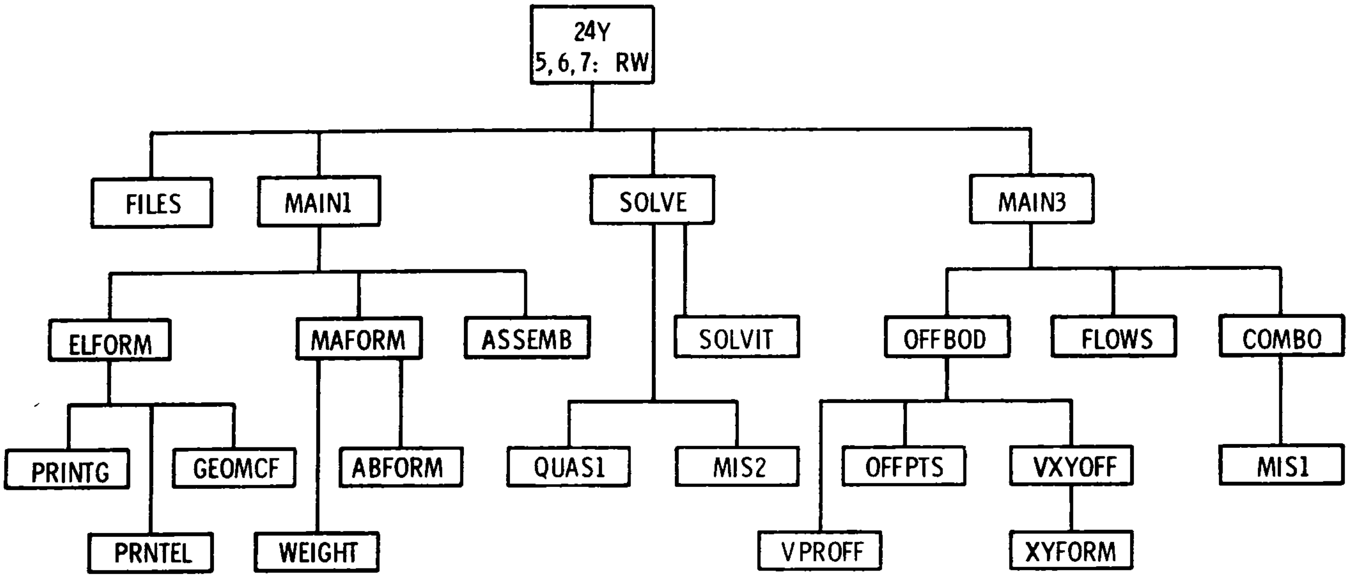
Calcomp Routines Referenced

(L) - Line
(P) - Plot
(S) - Symbol
(SC) - Scale
(N) - Number

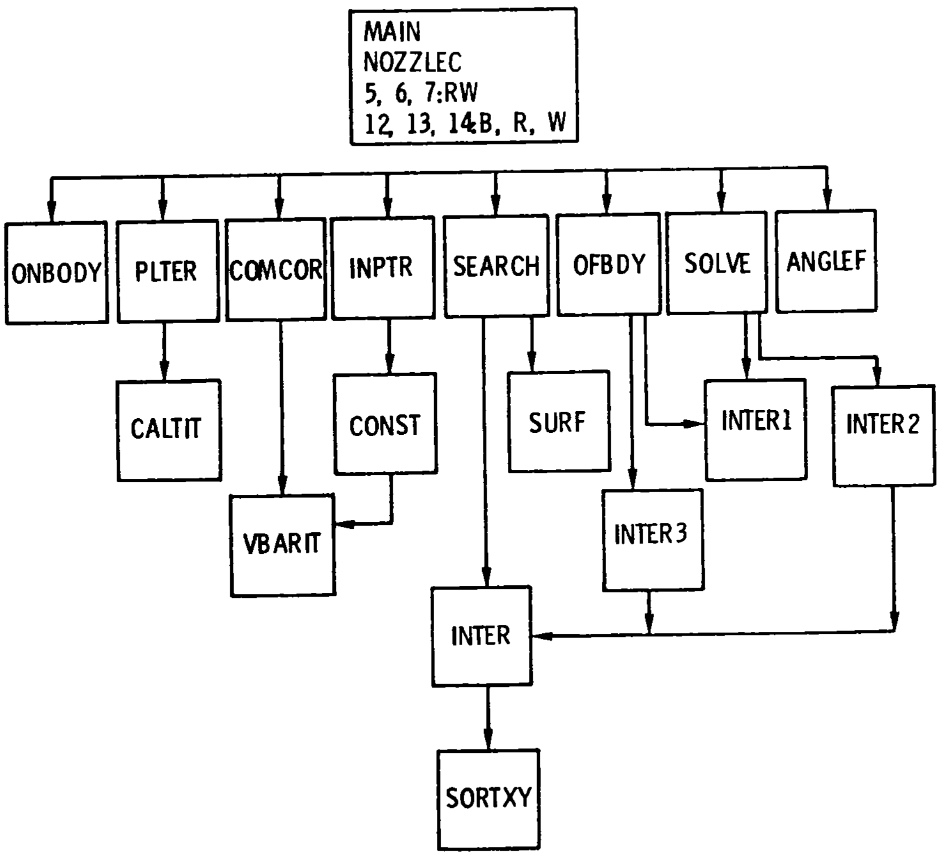
5 - I/O Unit No.
B - Rewind
W - Write
R - Read

(a) SCIRCL.

Figure 16. - Call Sequences.



(b) 24Y



(c) NOZZLEC.

Figure 16. - Concluded.

End of Document